

Proceedings of the

2019 International Conference on Asian Language Processing (IALP)

15-17 November 2019
East China Normal University, Shanghai, P.R.China

IEEE Part Number: CFP1944I-USB
ISBN: 978-1-7281-5013-0

Edited by

Man Lan, Yuanbin Wu, Minghui Dong, Yanfeng Lu, Yan Yang

Organizers



COLIPS

Technical Sponsor



Supporters



Technical support & Inquiries

COLIPS c/o Institute for Infocomm Research
1 Fusionopolis Way, #21-01 Connexis South, Singapore 138632.
Tel: +65-64082757
Email: mhdong@i2r.a-star.edu.sg

Copyright and Reprint Permission: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limit of U.S. copyright law for private use of patrons those articles in this volume that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923. For reprint or republication permission, email to IEEE Copyrights Manager at pubs-permissions@ieee.org. All rights reserved. Copyright ©2019 by IEEE.

Contents

Message -----	I
Committees -----	III
Keynote Speeches-----	VII
Program -----	IX
Author Index -----	XVII
 Papers	
Quantifying the Use of English Words in Urdu News-Stories	
<i>Mehtab Alam Syed, Arif Ur Rahman, Muzammil Khan-----</i>	1
A New Method of Tonal Determination for Chinese Dialects	
<i>Yan Li, Zhiyi Wu-----</i>	7
Duplicate Question Detection based on Neural Networks and Multi-head Attention	
<i>Heng Zhang,Liangyu Chen-----</i>	13
Acoustic Scene Classification Using Deep Convolutional Neural Network via Transfer Learning	
<i>Min Ye, Hong Zhong , Xiao Song , Shilei Huang, Gang Cheng-----</i>	19
Japanese Particle Error Correction employing Classification Model	
<i>Youichiro Ogawa, Kazuhide Yamamoto-----</i>	23
Effect of Preprocessing for Distributed Representations: Case Study of Japanese Radiology Reports	
<i>Taro Tada, Kazuhide Yamamoto-----</i>	29
Character Decomposition for Japanese-Chinese Character-Level Neural Machine	
<i>Jinyi Zhang, Tadahiro Matsumoto-----</i>	35
Japanese grammatical simplification with simplified Corpus	
<i>Yumeto Inaoka and Kazuhide Yamamoto-----</i>	41
Effective Data Augmentation Approaches to End-to-End Task-Oriented Dialogue	
<i>Jun Quan, Deyi Xiong-----</i>	47
Extremely Low Resource Text simplification with Pre-trained Transformer Language Model	
<i>Takumi Maruyama and Kazuhide Yamamoto-----</i>	53
Aspect-based Opinion Mining for Code-Mixed Restaurant Reviews in Indonesia	
<i>Andi Suciati, Indra Budi-----</i>	59
Using Convolutional Neural Network with BERT for Intent Determination	
<i>Changai He,Sibao Chen-----</i>	65
What affects the difficulty of Chinese syntax?	
<i>Yueming Du, Lijiao Yang-----</i>	71

Classified Description and Application of Chinese Constitutive Role	
<i>Mengxiang Wang, Cuiyan Ma-</i>	75
Design and Implementation of Myanmar Speech Synthesis System Based on HMM-DNN	
<i>Mengyuan Liu and Jian Yang-</i>	79
Exploring Letter's Differences between Partial Indonesian Branch Language and English	
<i>Nankai Lin, Sihui Fu, Jiawen Huang, Shengyi Jiang-</i>	84
CIEA: A Corpus for Chinese Implicit Emotion Analysis	
<i>Dawei Li, Jin Wang, Xuejie Zhang-</i>	90
Learning Deep Matching-Aware Network for Text Recommendation using Clickthrough Data	
<i>Haonan Liu, Nankai Lin, Shengyi Jiang, Zitao Chen and Ke Li-</i>	96
Comprehension correlates of the occurrence and deletion of “de” in Mandarin “N1 (de) N2” structures	
<i>Junyuan Zhao, Junru Wu-</i>	102
Effects of English Capitals On Reading Performance of Chinese Learners: Evidence from Eye Tracking	
<i>Yang Wei, Fu Xinyu-</i>	108
A Machine Learning Model for the Dating of Ancient Chinese Texts	
<i>Xuejin Yu, Wei Huangfu-</i>	115
Correlational Neural Network Based Feature Adaptation in L2 Mispronunciation Detection	
<i>Wenwei Dong and Yanlu Xie-</i>	121
Acoustic Cues for Sensory Modality and Referential Vagueness of Personal Pronouns: Me or Not? You or Not?	
<i>Aibao Zhou, Yanbing Hu, Xiaoyong Lu, Tao Pan-</i>	126
Automated Prediction of Item Difficulty in Reading Comprehension Using Long Short-Term Memory	
<i>Li-Huai Lin, Tao-Hsing Chang, Fu-Yuan Hsu-</i>	132
Automatic Meta-evaluation of Low-Resource Machine Translation Evaluation Metrics	
<i>Junting Yu, Wuying Liu , Hongye He , Lin Wang-</i>	136
Combination of Semantic Relatedness with Supervised Method for Word Sense Disambiguation	
<i>Qiaoli Zhou, Yuguang Meng-</i>	142
Acquisition of Knowledge with Time Information from Twitter	
<i>Kohei Yamamoto, Kazutaka Shimada-</i>	148
Research on Chinese Text Error Correction Based on Sequence Model	
<i>Jianyong Duan Yang Yuan, Hao Wang, Xiaopeng Wei, Zheng Tan-----</i>	154
Integrating Knowledge in End-to-End Automatic Speech Recognition for Mandarin-English Code-Switching	
<i>Chia-Yu Li, Ngoc Thang Vu-</i>	160
Improving Question Classification with Hybrid Networks	
<i>Yichao Cao, Miao Li, Tao Feng, Ruijing Wang, Yue Wu-----</i>	166

A Study on the Robustness of Pitch Range Estimation from Brief Speech Segments	
<i>Wenjie Peng, Kaiqi Fu, Wei Zhang, Yanlu Xie, Jinsong Zhang-----</i>	172
Are Scoring Feedback of CAPT Systems Helpful for Pronunciation Correction? --An Exception of Mandarin Nasal Finals	
<i>Rui Cai, Wei Wei, Jinsong Zhang-----</i>	177
Examination-Style Reading Comprehension with Neural augmented Retrieval	
<i>Yiqing Zhang, Hai Zhao and Zhuosheng Zhang-----</i>	182
Automatic Extraction and Quantitative Evaluation of the Character Relationship Networks from Children's Literature works	
<i>Kun Ma, Lijiao Yang-----</i>	188
A Chinese word segment model for energy literature based on Neural Networks with Electricity User Dictionary	
<i>Bochuan Song, Bo Chai, Qiang Zhang, Quanye Jia-----</i>	194
Fusion of Image-text attention for Transformer-based Multimodal Machine Translation	
<i>Junteng Ma, Shihao Qin, Lan Su, Xia Li, Lixian Xiao-----</i>	199
Speech Command Classification System for Sinhala Language based on Automatic Speech Recognition	
<i>Thilini Ranagalage, Lakshika Athapaththu, Pamoda Abeyawardana, Uthayasanker Thayasilvam and Sanath Jayasena-----</i>	205
An Enhancement of Malay Social Media Text Normalization for Lexicon-Based Sentiment Analysis	
<i>Muhammad Fakhrur Razi Abu Bakar, Norisma Idris, Liyana Shuib-----</i>	211
Improving text simplification by corpus expansion with unsupervised Learning	
<i>Akihiro Katsuta and Kazuhide Yamamoto-----</i>	216
Cross Language Information Retrieval Using Parallel Corpus with Bilingual Mapping Method	
<i>Rinaldi Andrian Rahmada, Mirna Adriani and Dipta Tanaya-----</i>	222
Language Detection in Sinhala-English Code-mixed Data	
<i>Ian Smith, Uthayasanker Thayasilvam-----</i>	228
Sinhala and Tamil Speech Intent Identification From English Phoneme Based ASR	
<i>Yohan Karunanayake, Uthayasanker Thayasilvam, Surangika Ranathunga-----</i>	234
An experimental Tagalog Finite State Automata spellchecker with Levenshtein edit-distance feature	
<i>Joseph Marvin R. Imperial Czerittonnie Gail V. Ya-On, Jennifer C. Ureta-----</i>	240
Statistical Machine Learning for Transliteration: Transliterating names between Sinhala, Tamil and English	
<i>H.S. Priyadarshani, M.D.W. Rajapaksha, M.M.S.P. Ranasinghe, K. Sarveswaran, G.V. Dias-----</i>	244
Automatic Recognition of Allusions in Tang Poetry Based on BERT	
<i>Xuemei Tang , Shichen Liang, Jianyu Zheng, Renfen Hu, Zhiying Liu-----</i>	255
Exploring Characteristics of Word Co-occurrence Network in Translated Chinese	
<i>Jianyu Zheng, Kun Ma, Xuemei Tang, Shichen Liang-----</i>	261

Diachronic Synonymy and Polysemy: Exploring Dynamic Relation Between Forms and Meanings of Words Based on Word Embeddings	
<i>Shichen Liang, Jianyu Zheng, Xuemei Tang, Renfen Hu, Zhiying Liu-----</i>	267
Automatic answer ranking based on sememe vector in KBQA	
<i>Yadi Li, Lingling Mu, Hao Li, Hongying Zan-----</i>	273
Improving Japanese-English Bilingual Mapping of Word Embeddings based on Language Specificity	
<i>Yuting Song, Biligsaikhan Batjargal, Akira Maeda-----</i>	279
The Initial Research of Mongolian Literary Corpus-Take the Text of Da.Nachugdorji's Work for Instance	
<i>YinhuaHai-----</i>	284
Towards Robust Neural Machine Reading Comprehension via Question Paraphrases	
<i>Ying Li, Hongyu Li and Jing Liu-----</i>	290
Employing Gated Attention and Multi-similarities to Resolve Document-level Chinese Event Coreference	
<i>Haoyi Cheng, Peifeng Li, Qiaoming Zhu-----</i>	296
A Multi-stage Strategy for Chinese Discourse Tree Construction	
<i>Tishuang Wang, Peifeng Li, Qiaoming Zhu-----</i>	302
Articulatory Features Based TDNN Model for Spoken Language Recognition	
<i>Jiawei Yu, Minghao Guo, Yanlu Xie and Jinsong Zhang-----</i>	308
A Study on Syntactic Complexity and Text Readability of ASEAN English News	
<i>Yusha Zhang, Nankai Lin, Shengyi Jiang-----</i>	313
Tibetan word segmentation method based on CNN-BiLSTM-CRF model	
<i>Lili Wang, Hongwu Yang, Xiaotian Xing, Yajing Yan-----</i>	319
Separate Answer Decoding for Multi-class Question Generation	
<i>Kaili Wu, Yu Hong, Mengmeng Zhu, Hongxuan Tang, Min Zhang-----</i>	325
Converting an Indonesian Constituency Treebank to the Penn Treebank Format	
<i>Jessica Naraiswari Arwidarasti, Ika Alfina, Adila Alfa Krisnadhi-----</i>	331
How to Answer Comparison Questions	
<i>Hongxuan Tang, Yu Hong, Xin Chen, Kaili Wu, Min Zhang-----</i>	337
Improved DNN-HMM English Acoustic Model Specially For Phonotactic Language Recognition	
<i>Wei-wei Liu, Guo-Chun Li Li, Cun-Xue Zhang, Hai-Feng Yan, Jing He, Ying-Xin Gan, Yan-Miao Song, Jian-Hua Zhou, Jian-Zhong Liu, Ying Yin, Ya-nan Li, Yu-Bin Huang, Ting Ruan, Wei Liu, Rui-Li Du, Hua-ying Bai, Wei Li and Sheng-Ge Zhang-----</i>	343
Confidence Modeling for Neural Machine Translation	
<i>Taichi Aida, Kazuhide Yamamoto-----</i>	349
Using WHY-type Question-Answer Pairs to Improve Implicit Causal Relation Recognition	
<i>Huibin Ruan, Yu Hong, Yu Sun, Yang Xu and Min Zhang-----</i>	355
On the Etymology of he 'river' in Chinese	
<i>Huibin Zhuang, Zhanting Bu-----</i>	361
Using Mention Segmentation to Improve Event Detection with Multi-head Attention	
<i>Jiali Chen, Yu Hong, jingli zhang, Jianmin Yao-----</i>	367

Chinese Spelling Check based on Sequence Labeling	
<i>Zijia Han, Zonglin Liu, Qiansheng Wang, Guohong Fu-----</i>	373
Research on Tibetan Text Classification Method Based on Neural Network	
<i>Zhensong Li, Jie Zhu, Zhixiang Luo, Saihu Liu-----</i>	379
BERT with Enhanced Layer for Assistant Diagnosis Based on Chinese Obstetric EMRs	
<i>Kunli Zhang, Chuang Liu, Xuemin Duan, Lijuan Zhou, Yueshu Zhao, Hongying Zan---</i>	384
A Systematic Investigation of Neural Models for Chinese Implicit Discourse Relationship Recognition	
<i>Dejian Li, Man Lan, Yuanbin Wu-----</i>	390
Syntax-aware Transformer Encoder for Neural Machine Translation	
<i>Sufeng Duan, Hai Zhao, Junru Zhou, Rui Wang-----</i>	396
An End-to-End Model Based on TDNN-BiGRU for Keyword Spotting	
<i>Shuzhou Chai, Zhenye Yang, Changsheng Lv and Wei-Qiang Zhang-----</i>	402
Coarse-to-Fine Document Ranking for Multi-Document Reading Comprehension with Answer-Completion	
<i>Hongyu Liu, Shumin Shi, Heyan Huang-----</i>	407
Developing a machine learning-based grade level classifier for Filipino children's literature	
<i>Joseph Marvin Imperial, Rachel Edita Roxas, Erica Mae Campos, Ani Rosa Almario---</i>	413
Exploring Context's Diversity to Improve Neural Language Model	
<i>Yanchun Zhang, Xingyuan Chen, Peng Jin, Yajun Du-----</i>	419
Phrase-Based Tibetan-Chinese Statistical Machine Translation	
<i>YONG Cuo, Xiaodong SHI, NYIMA Tashi, Yidong CHEN-----</i>	424
A General Procedure for Improving Language Models in Low-Resource Speech Recognition	
<i>Qian Liu, Wei-Qiang Zhang, Jia Liu, Yao Liu-----</i>	428
Effect of Music Training on the Production of English Lexical Stress by Chinese English Learners	
<i>Jie Lian, hui feng and Yingjie Zhao-----</i>	434
Carrier Sentence Selection with Word and Context Embeddings	
<i>Chak Yan Yeung, John Lee, Benjamin Tsou-----</i>	439
Multiple-source Entity Linking with Incomplete Sources	
<i>Qinguo Liu, Shui Liu, Lemao Liu, Bo Xiao-----</i>	445
Neural Machine Translation Strategies for Generating Honorific-style Korean	
<i>Lijie Wang, Mei Tu, Mengxia Zhai, Huadong Wang, Song Liu, Sang Ha Kim-----</i>	450
Ranking Like Human: Global-View Matching via Reinforcement Learning for Answer Selection	
<i>Yingxue Zhang, Ping Jian, Ruiying Geng and Fandong Meng-----</i>	456
Development of a Filipino Speaker Diarization in Meeting Room Conversations	
<i>Angelica H. De La Cruz, Rodolfo C. Raga Jr-----</i>	462
Statistical Analysis of Syllable Duration of Uyghur Language	
<i>Askar Hamdulla, Guzalnur Dilmurat, Gulnur Arkin, Mijit Ablimit-----</i>	468

A Comparative Analysis of Acoustic Characteristics between Kazak & Uyghur Mandarin Learners and Standard Mandarin Speakers	
<i>Gulnur Arkin, Gvljan Alijan, Askar Hamdulla, Mijit ablmit-----</i>	474
Construction of Quantitative Index System of Vocabulary Difficulty in Chinese Grade Reading	
<i>Huiping Wang,Lijiao Yang, Huimin Xiao-----</i>	480
An Measurement Method of Ancient Poetry Difficulty for Adaptive Testing	
<i>Huiping Wang, Bihua Wang-----</i>	487
Prosodic Realization of Focus in Changchun Mandarin and Nanjing Mandarin	
<i>Ying Chen, Jiajing Zhang, Bingying Ye, Chenfang Zhou-----</i>	492
Research on New Event Detection Methods for Mongolian News	
<i>Shijie Wang, Feilong Bao, Guanglai Gao-----</i>	498
Improving Mandarin Prosody Boundary Detection by Using Phonetic Information and Deep LSTM Model	
<i>Ju Lin, Zhuanzhuan Ji, Wenwei Dong, Yanlu Xie, Jinsong Zhang-----</i>	504

Welcome Message

Welcome to the 2019 International Conference on Asian Language Processing (IALP) in Shanghai, where the Yangtze River flows into the East China Sea. Shanghai, a global hub with a distinctive culture and the most populous city in China, is also a global center for finance and innovation. It is an extraordinary experience for the conference on Asian language study to take place in this great Asian city.

IALP is the flagship conference of the Chinese and Oriental Languages Information Processing Society (COLIPS). IALP 2019, the 23rd event of this conference series, is jointly organized by COLIPS and East China Normal University (ECNU).

This year, we received 134 paper submissions from 15 countries or regions including China, Japan, Germany, Indonesia, India, Iran, Italy, Sri Lanka, Malaysia, Philippines, Saudi Arabia, Taiwan, and Hong Kong. Finally, 88 high-quality papers have been accepted for presentation at the conference. We are delighted to see the research topics cover a wide range of Asian languages. The program committee also invited two distinguished keynote speakers, Prof. Xuanjing Huang from Fudan University and Dr. Luo Si from Alibaba Inc, to present the latest technologies.

As the conference chairs, we would like to express our gratitude for the dedicated organization work by the local team from East China Normal University and COLIPS team in Singapore. We thank IEEE Singapore SMC Chapter for the technical sponsorship. Thanks also go to Fudan University and Alibaba Inc, for their support to the event.

We would like to thank the technical program chairs, Prof. Man Lan from School of Computer Science and Technology, ECNU and Prof. Xipeng Qiu from Fudan University, Ridong Jiang from Institute for Infocomm Research and COLIPS, Singapore and Prof. Sophia Ananiadou from the University of Manchester, UK for their excellent work in the paper review and the conference program. Thanks also to the program committee members for their diligence and carefulness in the review process.

We thank the organization chairs Prof. Yuanbin Wu, Prof. Yan Yang, Prof. Liang He and Dr. Minghui Dong for the effort on various organization tasks, thank the publication chairs, Dr. Yanfeng Lu and Prof. Xiaofeng He for the publication of the proceedings and the conference abstract book. Our gratitude should also go to the finance chairs, Dr. Yan Wu and Haoyu Ji for managing the conference finance. We thank the publicity team Dr. Jie Zhou, Yang Wei, Tao Ji, Shangqing Zhao, Xin Mao, Changzhi Sun, and Lei Wang for their effort in publicizing the conference. Thanks should also go to Prof. Aimin Zhou, Prof. Xinli Huang and Liqun Liu along with 10 volunteer students for the hard work in the logistics of the event. Great thanks must also go to the authors and the participants of the event for their tremendous support in making the conference a success.

We hope that you would have an enjoyable and productive time in Shanghai, and leave with fond memories. With our best wishes for a successful conference!

Aoying Zhou, Haizhou Li
IALP 2019 Conference Chairs

Conference Committees

Conference Chairs	
Aoying Zhou	East China Normal University, China
Haizhou Li	Institute for Infocomm Research and COLIPS, Singapore
PC Chairs	
Man Lan	East China Normal University, China
Xipeng Qiu	Fudan University, China
Ridong Jiang	Institute for Infocomm Research and COLIPS, Singapore
Sophia Ananiadou	The University of Manchester, Manchester Interdisciplinary Biocentre, UK
Publication Chairs	
Yanfeng Lu	Institute for Infocomm Research and COLIPS, Singapore
Xiaofeng He	East China Normal University, China
Organizing Chairs	
Yuanbin Wu	East China Normal University, China
Liang He	East China Normal University, China
Minghui Dong	Institute for Infocomm Research and COLIPS, Singapore
Local Chairs	
Yan Yang	East China Normal University, China
Xinli Huang	East China Normal University, China
Finance Chairs	
Haoyu Ji	East China Normal University, China
Yan Wu	Institute for Infocomm Research and COLIPS, Singapore
Publicity Chairs	
Yang Wei	East China Normal University, China
Tao Ji	East China Normal University, China

Shangqing Zhao	East China Normal University, China
Xin Mao	East China Normal University, China
Jie Zhou	East China Normal University, China
Changzhi Sun	East China Normal University, China
Lei Wang	Institute for Infocomm Research and COLIPS, Singapore

Program Committee

Dahu Baiyila	Inner Mongolia University
Rafael E. Banchs	Nanyang Technological University
Aliridho Barakbah	Politeknik Elektronika Negeri Surabaya
Moch Arif Bijaksana	Telkom University
Shu Cai	Google
Tru Cao	Ho Chi Minh City University of Technology
Eric Castelli	International Research Institute MICA - CNRS
Paul Chan	Institute for Infocomm Research, A-Star
Chin-Chen Chang	Feng Chia University
Tao-Hsing Chang	National Kaohsiung University of Science and Technology
Qin Chen	Fudan University
Wenliang Chen	Soochow University
Ying Chen	Nanjing University of Science and Technology
Yidong Chen	Xiamen University
Anfeng Cheng	Institute of Information Engineering, Chinese Academy of Sciences
Key-Sun Choi	KAIST
Luis Fernando D'Haro	Universidad Politecnica de Madrid
Ning Dai	Fudan University
Rohan Kumar Das	National University Singapore
Minghui Dong	Institute for Infocomm Research, A-Star
Yupei Du	East China Normal University
Wenhe Feng	Guangdong University of Foreign Studies
Helena Gao	Nanyang Technological University
Tao Gui	Fudan university
Guoxiu He	Wuhan University
Lin He	Wuhan University
Afrida Helen	Politeknik Elektronika Negeri Surabaya
Jiafei Hong	National Taiwan Normal University
Huichen Hsiao	National Taiwan Normal University
Shu-Kai Hsieh	National Taiwan University
Sanath Jayasena	University of Moratuwa
Yu Ji	East China Normal University

Yuxiang Jia	Zhengzhou University
Ridong Jiang	Institute for Infocomm Research, A-Star
Oi Yee Kwong	The Chinese University of Hong Kong
Man Lan	East China Normal University
Lung-Hao Lee	National Central University
Baoli Li	Bo-zhi.com Inc.
Junhui Li	Soochow University
Zhifu Liu	Wuhan University
Jiangming Liu	University of Edinburgh
Donghong Liu	Central China Normal University
Qun Liu	Huawei Noah's Ark Lab
Mengbin Liu	College of Foreign Languages of Sun Yat-sen University
Xiaoyong Lu	Northwest Normal University
Yanfeng Lu	Institute for Infocomm Research, A-Star
Xugang Lu	NICT
Yaojie Lu	Institute of Software, Chinese Academy of Sciences
Maulik Madhavi	National University of Singapore
Philippe Martin	Universite Paris Diderot
Masaki Murata	Tottori University
Arif Bijaksana Putra Negara	Universitas Tanjungpura
Yongkyoon No	Chungnam National University
Anto Satriyo Nugroho	Agency for Assessment & Application of Technology
Hilman Pardede	Indonesian Institute of Sciences
Minlong Peng	Fudan University
Haoliang Qi	Heilongjiang Institute of Technology
Tao Qian	Wuhan University
Shumin Shi	Beijing Institute of Technology
Liye Shi	East China Normal University
Xiaodong Shi	Xiamen University
Yangyang Shi	Mobvoi AI Lab
Kai Song	Soochow University; Alibaba Group
Surya Sumpeno	Surabaya Institute of Technology
Changzhi Sun	East China Normal University
Sung Lin Chen	National Cheng Kung University
Jianhua Tao	Institute of Automation, Chinese Academy of Sciences
Hapnes Toba	Universitas Kristen Maranatha
Zhiqiang Toh	Bigo Technology Pte Ltd
Xinyu Wang	Shanghai Tech University
Yining Wang	Institute of Automation, Chinese Academy of Sciences
Kai Wang	Sun Yat-sen University
Jin Wang	Yunnan University
Lei Wang	Institute for Infocomm Research, A-Star

Zhongqing Wang	Soochow University
Hsin-Min Wang	Academia Sinica
Zhijin Wang	Jimei University
Dewi Wardani	Universitas Sebelas Maret
Zhongyu Wei	Fudan University
Peng Wu	Nanjing University
Junshuang Wu	Beihang University
Xianchao Wu	Microsoft
Yuanbin Wu	East China Normal University
Wen Wu	East China Normal University
Tong Xiao	Northeastern University
Yanlu Xie	Beijing Advanced Innovation Center for Language Resources
Liang-Chih Yu	Yuan Ze University
Yue Zhang	Westlake University
Jiajun Zhang	Institute of Automation, Chinese Academy of Sciences
Meishan Zhang	Tianjin University, China
Haijun Zhang	Xinjiang Normal University
Yingyi Zhang	Nanjing University of Science and Technology
Quan Zhang	Institute of Acoustics, Chinese Academy of Sciences
Xueliang Zhao	Peking University
Sequan Zheng	Harbin Institute of Technology
Zezhi Zheng	Xiamen University
Qiang Zhou	Tsinghua University
Muhua Zhu	Alibaba Inc.

Keynote Speeches

Keynote Speech 1	Representation Learning in Natural Language Processing
Date/Time	November 15, 2019 (Friday) / 08:40-09:30
Abstract	Recently, deep learning provides some powerful new techniques which are successfully applied in NLP tasks, ranging from text classification to sequence labeling, from machine translation to question answering. These neural-based models can not only compete with or in some cases outperform traditional statistical approaches, but also, can be trained with a single end-to-end model, which do not require task-specific feature engineering. In this talk, I will first give a brief overview of current research status about deep learning in NLP, especially neural representation learning, which means to convert text spans, for example, words, phrases, sentences and sentence pairs into real-valued vectors. Next, I will introduce the frontiers in neural representation learning for NLP, ranging from models beyond RNN, such as graph neural networks, transformer and the pre-trained embeddings, to various learning schemes such as transfer learning, multi-task learning and meta learning.



Xuanjing Huang

Professor, PhD Supervisor
Fudan University

Biography

Xuanjing Huang is a Professor of the School of Computer Science, Fudan University, Shanghai, China. She received her Ph.D. degree in Computer Science from Fudan University in 1998. From 2008 to 2009, she is a visiting scholar in CIIR, UMass Amherst. Her research interest includes natural language processing, information retrieval, artificial intelligence, deep learning and data intensive computing. She has published more than 100 papers in major conferences including ACL, SIGIR, IJCAI, AAAI, NIPS, ICML, CIKM, EMNLP, WSDM and COLING. In the research community, she served as the PC Co-Chair of CCL 2019, NLPCC 2017, CCL 2016, SMP 2015 and SMP 2014, the organizer of WSDM 2015, competition chair of CIKM 2014, tutorial chair of COLING 2010, SPC or PC member of past WSDM, SIGIR, WWW, CIKM, ACL, IJCAI, KDD, EMNLP, COLING and many other conferences.

Keynote Speech 2	Natural Language Processing R&D for E-commerce and Beyond
Date/Time	November 16, 2019 (Saturday) / 08:30-09:20
Abstract	Natural Language Processing (NLP) and related technologies are critical for the success of Internet business like e-commerce. Alibaba's NLP R&D aims at supporting the business demands of Alibaba's eco-system, creating new opportunities for Alibaba's partners and advancing the state-of-the-art of NLP technologies. This talk will introduce our efforts to build NLP technique platform and machine translation (MT) platform that power Alibaba's eco-system. Furthermore, some recent research work will be presented on product title compression with user-log information, sentiment classification with questions & answers, machine reading comprehension in real-world custom service, and cascade ranking for large-scale e-commerce search. The R&D work attracts hundreds of millions of users and generates significant business value every day.

**Luo Si**

Senior Researcher
Alibaba Inc.

Biography

Dr. Luo Si is a Distinguished Engineer / Vice President of Alibaba Group Inc. He is also the Chief Scientist of Natural Language Processing with Alibaba DAMO Academy. He leads a cross-country team in China, USA and Singapore with the focus on developing cutting edge technologies in natural language processing, machine translation, text mining and information retrieval. The work attracts hundreds of millions of users and generates millions of revenues each day. Luo has published more than 150 journal and conference papers with substantial citations. His research has obtained many industry awards from Yahoo!, Google and Alibaba as well as NSF career award. Prior to joining Alibaba in 2014, he was a tenured Professor with Purdue University. He obtained BS, MS and Ph.D. degrees in computer science from Tsinghua University and Carnegie Mellon University.

Conference Program

November 15, 2019			
10:10-11:25	Oral Session 1(AS)	Chair: Minghui Dong	Sunshine Hall
AS1	Acoustic Scene Classification Using Deep Convolutional Neural Network via Transfer Learning <i>Min Ye, Hong Zhong , Xiao Song , Shilei Huang, Gang Cheng</i>		
AS2	Acoustic Cues for Sensory Modality and Referential Vagueness of Personal Pronouns: Me or Not? You or Not? <i>Aibao Zhou, Yanbing Hu, Xiaoyong Lu, Tao Pan</i>		
AS3	Integrating Knowledge in End-to-End Automatic Speech Recognition for Mandarin-English Code-Switching <i>Chia-Yu Li, Ngoc Thang Vu</i>		
AS4	A Study on the Robustness of Pitch Range Estimation from Brief Speech Segments <i>Wenjie Peng, Kaiqi Fu, Wei Zhang, Yanlu Xie, Jinsong Zhang</i>		
AS5	Improving Mandarin Prosody Boundary Detection by Using Phonetic Information and Deep LSTM Model <i>Ju Lin, Zhuanzhuan Ji, Wenwei Dong, Yanlu Xie, Jinsong Zhang</i>		
11:35-12:35	Oral Session 2 (DD)	Chair: Xipeng Qiu	Sunshine Hall
DD1	Effective Data Augmentation Approaches to End-to-End Task-Oriented Dialogue <i>Jun Quan, Deyi Xiong</i>		
DD2	A Multi-stage Strategy for Chinese Discourse Tree Construction <i>Tishuang Wang, Peifeng Li, Qiaoming Zhu</i>		
DD3	A Systematic Investigation of Neural Models for Chinese Implicit Discourse Relationship Recognition <i>Dejian Li, Man Lan, Yuanbin Wu</i>		
DD4	Neural Machine Translation Strategies for Generating Honorific-style Korean <i>Lijie Wang, Mei Tu, Mengxia Zhai, Huadong Wang, Song Liu, Sang Ha Kim</i>		

14:00-15:15	Oral Session 3 (IE)	Chair: Changzhi Sun	Sunshine Hall
IE1	Effect of Preprocessing for Distributed Representations: Case Study of Japanese Radiology Reports <i>Taro Tada, Kazuhide Yamamoto</i>		
IE2	Acquisition of Knowledge with Time Information from Twitter <i>Kohei Yamamoto, Kazutaka Shimada</i>		
IE3	Automatic Extraction and Quantitative Evaluation of the Character Relationship Networks from Children's Literature works <i>Kun Ma, Lijiao Yang</i>		
IE4	Employing Gated Attention and Multi-similarities to Resolve Document-level Chinese Event Coreference <i>Haoyi Cheng, Peifeng Li, Qiaoming Zhu</i>		
IE5	Language Detection in Sinhala-English Code-mixed Data <i>Ian Smith, Uthayasanker Thayasivam</i>		
15:30-16:45	Oral Session 4 (LI)	Chair: Huibing Zhuang	Sunshine Hall
LI1	Classified Description and Application of Chinese Constitutive Role <i>Mengxiang Wang, Cuiyan Ma</i>		
LI2	Exploring Letter's Differences between Partial Indonesian Branch Language and English <i>Nankai Lin, Sihui Fu, Jiawen Huang, Shengyi Jiang</i>		
LI3	Comprehension correlates of the occurrence and deletion of "de" in Mandarin "N1 (de) N2" structures <i>Junyuan Zhao, Junru Wu</i>		
LI4	Effects of English Capitals On Reading Performance of Chinese Learners: Evidence from Eye Tracking <i>Yang Wei, Fu Xinyu</i>		
LI5	On the Etymology of he 'river' in Chinese <i>Huibin Zhuang, Zhanting Bu</i>		
15:30-16:45	Poster 1		Sunshine Hall
AP2 Poster	Japanese Particle Error Correction employing Classification Model <i>Youichiro Ogawa, Kazuhide Yamamoto</i>		
AP3 Poster	Research on Chinese Text Error Correction Based on Sequence Model <i>Jianyong Duan Yang Yuan, Hao Wang, Xiaopeng Wei, Zheng Tan</i>		
AP4 Poster	Are Scoring Feedback of CAPT Systems Helpful for Pronunciation Correction? --An Exception of Mandarin Nasal Finals <i>Rui Cai, Wei Wei, Jinsong Zhang</i>		
AP6 Poster	An experimental Tagalog Finite State Automata spellchecker with Levenshtein edit-distance feature <i>Joseph Marvin R. Imperial Czeritonnie Gail V. Ya-On, Jennifer C. Ureta</i>		
AP7 Poster	Chinese Spelling Check based on Sequence Labeling <i>Zijia Han, Zonglin Liu, Qiansheng Wang, Guohong Fu</i>		
AP8 Poster	Research on Tibetan Text Classification Method Based on Neural Network <i>Zhensong Li, Jie Zhu, Zhixiang Luo, Saihu Liu</i>		

AP9 Poster	Developing a machine learning-based grade level classifier for Filipino children's literature <i>Joseph Marvin Imperial, Rachel Edita Roxas, Erica Mae Campos, Ani Rosa Almario</i>
AP10 Poster	Development of a Filipino Speaker Diarization in Meeting Room Conversations <i>Angelica H. De La Cruz, Rodolfo C. Raga Jr</i>
AP11 Poster	Construction of Quantitative Index System of Vocabulary Difficulty in Chinese Grade Reading <i>Huiping Wang, Lijiao Yang, Huimin Xiao</i>
AP12 Poster	An Measurement Method of Ancient Poetry Difficulty for Adaptive Testing <i>Huiping Wang, Bihua Wang</i>
AP13 Poster	Prosodic Realization of Focus in Changchun Mandarin and Nanjing Mandarin <i>Ying Chen, Jiajing Zhang, Bingying Ye, Chenfang Zhou</i>

November 16, 2019			
09:40-10:55	Oral Session 5 (MT)	Chair: Tao Ji	Meeting Room
MT1	Character Decomposition for Japanese-Chinese Character-Level Neural Machine Translation <i>Jinyi Zhang, Tadahiro Matsumoto</i>		
MT2	Fusion of Image-text attention for Transformer-based Multimodal Machine Translation <i>Junteng Ma, Shihao Qin, Lan Su, Xia Li, Lixian Xiao</i>		
MT3	Statistical Machine Learning for Transliteration: Transliterating names between Sinhala, Tamil and English <i>H.S. Priyadarshani, M.D.W. Rajapaksha, M.M.S.P. Ranasinghe, K. Sarveswaran, G.V. Dias</i>		
MT4	Confidence Modeling for Neural Machine Translation <i>Taichi Aida, Kazuhide Yamamoto</i>		
MT5	Syntax-aware Transformer Encoder for Neural Machine Translation <i>Sufeng Duan, Hai Zhao, Junru Zhou, Rui Wang</i>		
11:10-12:25	Oral Session 6 (SY) Chair: Kazuhide Yamamoto	Meeting Room	
SY1	What affects the difficulty of Chinese syntax? <i>Yueming Du, Lijiao Yang</i>		
SY2	A Chinese word segment model for energy literature based on Neural Networks with Electricity User Dictionary <i>Bochuan Song, Bo Chai, Qiang Zhang, Quanye Jia</i>		
SY3	A Study on Syntactic Complexity and Text Readability of ASEAN English News <i>Yusha Zhang, Nankai Lin, Shengyi Jiang</i>		
SY4	Tibetan word segmentation method based on CNN-BiLSTM-CRF model <i>Lili Wang, Hongwu Yang, Xiaotian Xing, Yajing Yan</i>		
SY5	Converting an Indonesian Constituency Treebank to the Penn Treebank Format <i>Jessica Naraiswari Arwidarasti, Ika Alfina, Adila Alfa Krisnadhi</i>		

14:00-16:00	Poster 2	Meeting Room
QA1 Poster	Duplicate Question Detection based on Neural Networks and Multi-head Attention <i>Heng Zhang, Liangyu Chen</i>	
QA2 Poster	Automated Prediction of Item Difficulty in Reading Comprehension Using Long Short-Term Memory <i>Li-Huai Lin, Tao-Hsing Chang, Fu-Yuan Hsu</i>	
QA3 Poster	Improving Question Classification with Hybrid Networks <i>Yichao Cao, Miao Li, Tao Feng, Ruijing Wang, Yue Wu</i>	
QA4 Poster	Separate Answer Decoding for Multi-class Question Generation <i>Kaili Wu, Yu Hong, Mengmeng Zhu, Hongxuan Tang, Min Zhang</i>	
QA5 Poster	How to Answer Comparison Questions <i>Hongxuan Tang, Yu Hong, Xin Chen, Kaili Wu, Min Zhang</i>	
QA6 Poster	Coarse-to-Fine Document Ranking for Multi-Document Reading Comprehension with Answer-Completion <i>Hongyu Liu, Shumin Shi, Heyan Huang</i>	
SE1 Poster	Aspect-based Opinion Mining for Code-Mixed Restaurant Reviews in Indonesia <i>Andi Suciati, Indra Budi</i>	
SE2 Poster	CIEA: A Corpus for Chinese Implicit Emotion Analysis <i>Dawei Li, Jin Wang, Xuejie Zhang</i>	
SE3 Poster	An Enhancement of Malay Social Media Text Normalization for Lexicon-Based Sentiment Analysis <i>Muhammad Fakhrur Razi Abu Bakar, Norisma Idris, Liyana Shuib</i>	
SE4 Poster	Sinhala and Tamil Speech Intent Identification From English Phoneme Based ASR <i>Yohan Karunanayake, Uthayasanker Thayasiyam, Surangika Ranathunga</i>	
SE5 Poster	Exploring Characteristics of Word Co-occurrence Network in Translated Chinese <i>Jianyu Zheng, Kun Ma, Xuemei Tang, Shichen Liang</i>	
SE6 Poster	Improving Japanese-English Bilingual Mapping of Word Embeddings based on Language Specificity <i>Yuting Song, Biligsaikhan Batjargal, Akira Maeda</i>	
SE7 Poster	Exploring Context's Diversity to Improve Neural Language Model <i>Yanchun Zhang, Xingyuan Chen, Peng Jin, Yajun Du</i>	
SE8 Poster	A General Procedure for Improving Language Models in Low-Resource Speech Recognition <i>Qian Liu, Wei-Qiang Zhang, Jia Liu, Yao Liu</i>	
SE9 Poster	Using Convolutional Neural Network with BERT for Intent Determination <i>Changai He, Sibao Chen</i>	
IE5 Poster	Multiple-source Entity Linking with Incomplete Sources <i>Qinguo Liu, Shui Liu, Lemao Liu, Bo Xiao</i>	
14:45-16:00	Oral Session 7(SE) Chair: Shumin Shi	Meeting Room
SE1	Combination of Semantic Relatedness with Supervised Method for Word Sense Disambiguation <i>Qiaoli Zhou, Yuguang Meng</i>	

SE3	Automatic Recognition of Allusions in Tang Poetry Based on BERT <i>Xuemei Tang, Shichen Liang, Jianyu Zheng, Renfen Hu, Zhiying Liu</i>	
SE4	Diachronic Synonymy and Polysemy: Exploring Dynamic Relation Between Forms and Meanings of Words Based on Word Embeddings <i>Shichen Liang, Jianyu Zheng, Xuemei Tang, Renfen Hu, Zhiying Liu</i>	
SE5	Carrier Sentence Selection with Word and Context Embeddings <i>Chak Yan Yeung, John Lee, Benjamin Tsou</i>	
16:15-17:45	Oral Session 8 (LR)	Chair: Chia-Yu Li
LR1	A New Method of Tonal Determination for Chinese Dialects <i>Yan Li, Zhiyi Wu</i>	
LR2	Extremely Low Resource Text simplification with Pre-trained Transformer Language Model <i>Takumi Maruyama and Kazuhide Yamamoto</i>	
LR3	Automatic Meta-evaluation of Low-Resource Machine Translation Evaluation Metrics <i>Junting Yu, Wuying Liu, Hongye He, Lin Wang</i>	
LR4	The Initial Research of Mongolian Literary Corpus-Take the Text of Da.Nachugdorji's Work for Instance <i>YinhuaHai</i>	
LR5	Statistical Analysis of Syllable Duration of Uyghur Language <i>Askar Hamdulla, Guzalnur Dilmurat, Gulnur Arkin, Mijit Abilimit</i>	
AP1	Quantifying the Use of English Words in Urdu News-Stories <i>Mehtab Alam Syed, Arif Ur Rahman and Muzammil Khan</i>	

November 17, 2019		
08:30-09:45	Oral Session 9 (QA) Chair: Kazutaka Shimada	Meeting Room
QA1	Examination-Style Reading Comprehension with Neural augmented Retrieval <i>Yiqing Zhang, HaiZhao, ZhuoshengZhang</i>	
QA2	Automatic answer ranking based on sememe vector in KBQA <i>Yadi Li, Lingling Mu, Hao Li, Hongying Zan</i>	
QA3	Towards Robust Neural Machine Reading Comprehension via Question Paraphrases <i>Ying Li, Hongyu Li, Jing Liu</i>	
QA4	Using WHY-type Question-Answer Pairs to Improve Implicit Causal Relation Recognition <i>Huibin Ruan, Yu Hong, Yu Sun, Yang Xu, Min Zhang</i>	
QA5	Ranking Like Human: Global-View Matching via Reinforcement Learning for Answer Selection <i>Yingxue Zhang, Ping Jian, Ruiying Geng, Yuansheng Song, Fandong Meng</i>	
08:30-09:45	Poster 3	Meeting Room
IE1 Poster	A Machine Learning Model for Dating of Ancient Chinese Texts <i>Xuejin Yu, Wei Huangfu</i>	
IE2 Poster	Using Mention Segmentation to Improve Event Detection with Multi-head Attention <i>Jiali Chen, Yu Hong, Jingli Zhang, Jianmin Yao</i>	
IE3 Poster	An End-to-End Model Based on TDNN-BiGRU for Keyword Spotting <i>Shuzhou Chai, Zhenyu Yang, Changsheng Lv, Wei-Qiang Zhang</i>	
IE4 Poster	Research on New Event Detection Methods for Mongolian News <i>Shijie Wang, Feilong Bao, Guanglai Gao</i>	
MT1 Poster	Cross Language Information Retrieval Using Parallel Corpus with Bilingual Mapping Method <i>Rinaldi Andrian Rahmada, Mirna Adriani, Dipta Tanaya</i>	
MT2 Poster	Phrase-Based Tibetan-Chinese Statistical Machine Translation <i>YONG Cuo, Xiaodong SHI, NYIMA Tashi, Yidong CHEN</i>	
AS1 Poster	Design and Implementation of Burmese Speech Synthesis System Based on HMM-DNN <i>Mengyuan Liu, Jian Yang</i>	
AS2 Poster	Speech Command Classification System for Sinhala Language based on Automatic Speech Recognition <i>Thilini Dinushika, Lakshika Kavmini, Pamoda Abeyawardhana, Uthayasanker Thayasilam, Sanath Jayasena</i>	
AS3 Poster	Articulatory Features Based TDNN Model for Spoken Language Recognition <i>Jiawei Yu, Minghao Guo, Yanlu Xie, Jinsong Zhang</i>	
AS4 Poster	Improved DNN-HMM English Acoustic Model Specially For Phonotactic Language Recognition <i>Wei-Wei Liu, Guo-Chun Li, Cun-Xue Zhang, Hai-Feng Yan, Jing He, Ying-Xin Gan, Yan-Miao Song, Jian-Hua Zhou, Jian-Zhong Liu, Ying Yin, Ya-Nan Li*, Yu-Bin Huang, Ting Ruan, Wei Liu, Rui-Li Du, Hua-ying Bai, Wei Li, Sheng-Ge Zhang</i>	

AS5 Poster	Effect of Music Training on the Production of English Lexical Stress by Chinese English Learners <i>Hui Feng, Jie Lian, Yingjie Zhao</i>	
AS6 Poster	A Comparative Analysis of Acoustic Characteristics between Kazak & Uyghur Mandarin Learners and Standard Mandarin Speakers <i>Gulnur Arkin, Gvjan Aljian, Askar Hamdulla, Mijit ablimit</i>	
10:00-11:15	Oral Session 10 (AP)	Chair: Xin Mao
AP1	Japanese grammatical simplification with simplified corpus <i>Yumeto Inaoka, Kazuhide Yamamoto</i>	
AP2	Learning Deep Matching-Aware Network for Text Recommendation using Clickthrough Data <i>Haonan Liu, Nankai Lin, Zitao Chen, Ke Li, Shengyi Jiang</i>	
AP3	Correlational Neural Network Based Feature Adaptation in L2 Mispronunciation Detection <i>Wenwei Dong, Yanlu Xie</i>	
AP4	Improving text simplification by corpus expansion with unsupervised learning <i>Akihiro Katsuta, Kazuhide Yamamoto</i>	
AP5	BERT with Enhanced Layer for Assistant Diagnosis Based on Chinese Obstetric EMRs <i>Kunli Zhang, Chuang Liu, Xuemin Duan, Lijuan Zhou, Yueshu Zhao, Hongying Zan</i>	
11:30-12:00	Best Paper Award, Next Conference, Closing	Meeting Room

Author Index

A

Adila Alfa Krisnadhi	331
Aibao Zhou	126
Akihiro Katsuta	216
Akira Maeda	279
Andi Suciati	59
Angelica H. De La Cruz	462
Ani Rosa Almario	413
Arif Ur Rahman	1
Askar Hamdulla	468,474

D

Dawei Li	90
Dejian Li	390
Deyi Xiong	47
Dipta Tanaya	222

E

Erica Mae Campos	413
------------------	-----

B

Benjamin Tsou	439
Bihua Wang	487
Biligsaikhan Batjargal	279
Bingying Ye	492
Bo Chai	194
Bo Xiao	445
Bochuan Song	194

F

Fandong Meng	456
Feilong Bao	498
Fu Xinyu	108
Fu-Yuan Hsu	132

C

Chak Yan Yeung	439
Changai He	65
Changsheng Lv	402
Chenfang Zhou	492
Chia-Yu Li	160
Chuang Liu	384
Cuiyan Ma	75
Cun-Xue Zhang	343
Czerittonnie Gail V. Ya-On	240

G

G.V. Dias	244
Gang Cheng	19
Guanglai Gao	498
Gulnur Arkin	468,474
Guo-Chun Li Li	343
Guohong Fu	373
Guzalnur Dilmurat	468
Gvljan Alijan	474

H

H.S. Priyadarshani	244
Hai Zhao	182,396
Hai-Feng Yan	343

Hao Li	273	Jie Zhu	379
Hao Wang	154	Jin Wang	90
Haonan Liu	96	Jing He	343
Haoyi Cheng	296	Jing Liu	290
Heng Zhang	13	Jingli Zhang	367
Heyan Huang	407	Jinsong Zhang	172,177,308,504
Hong Zhong	19	Jinyi Zhang	35
Hongwu Yang	319	John Lee	439
Hongxuan Tang	325,337	Joseph Marvin Imperial	413
Hongye He	136	Joseph Marvin R. Imperial	240
Hongying Zan	273,384	Ju Lin	504
Hongyu Li	290	Jun Quan	47
Hongyu Liu	407	Junru Wu	102
Hua-ying Bai	343	Junru Zhou	396
Huadong Wang	450	Junteng Ma	199
Hui Feng	434	Junting Yu	136
Huibin Ruan	355	Junyuan Zhao	102
Huibin Zhuang	361		
Huimin Xiao	480		
Huiping Wang	480,487		

I

Ian Smith	228
Ika Alfina	331
Indra Budi	59

J

Jennifer C. Ureta	240
Jessica Naraiswari Arwidarasti	331
Jia Liu	428
Jiajing Zhang	492
Jiali Chen	367
Jian Yang	79
Jian-Hua Zhou	343
Jian-Zhong Liu	343
Jianmin Yao	367
Jianyong Duan	154
Jianyu Zheng	255,261,267
Jiawei Yu	308
Jiawen Huang	84
Jie Lian	434

K

K. Sarveswaran	244
Kaili Wu	325,337
Kaiqi Fu	172
Kazuhide Yamamoto	23,29,41,53,216,349
Kazutaka Shimada	148
Ke Li	96
Kohei Yamamoto	148
Kun Ma	188,261
Kunli Zhang	384

L

Lakshika Athapaththu	205
Lan Su	199
Lemao Liu	445
Li-Huai Lin	132
Liangyu Chen	13
Lijiao Yang	71,188,480
Lijie Wang	450
Lijuan Zhou	384
Lili Wang	319
Lin Wang	136

Lingling Mu	273	Q	
Lixian Xiao	199	Qian Liu	428
Liyana Shuib	211	Qiang Zhang	194
M		Qiansheng Wang	373
M.D.W. Rajapaksha	244	Qiaoli Zhou	142
M.M.S.P. Ranasinghe	244	Qiaoming Zhu	296,302
Man Lan	390	Qinguo Liu	445
Mehtab Alam Syed	1	Quanye Jia	194
Mei Tu	450	R	
Mengmeng Zhu	325	Rachel Edita Roxas	413
Mengxia Zhai	450	Renfen Hu	255,267
Mengxiang Wang	75	Rinaldi Andrian Rahmanda	222
Mengyuan Liu	79	Rodolfo C. Raga Jr	462
Miao Li	166	Rui Cai	177
Mijit Ablimit	468,474	Rui Wang	396
Min Ye	19	Rui-Li Du	343
Min Zhang	325,337,355	Ruijing Wang	166
Minghao Guo	308	Ruiying Geng	456
Mirna Adriani	222	S	
Muhammad Fakhrur Razi Abu Bakar	211	Saihu Liu	379
Muzammil Khan	1	Sanath Jayasena	205
N		Sang Ha Kim	450
NYIMA Tashi	424	Sheng-Ge Zhang	343
Nankai Lin	84,96,313	Shengyi Jiang	84,96,313
Ngoc Thang Vu	160	Shichen Liang	255,261,267
Norisma Idris	211	Shihao Qin	199
P		Shijie Wang	498
Pamoda Abeyawardana	205	Shilei Huang	19
Peifeng Li	296,302	Shui Liu	445
Peng Jin	419	Shumin Shi	407
Ping Jian	456	Shuzhou Chai	402
		Sibao Chen	65
		Sihui Fu	84
		Song Liu	450
		Sufeng Duan	396
		Surangika Ranathunga	234

T

Tadahiro Matsumoto	35
Taichi Aida	349
Takumi Maruyama	53
Tao Feng	166
Tao Pan	126
Tao-Hsing Chang	132
Taro Tada	29
Thilini Ranagalage	205
Ting Ruan	343
Tishuang Wang	302

Xuejin Yu	115
Xuemei Tang	255,261,267
Xuemin Duan	384

Y

Uthayasanker Thayasivam	205,228,234
-------------------------	-------------

Ya-nan Li	343
Yadi Li	273
Yajing Yan	319
Yajun Du	419
Yan Li	7
Yan-Miao Song	343
Yanbing Hu	126
Yanchun Zhang	419
Yang Wei	108
Yang Xu	355
Yang Yuan	154
Yanlu Xie	121,172,308,504

W

Wei Huangfu	115
Wei Li	343
Wei Liu	343
Wei Wei	177
Wei Zhang	172
Wei-Qiang Zhang	402,428
Wei-wei Liu	343
Wenjie Peng	172
Wenwei Dong	121,504
Wuying Liu	136

Yidong Chen	424
Ying Chen	492
Ying Li	290
Ying Yin	343
Ying-Xin Gan	343
Yingjie Zhao	434
Yingxue Zhang	456
YinhuaHai	284
Yiqing Zhang	182
Yohan Karunanayake	234
Yong Cuo	424
Youichiro Ogawa	23

X

Xia Li	199
Xiao Song	19
Xiaodong Shi	424
Xiaopeng Wei	154
Xiaotian Xing	319
Xiaoyong Lu	126
Xin Chen	337
Xingyuan Chen	419
Xuejie Zhang	90

Yu Hong	325,337,355,367
Yu Sun	355
Yu-Bin Huang	343
Yuanbin Wu	390
Yue Wu	166
Yueming Du	71
Yueshu Zhao	384
Yuguang Meng	142
Yumeto Inaoka	41
Yusha Zhang	313
Yuting Song	279

Z

Zhanting Bu	361
Zheng Tan	154
Zhensong Li	379
Zhenye Yang	402
Zhixiang Luo	379
Zhiyi Wu	7
Zhiying Liu	255,267
Zhuanzhuan Ji	504
Zhuosheng Zhang	182
Zijia Han	373
Zitao Chen	96
Zonglin Liu	373

Quantifying the Use of English Words in Urdu News-Stories

Mehtab Alam Syed¹, Arif Ur Rahman², Muzammil Khan³

¹*Department of Computer Science, Free University of Bozen-Bolzano, Bolzano, Italy*

²*Department of Computer Science, Bahria University, Islamabad, Pakistan*

³*Department of Computer Science, University of Swat, Swat, Pakistan*

msyed@unibz.it, badwanpk@bui.edu.pk, muzammilkhan86@gmail.com

Abstract—The vocabulary of Urdu language is a mixture of many other languages including Farsi, Arabic and Sanskrit. Though, Urdu is the national language of Pakistan, English has the status of official language of Pakistan. The use of English words in spoken Urdu as well as documents written in Urdu is increasing with the passage of time.

The automatic detection of English words written using Urdu script in Urdu text is a complicated task. This may require the use of advanced machine/deep learning techniques. However, the lack of initial work for developing a fully automatic system makes it a more challenging task. The current paper presents the result of an initial work which may lead to the development of an approach which may detect any English word written Urdu text. First, an approach is developed to preserve Urdu stories from online sources in a normalized format. Second, a dictionary of English words transliterated into Urdu was developed. The results show that there can be different categories of words in Urdu text including transliterated words, words originating from English and words having exactly similar pronunciation but different meaning.

Keywords-Urdu Text Processing, Urdu Transliteration to English

I. INTRODUCTION

Urdu is the national language of Pakistan and is spoken in many countries. Estimates show that Urdu is spoken by 164 million people around the world [1], [2]. The word “Urdu” is basically taken from Turkish language and it means “army”. The vocabulary of Urdu language contains words from many languages including Persian, Arabic, Turkish, Sanskrit, and Hindi.

The issue of using Urdu as official language is raised in Pakistan every once in a while. Once even the supreme court of Pakistan ordered to use Urdu in official communications. However, the task of language switching is very challenging as people almost do not know the pure Urdu in daily life. People mostly use a mix of Urdu and English in daily life conversations. The reasons of not knowing the words is the use of a mix of Urdu-English language in daily life.

It is a challenging task to write words which are borrowed from other languages (loan words) with different alphabets and sound inventories. Typically, loan words are transliterated, i.e. replaced with approximate phonetic equivalents. Some approaches have already been developed for automatically transliterating Japanese to English, English to Japanese, Arabic to English and English to Arabic [3], [4]. Though, Google has developed a tool for English to Urdu transliteration, our analysis shows

that it is still in its initial stages. Moreover, there are no tools available to automatically detect English words transliterated to Urdu in Urdu text. The tools developed as a part of the current work uses a lexicon based approach for detection of English words transliterated in Urdu. This can prove to be a starting point for developing approaches using more sophisticated techniques like neural networks e.g. Long Short Term Memory(LSTM) networks.

II. RELATED WORK

Digital preservation can be understood as ‘the ability to sustain the accessibility, understandable and usability of digital objects in the distant future regardless of changes in technologies and in the designated communities’ [5]. The data which is needed for long term plays an important role in our daily life. These data is to be preserved in an organized format so that it will be retrieved whenever it is need. The data which is preserved can be organized using time or other parameters which is important according to the perspective [6].

Long-term preservation approaches have comprised of emulation, migration, normalization, and metadata or some combination of these. Most existing work has focused on applying these approaches to digital objects of a singular media type: text, images, databases, video or audio. There is also a need to consider the preservation of composite, mixed-media digital objects which is a rapidly growing category of resources. It describes an integrated, flexible system that is to be developed, which leverages existing tools and services and assists organizations to dynamically discover the optimum preservation strategy as it is required. The system captures and periodically compares preservation metadata with software and format registries to determine those objects at risk. By making preservation software modules available as Web services and describing them semantically using a machine-processable ontology, the most appropriate preservation service for each object can then be dynamically discovered, composed and invoked by software agents. The growing array of available preservation tools and services can be integrated to provide a sustainable, collaborative solution to the long-term preservation of large-scale collections of complex digital objects [7].

There are other approaches available which proposed to look at, inspect, and report the stream of news data, its content, and information for four noteworthy daily papers

from generation and sourcing, through editing and printing, to dissemination to end clients [8]. It was the creator's trust that their report may pinpoint the high impact factors in the work process where libraries and other memory associations could catch basic news substance and metadata; and guarantee the long awaited survival and availability of the American journalistic record. The outcomes, however as anyone might expect, indicated little consistency among work processes. The results also indicated that a high level of standardization would be vital for fruitful preservation of repositories. There are other studies available which link news stories in the process of preservation [9]. The identification of transliterated words can also be helpful in linking news stories.

There are three unique techniques to capture online news and other web resources, namely by authoring system, by browser and by web crawler [10]. The web crawler approach is followed to preserved the digital news to build the archive. Web crawler is a program which analyze and extract desired information from the web page or pages.

The other aspect of the research after building the archive is to find the transliterated English words in Urdu news archive. The dictionary meaning of translation is the process of changing something that is written or spoken into another language, whereas transliteration is to write or describe words or letters using letters of a different alphabet or language [11].

Transliteration process is the replacement of words from source language to the approximate phonetic or spelling equivalents in the target language. Transliterating names between languages that use similar alphabets and sound systems are very simple, since the phrase mostly remains the same. The transliteration becomes difficult when transliterating between languages having different sound systems and having different writing system [12].

It is challenging to translate all the words or lexicons across languages with different sounds and phonetics. These words or lexicons are mostly transliterated from source language to the target language with its approximate sounds and phonetics. For example, "computer" in English comes out as "konpyuutaa" in Japanese [13]. Same phonetics words or lexicons used in different languages worth a lot because of it gives an interesting results about the different origin words are used across different languages.

A. Translation vs. Transliteration

Translation is the process of translated words or text from one language to another language. It gives the equivalent semantic meanings in the destination language in which it is to be converted. Translation is the spoken word or text meaning of a word in source language and translated in the destination language. e.g. To translate the English language words into Urdu language 'Word' to لفظ, 'Sentence' to جملہ respectively.

Transliteration is converting the text from one language to another and does not render meaning. For example the English word 'break' is very frequently used

in Urdu and written as 'بریک', 'make' as 'میک', and 'cake' as 'کیک' respectively.

III. CORPUS BUILDING

Though there are some corpora available which could be used but they lack certain information and are not openly available. The purpose of building a new corpus is that the work could be easily taken forward after properly annotating the tokens. The annotating of tokens is a basic requirement for developing more sophisticated approaches i.e. using deep learning.

A. Workflow for Archive Building

The first step is to identify the sources which are used for building the archive of News stories. Three News sources are selected for building the archive which include popular online Urdu newspapers.

The first step is to analyze the 'HTML' source of each News source, identify the URLs of in 'href' tags which contain the News story. Identify the HTML tags which carry the information about the news story in each news story web page. Once analyzing the News web story is completed the data from the news is extracted and converted to the normalized format and thus preserved. This task is completed using a different extractor for each source which takes into account the specific structure of the HTML of the source. A normalized format is a standard format which can be accessed using standard tools and technologies. The story and its related information is further categorized into metadata and the story itself with hash comparison. The metadata is then categorized into 'explicit metadata' and 'implicit metadata'. Explicit metadata is directly available in with the story in HTML tags and includes information like title, publishDate, category, description, format, keywords, and language. Explicit metadata parameters are identified, extracted and mapped on Dublin core metadata standard. Implicit metadata on the other hand needs to be extracted from the title and story. The title and story are tokenize so that all the words in the story and title are separated and its frequency of occurrence is calculated and added in the implicit metadata respectively.

Individual words are referred to as Tokens (at least in languages like English) and tokenization is taking a text or set of text and breaking it up into its individual words. Tokenization is usually done for various tasks like finding frequencies of words. In Urdu language the tokenization is different from English as in English language the words are separated with a space in between them but in Urdu language two words may or may not need to be separated using a space character e.g. بڑا میز - are two words without a space between them. The tokenization for building the archive has some incorrect which have some error rate while finding the transliterated words in the archive.

Metadata is stored in `metadata.xml`, story is stored in `story.xml` and hash is calculated for comparison is stored in `hash.xml` respectively.

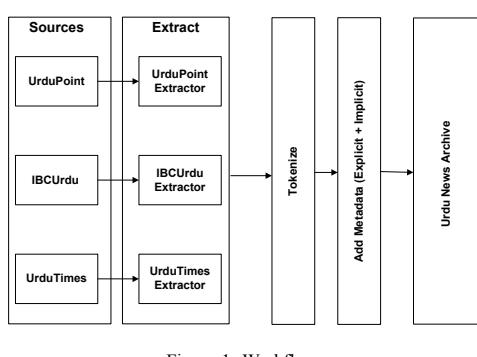
1) Hash Comparison and Issues: The hash is calculated for the story and metadata digital objects to identify if a previous version of the story is already stored in the archive. Hash is calculated for the story and metadata is compared with the `hash.xml` of all stories.

- The calculated hash should ignore the `storyId` as the `storyId` will be unique for all the stories.
- If the calculated metadata hash of the story is same with any story metadata hash in `hash.xml` and the calculated hash of story is different from story hash in `hash.xml` then a new version ‘v1’, ‘v2’ or so is updated in the archive.
- If the calculated story hash of the story is same with any story hash in `hash.xml` and the calculated hash of metadata is different from metadata hash in `hash.xml` then a new version ‘v1’, ‘v2’ or so is updated in the archive.

The complete workflow of building the archival of Urdu news stories is presented in figure 1.

The issue with Urdu language is that there is a very small number of sources from where digital-born text can be collected. Even the digital version of the most famous Urdu dictionary i.e. Ferozul Lughat (فیروز لughat) is a scanned version of the printed book and does not allow to search words using latest searching techniques. The issue with the scanned versions of books is that Urdu OCR technology is still very weak and does not perform good in many situations [14], [15]. Each story is a combination of metadata, some associated files and the text of story.

The metadata is stored in a `metadata.xml` and includes story ID, author name, publication date & time, category, keywords and subject. All the metadata is taken automatically from the source except the Story ID. The Story ID is composed of the publication date and time, a random number and version number e.g. 2016051600001347300507v0. A default date and time i.e. 000101010000 is added in case the publishing date and time is not known. It is assigned to each story at the time of archival. Sometimes stories may be updated after publishing them. In such cases, a second version of the same story is archived. The categories include sports, showbiz, poetry, cooking and horoscope. The analysis of usage of words (English or Urdu) is done on the basis of



category. The associated files are stored in a folder and may include images, audio and video included in a story. The `story.xml` file includes the text of story which is divided into paragraphs.

IV. LEXICON BUILDING

A manual mechanism was used to build the lexicon which was further used for transliterated words in the Urdu news archive. A tool is created for adding the lexicon from the text which is provided as input. It will show the lexicon one by one to manually add by user input as if the word is an Urdu word then it will be added as in `UrduDictionary.xls`, Similarly as if the word is a transliterated English words it will be added as in `UrduEnglishDictionary.xls`, On the other hand the ignored words are added in `otherDictionary.xls` respectively. More than 2100 transliterated English words are added in `UrduEnglishDictionary.xls` which is further used to process the build archive.

The corpus was used to build two lexicons i.e. an Urdu and English. This may seem a bit trivial but this is the first attempt of this sort and helped in identifying some issues which may need to be taken care of while developing an Urdu lexicon through an automated process.

The Urdu lexicon contains all valid Urdu words in the corpus. The following rules are followed.

- Nouns are included without considering language.
- Some words may be written in multiple form e.g. رحمان and رحم (the same words written in two different forms).
- There are some words which are neither English words nor Urdu words. However, these words are derived from English words and are frequently used in Urdu. Table I presents some examples of such words. These words are included in the Urdu lexicon.
- There are some words in Urdu and in English which has the same pronunciation like the words presented in table II. These words are added in the Urdu lexicon.

A second lexicon is developed i.e. English/Urdu lexicon which contains transliterated English words written in Urdu e.g. challenging and archive can be written as ارکئچنگ and چیلنجنگ respectively. The following rules are followed for building the lexicon.

TABLE I. URDU WORDS HAVING ENGLISH ORIGIN

English	Urdu	English	Urdu
scheme	اسکیمیں	agent	ایجنس
agenda	ایجندے	agency	ایجنسیاں
community	کمیونٹیوں	ambulance	ایمبولنسوں

TABLE II. WORDS WITH SIMILAR PRONUNCIATION

English	Urdu	English	Urdu
eye	ائی	kiss	کس
server	سرور	cone	کون
peace	پیس	same	سم

- English text written using English letters is ignored.
- English homophones e.g. lift and left, are written the same in Urdu as presented in table III. Therefore, one of the English words is listed in the English lexicon.

V. CORPUS ANALYSIS

The tool for building the corpus is completed and only in the past few days a corpus which contains more than six hundred stories was developed. The contains a total number of:

- Tokens in the corpus: 117393
- Unique tokens in the corpus: 10914
- Urdu Words in the corpus: 101147
- English Words in the corpus: 9962
- Unique Urdu Words in the corpus: 7770
- Unique English Words in the corpus: 1038

The table IV presents further statistics about the corpus. The size of the corpus will grow with the passage of time and these statistics may be affected by the change in size of the corpus. The last column in the table presents the number of ‘Other Words’ which may be unrecognized words. The unrecognized words include words which are not properly tokenized and words of languages other than English & Urdu.

A. Tokenization and its issues

Tokenization is the initial step for every language processing tasks e.g. machine translation, sentence ending detection, information retrieval and information extraction and preservation of corpus from which information is to be extracted for the long term which is further to be used by different standard tools. In most of the Asian languages the space is to be used after each character placement which is to be omitted to form the token as token consist of each different single word. Joiners are the characters which can occupy the initial, medial or final forms in the word.

TABLE III. WRITING HOMOPHONE IN URDU

Urdu	English	English
لفت	lift	left
مین	man	main
ہیٹ	hat	hate
کیپس	hat	hate
کیریئر	career	carrier
نیس	niece	Nice (city)

TABLE IV. CATEGORY WISE STATISTICS

Category	A	B	C	D
Sports	243	2674	487	1004
Showbiz	143	1677	286	598
Cooking	100	487	66	154
Horoscope	38	288	17	22
Kids	36	1338	17	86
International	23	802	132	207
Women	18	399	27	23

A: Number of Stories

B: Number of Distinct Urdu Words

C: Number of Distinct English Words

D: Number of Distinct Other Words

Tokenization issues may be of various types. For example some English words cannot be separated from another Urdu Word which means the token being analyzed is basically a combination of an Urdu word and an English word like اولپیٹز کیساتھ and کیریئر کا, ڈیزائنر میں کیروالیفائی. Moreover, there may be tokens which are a combination of a word and a number. Additionally, some tokens may be a combination of an Urdu word and a word written using English letters.

Tokenization is a step required for applying other algorithms for accurate NER and POS tagging. However, the lack of an efficient tokenizer make it even difficult to process Urdu text for any purpose. The following issues make Urdu NER a challenging task. The examples of Urdu text ambiguities are presented in Table V.

- 1) **Punctuation Ambiguity:** Urdu language has punctuation ambiguity as punctuation marks are also used to specify the range, inside date, acronyms and abbreviations [16]. NER also used to extract the dates from the text, when it extract the date which separates the day, month and year with (-), it creates confusion because (-) is used for sentence termination. Therefore, smart techniques are required for accurate NER.
- 2) **Diacritics:** Some Urdu writers may use diacritics on some words which may change the word. For example حسین is a noun but حسین is an adjective.
- 3) **Word Ending with Non-connectors:** Urdu characters are divided into two groups i.e. connector and non-connector. Whenever a word ends with a non-connector, there is no need to insert space, as native speaker can easily differentiate it from other word [2], [16]. For example قارداد, قارداد پاکستان and پاکستان are two words with no space between them. Space is not inserted between words as the word قارداد ends with a non-connector which makes it difficult to extract پاکستان as location name. Therefore, such situations will require special techniques to separate the words for accurate NER.
- 4) **Sentence Boundary Ambiguity:** A native speaker can easily define the boundary of sentences by considering the context of words [2], [16]. It is complicated to identify the end of a sentence due to punctuation ambiguity and no capitalization. NER will fail to extract the relationship of the entities as well as the position of occurrence of the entities at sentence level if sentences are not segmented properly.
- 5) **Sentence Segmentation** The sentence terminator (-), in Urdu text is a confusing punctuation mark. Moreover, in Urdu text, a sentence is usually followed by dash (-) to terminate the sentence, which is not consistent. Most of the time, sentence also terminated with other word such as mark of exclamation (!), question mark (-), ellipsis (...) and bullets. Therefore, some techniques need to be developed for Urdu sentence segmentation. The technique to be

- developed may be based on existing approaches for other languages.
- 6) **Order of Words:** Typically in Urdu, subject, object and verb sequence is used to make sentences but sometimes the order of words does not matter [16], [17]. The position of words can change the meaning of a sentence in Urdu. Therefore, it may be important to consider the whole context in which words are used to accurately identify named entities.
 - 7) **Spaces Ambiguity:** Urdu language has agglutinative property which may lead to ambiguous spaces between words. Following are the scenarios in which space should be neglected.
 - 8) **Word Segmentation:** Word Segmentation process needs to determine the boundaries of the words. In Urdu language the word is not separated by space so it is difficult to measure the boundaries of the word in Urdu language.
 - 9) **Compound Words and Suffixation:** In Urdu words, two nouns, verbs and an adjective can be combined with or without the use of و, اور to produce a single semantic word. Additionally, a suffix also generates words which may convert common nouns to proper nouns. If حسین و جمیل is considered a unit the meaning of the word is beautiful while حسین and جمیل are the names of the persons. The NER will consider حسین and جمیل two separate entities while it is a unit when و appears between these two words which is not an entity.
 - 10) **Reduplication:** Reduplication of words is divided into two categories i.e echo and full, which may lead to white space between words. In echo reduplication one set of words is deviated where as in full reduplication a word is repeated twice. In a sentence فیصل .فیصل مسجد جائے کا produces the reduplication of words while فیصل is the name of a person and فیصل مسجد is the name of the mosque. In this case NER could not be able to extract the entities properly.
 - 11) **Loan Words:** Urdu language has borrowed many words from other languages. The loan words should be considered single semantic words.
 - 12) **Nouns Ambiguity:** In Urdu language, nouns ambiguity may occur when proper nouns are used as common nouns. For example, identical other language words and person name. Also, most of the proper nouns have a white space between them, which should be neglected. NER will not be able to differentiate between common and proper nouns which may lead to false positive.

VI. Conclusions and Future Work

There are basically two motives for developing the corpus. First, to archive the news in a normalized format which allows easy access in the future. Second, to analyze the corpus for detection of English words transliterated in Urdu and get an idea of how many English words are used in Urdu. The size of the corpus is definitely a factor

which may influence the frequency of words. Moreover, if a corpus contains news from a specific category then the set of frequent may also be different. However, the presented work is a step towards developing an automatic approach for developing a corpus and identifying English words in the corpus.

Further research is required to develop an approach which does not use dictionaries and can extract English words from a document using modern information retrieval and deep learning techniques. Moreover, the corpus will need to be properly annotated before it can be used to develop neural networks for the same task.

REFERENCES

- [1] M. Humayoun, H. Hammarström, and A. Ranta, "Urdu morphology, orthography and lexicon extraction," *Chalmers University of Technology, Master*, 2006.
- [2] Z. Rehman, W. Anwar, and U. I. Bajwa, "Challenges in Urdu text tokenization and sentence boundary disambiguation," in *Proceedings of the 2nd Workshop on South and Southeast Asian Natural Language Processing (WSSANLP 2011)*. Citeseer, 2011, p. 40.
- [3] K. Knight and J. Graehl, "Machine transliteration," *Comput. Linguist.*, vol. 24, no. 4, pp. 599–612, Dec. 1998. [Online]. Available: <http://dl.acm.org/citation.cfm?id=972764.972767>
- [4] N. Habash, A. Soudi, and T. Buckwalter, *On Arabic Transliteration*. Dordrecht: Springer Netherlands, 2007, pp. 15–22. [Online]. Available: http://dx.doi.org/10.1007/978-1-4020-6046-5_2
- [5] S. Rabinovici-Cohen, R. Cummings, and S. Fineberg, "Self-contained information retention format for future semantic interoperability." in *SDA@ JCDL/TPDL*, 2014, pp. 4–15.
- [6] Muzammil Khan and Arif Ur Rahman, "A Systematic Approach Towards Web Preservation," *Information Technology and Libraries*, vol. 38, no. 1, pp. 71–90, 2019.
- [7] S. Ross, "Digital Preservation, Archival Science and Methodological Foundations for Digital Libraries," *New Review of Information Networking*, vol. 17, no. 1, pp. 43–68, 2012.
- [8] J. Alverson, K. Leetaru, V. McCargar, K. Ondracek, J. Simon, and B. Reilly, "Preserving news in the digital environment: Mapping the newspaper industry in transition," *crl.edu*, 2011.
- [9] M. Khan, A. Ur Rahman, and M. Daud Awan, "Term-based approach for linking digital news stories," in *Digital Libraries and Multimedia Archives*, G. Serra and C. Tasso, Eds. Cham: Springer International Publishing, 2018, pp. 127–138.

TABLE V. URDU TEXT AMBIGUITIES

Punctuation Ambiguity	کو قرارداد پاکستان منظور ہوئی	23-03-1940
Order of Words	کبھی بھی میدان سے باہر چل گئے	
Compound words	مادر ملت و نزد خارجہ	
Reduplication	چلے چلے ، کبھی کبھی	
Loan Words	چک اؤٹ ، سمارٹ فون	
Nouns	فیصل مسروں	
Bidirectional	علامہ اقبال ۹ نومبر کو پڑا ہوئے	23-03-1940
	کو قرارداد پاکستان منظور ہوئی	

- [10] S. Farrell, “A guide to web preservation,” *UKOLN / ULCC*, 2010.
- [11] K. Regmi, J. Naidoo, and P. Pilkington, “Understanding the processes of translation and transliteration in qualitative research,” *International Journal of Qualitative Methods*, vol. 9, no. 1, pp. 16–26, 2010.
- [12] Y. Al-Onaizan and K. Knight, “Machine transliteration of names in arabic text,” in *Proceedings of the ACL-02 Workshop on Computational Approaches to Semitic Languages*, ser. SEMITIC ’02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 1–13. [Online]. Available: <http://dx.doi.org/10.3115/1118637.1118642>
- [13] K. Knight and J. Graehl, “Machine transliteration,” *Comput. Linguit.*, vol. 24, no. 4, pp. 599–612, Dec. 1998. [Online]. Available: <http://dl.acm.org/citation.cfm?id=972764.972767>
- [14] S. T. Javed, S. Hussain, A. Maqbool, S. Asloob, S. Jamil, and H. Moin, “Segmentation free nastalique Urdu OCR,” *World Academy of Science, Engineering and Technology*, vol. 46, pp. 456–461, 2010.
- [15] N. Sabbour and F. Shafait, “A segmentation-free approach to Arabic and Urdu OCR,” in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2013, pp. 86 580N–86 580N.
- [16] K. Riaz, “Rule based named entity recognition in Urdu,” in *Proceedings of the 2010 Named Entities Workshop*, 2010, pp. 126–135.
- [17] S. Naz, A. I. Umar, S. H. Shirazi, S. A. Khan, I. Ahmed, and A. A. Khan, “Challenges of Urdu Named Entity Recognition: A Scarce Resourced Language,” *Research Journal of Applied Sciences, Engineering and Technology*, vol. 8, no. 10, pp. 1272–1278, 2014.

A New Method of Tonal Determination for Chinese Dialects

Yan Li

School of Foreign Languages,
Shaanxi Normal University,
Xi'an, China
e-mail: 2373515155@qq.com

Zhiyi Wu

Department of Psychology,
Carnegie Mellon University,
Pittsburgh, USA
e-mail: wuzhiyi.jenny@outlook.com

Abstract-Values of the basic tones are the key to do research on dialects in China. The traditional method of determining tones by ear and the more popular method used in experimental phonetics are either inaccurate to some degree or difficult to learn. The method provided and discussed in this paper is simple and reliable, requiring the use of only Praat and fundamental frequency value. More examples are given to prove this method's effectiveness.

Keywords-tonal determination; fundamental frequency value; Chinese dialects

I. INTRODUCTION

Tonal languages like Mandarin “generally use tones to represent phone level distinctions” [1]. Other than the official language called mandarin, there are many derivational languages distributed around China which are labeled as dialects. These dialects have high value in understanding the language development and diversity [2]. As a critical part of the dialects, basic tones need to be determined in order for further study. To understand the phonological features of a Chinese dialect, getting the basic tones is very fundamental and important.

II. TONAL DETERMINATION

The tonal studies have been more easily marked since Chao YR developed the five-degree system using tone numbers: pitch height with 5 being the highest and 1 being the lowest so it corresponds to the height and the movement. However, crucial as it is, the studies on Chinese dialects are still not advanced and unified enough to construct a more complete system. One reason lies in the difficulty of accurately determining the pitch heights of basic tones in each dialect. In the past few centuries, scholars relied mostly on their ears to distinguish among different tones in a diversity of dialects. Due to the differences among the auditory processing of individuals, the pitch heights of the basic tones in dialects tend to be numbered in more than one way. For example, sixteen different numbering systems have been developed for Lanzhou dialect by sixteen groups of scholars, who all determined the tones by ear [3]. Gladly, technology improvement has proved to compensate for the bad result caused by the subjectivity of human perception. As experimental linguistics advances, we now are able to normalize the “between-speaker (B-S) acoustical variance caused by physiological differences” among different individuals using different types of normalization methods

so that the invariant acoustic information can be extracted and specified [4]. Among all such methods, Chinese scholars favor D-value method, T-value method, and frequency domain quantile method more than the others [5][6]. Three studies analyzing Chongqing dialect with T-value method yielded the same basic tone values, which were different from those that were determined in the traditional way [7][8].

Although fundamental frequency method seemed promising and infallible for a while, undesirable results have arisen in many experiments regarding dialects. According to Feng Shi, “to investigate the tones, we should investigate the syllables with the same initial and final, and in the same place of the flow of speech.” [9] However, the flow of speech is difficult to control. Take the tonal determination of Ziyang dialect as example and apply simplified fundamental frequency graph to determine the tone. There are 4 tones in this dialect, we choose one character in each tone group, they are east “东”, door “门”, bitter “苦”, and ache “痛” to represent yinping, yangping, shangsheng and qusheng respectively, then get fundamental frequency value of each character through Praat (see Figure 1, 2, 3, 4). East “东” has a rising tone, fundamental frequency is between 139.8 and 197.1 Hz, door “门” has a falling tone, fundamental frequency is between 178.1 and 88.8 Hz, bitter “苦” has a rising tone, fundamental frequency is between 225 and 97.3 Hz, ache “痛” has a falling rising tone, fundamental frequency is from 133.8 to 128.4 to 206.2. From the fundamental frequency graph, we can get the detailed tones, here we still use 5-degree method (see Figure 5).

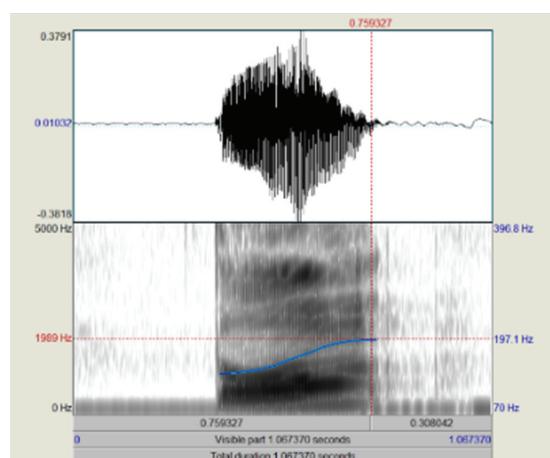


Figure 1: The Fundamental Frequency of East (“东”)

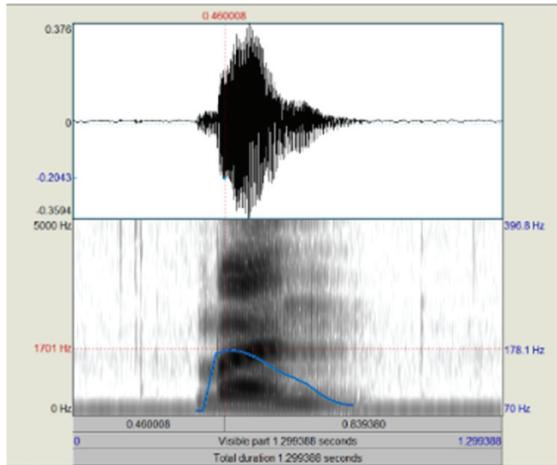


Figure 2: The Fundamental Frequency of *Door* (“门”)

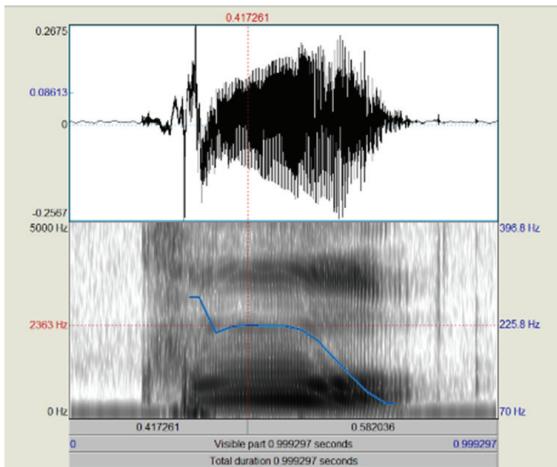


Figure 3: The Fundamental Frequency of *Bitter* (“苦”)

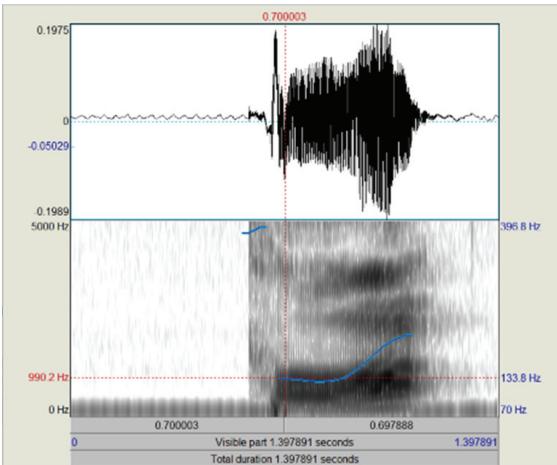


Figure 4: The Fundamental Frequency of *Ache* (“东”)

Blue represents yinping east “东”, red represents yangping door “门”, grey shangsheng bitter “苦”, yellow qusheng ache “痛”. From Figure 5 we can see that yinping and qusheng starts from 2 and ends in 4 but qusheng has a little concave curve, we use 22 to express. Thus yinping is 24, yet qusheng is 224, yangping and shangsheng are both falling tones, yangping starts from 4, shangsheng from 5,

both end in 1, so yangping is 41, shangsheng is 51. According to the fundamental frequency graph of these four characters, the tone value should be: yinping 24, yangping 41, shangsheng 51, qusheng 224.

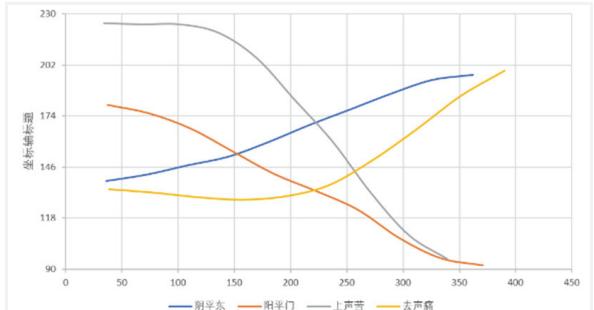


Figure 5: The Fundamental Frequency Curve Graph of “东”, “门”, “苦”, “痛”

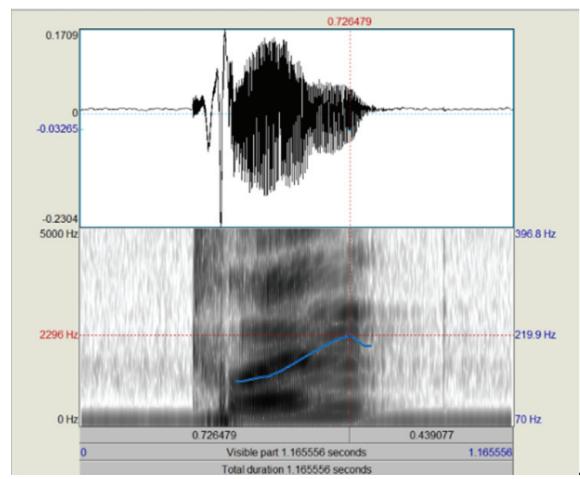


Figure 6: The Fundamental Frequency of *Spring* (“春”)

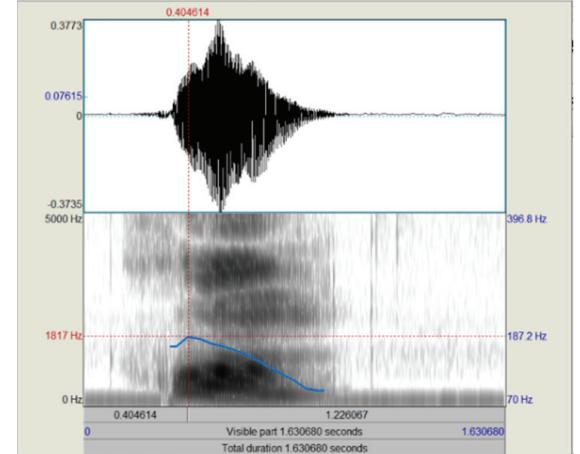


Figure 7: The Fundamental Frequency of *Red* (“红”)

However, if we choose another group of characters, what will happen? Will they have the same tone as previous group? Here we choose spring 春, red 红, old 老, earth 地. Their values in Praat are (see Figure 6, 7, 8, 9): spring’s fundamental frequency is between 145 and 219.9 Hz, red 186.1 and 97.3 Hz, old 209.7 and 105.6 Hz, earth

131.5, 119.7 and 196.6 Hz. According to fundamental frequency graph (see Figure 10), their tones are yinping 25, yangping 41, shangsheng 51, qusheng 114 respectively. They are not the same as the previous group, especially yinping and qusheng.

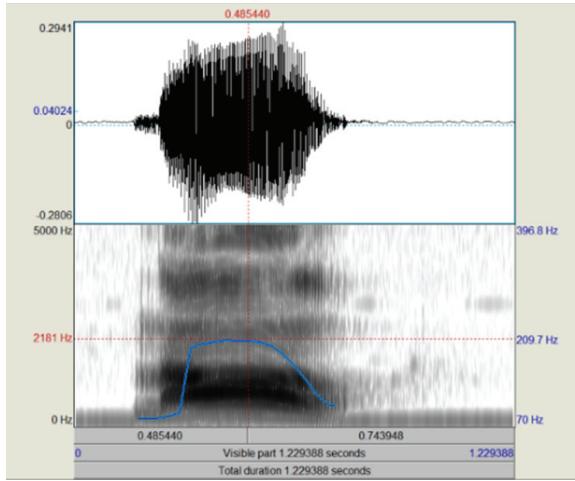


Figure 8: The Fundamental Frequency of *Old* (“老”)

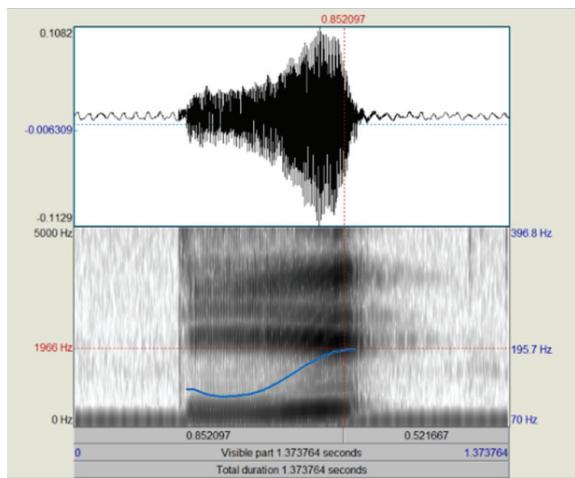


Figure 9: The Fundamental Frequency of *Old* (“地”)

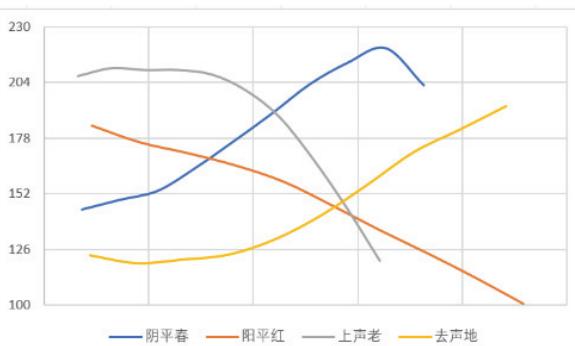


Figure 10: The Fundamental Frequency Curve Graph of “春”, “红”, “老”, “地”

Now we get more characters from each tone group, 7 characters from yinping group: should 该, east 东, light 灯,

wind 风, through 通, open 开, sky 天, 9 from yangping group: door 门, dragon 龙, oil 油, copper 铜, hundred 百, joint 节, worry 急, cry 哭, pat 拍, 9 from shangsheng group: understand 懂, ancient 古, ghost 鬼, nine 九, unite 统, bitter 苦, buy 买, five 五, have 有, 10 from qusheng group: sell 卖, road 路, hard 硬, chaos 乱, hole 洞, earth 地, meal 饭, tree 树, freeze 冻, odd 怪, then get the average value, according to average value we get the fundamental frequency curve graph (see Figure 11). From this figure the tone value should be 24, 41, 51, 224, quite different from the previous two (see Table 1, 1, 2, 3 represents three different results respectively).

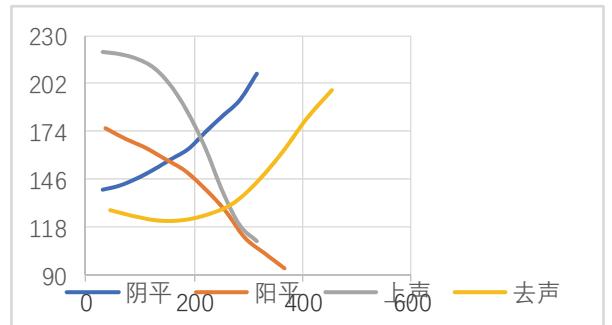


Figure 11: The Fundamental Frequency Curve Graph of Ziyang Dialect

Table 1: Different Tone Values from Different Groups of Characters

	yinping	yangping	shangsheng	qusheng
1	24	41	51	224
2	25	41	51	114
3	25	31	51	224

From the above discussion we can see that even if we use fundamental frequency graph, the tone value is not stable. The character example is the key point. When we investigate the tones of a character or a word, the informant couldn't possibly produce the same tone every time even if they belong to the same tone group, e.g., shangsheng group in Shuanghe town dialect in Ziyang County, the fundamental frequency of fire 火 is between 211-167 Hz, write 写 251-98 Hz, tile 瓦 201-184 Hz, tiger 虎 244-192 Hz. All these characters belong to shangsheng group, but their fundamental frequencies are quite different, the highest point of the beginning reaches 252 Hz, the lowest is 201, the ending part is from 192 Hz to 98 Hz. If we depend on the fundamental frequency graph, it's hard to get the same tone value. Now turn to another person's pronunciation, this time we take yangping in Huaping dialect in Ankang City as an example, eggplant 茄 171-120Hz, grandma 婆 153-93 Hz (in the middle there appears a creaky voice), tooth 牙 174-55 (also appears a creaky voice), except 除 163-125 Hz, fish 鱼 155-114 Hz, complete 齐 164-113 Hz, compensate 赔 179-119 Hz. They all belong to the same tone group, but their tones are not the same with the starting point from 179 Hz to 153 Hz and the ending point from 125 to 55. If we get about 10 characters

from each tone group and the average value from the fundamental frequency graph, there appears one problem, the duration of each word is not the same. Even if we have the average value, the duration time is hard to control.

The traditional method by ear is not reliable, and the phonetic experimental study on dialectal tones has more or less disadvantages and not credible. In addition, experimental phonetic method needs special training. Hence finding an objective, reliable, practical and convenient way to mark the tone is inevitable.

III. THE STEPS OF DETERMINING THE TONES

How to determine the tone exactly is an aporia for many scholars. Thus we try many ways to make sure that one way is more accurate and simpler than the others. The steps are as follows:

First, get the fundamental frequency of some representative characters, we can see the highest value and the lowest value clearly (it's better to abandon the characters which contain the highest value and the lowest value in order to avoid the fortuity and extremeness) through Praat software. Then the secondary highest value subtracts the secondary lowest value, we get the remainder. Since five degree system is always applied to mark the tone, the remainder is divided by 5, we get the average value. Add this average value to the secondary lowest tone, e.g., the secondary lowest tone is 95, the average value is 24, then the second number is 119, the third is 143, the fourth is 167, the fifth is 191, the sixth is 215. This time we take Huaping dialect as example.

First step: list the representative characters and fundamental frequency, 8 characters from each group, the more example characters, the better (because of the lack of space, we only get 8. Fundamental frequency unit: Hz (the number after the decimal point is deleted)) (see Appendix 1).

Second step: Find out the highest value and lowest value of fundamental frequency.

The highest value is 272, the lowest value is 100 (here we didn't delete the highest value and lowest value). Then the highest value subtracts the lowest value, we get the remainder 172. 172 divided by 5 is 34.4 (we delete the number after the decimal point and get 35). Then add 35 to the lowest value 100, we get 135, then add 35 to the previous number, we get 135, 170, 205, 240, 275 successively.

Third step: Draw a graph. Make a fundamental frequency graph for the 8 characters in yinping group. Divide the numbers from 100 to 272 by 5, we can see the tones of these 8 characters in Figure 12: low 低 24, open 开 22, should 该 23, book 书 22, beginning 初 34, pig 猪 23, boot 鞠 23, song 歌 23. According to majority rule, yinping should be 23. There is a much simpler method, that is, we don't need to draw a graph, just see which area each character belongs to. We have six numbers: 100, 135, 170, 205, 240, 275, five areas: 100-135, 136-170, 171-205,

206-240, 241-275. So low “低” (167-219 Hz)'s starting point 167 is in area 2: 136-170, ending point 219 is in area 4: 206-240, thus the tone of low “低” is 24.

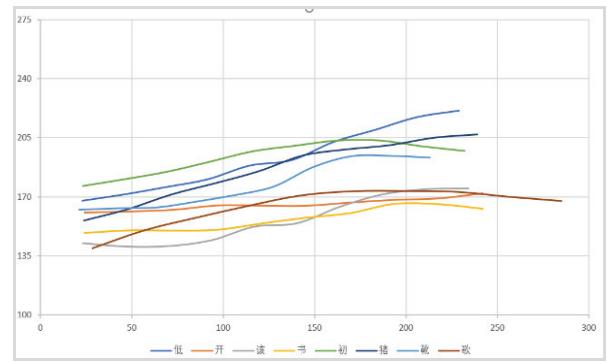


Figure 12: The Fundamental Frequency Graph for the 8 Characters of Yinping Group in Huaping Dialect

Fourth step: Determine the other tones in the same way. The tones of 8 characters in yangping group: compensate 赔 31, complete 齐 21, mud 泥 21, firewood 柴 21, hoe 锄 31, eggplant 茄 32, nose 鼻 21, late 迟 21. 21 is the majority, so yangping should be 21. The tones of 8 characters in shangsheng group: ghost 鬼 54, understand 懂 54, ancient 古 54, nine 九 54, bitter 苦 54, grass 草 43, buy 买 44, old 老 54, therefore shangsheng should be 54. The tones of 8 characters in qusheng group: move 动 224, crime 罪 224, odd 怪 212, ache 痛 224, inch 寸 225, near 近 113, hole 洞 224, tree 树 224, five characters' tone is 224, so qusheng is 224.

So far the four tones of Huaping dialect have been determined, yinping 23, yangping 21, shangsheng 54, qusheng 224. This method is feasible and objective and could avoid subjectivity. In the process of determining the tone, we should obey the following principles: (1) The more example characters, the better. We should get more example characters to make ensure the objectivity. In this paper we just choose 8 characters because of the limitation of space. (2) The majority rule. When a tone value appeared on more characters than the others, we choose this tone value. (3) Principle of balance. Each degree should appear, we can't lack any one of them. (4) Leave out the beginning and the end. We should delete characters with the highest fundamental frequency and the lowest one, in order to avoid extremity and fortuity.

IV. THE CASE STUDY

Let's return to Ziyang dialect. This dialect has been studied by Wang Li[10] and Zhou Zheng[11]. Wang Li[10] thinks Ziyang dialect has four tones: yinping45, yangping31, shangsheng53, qusheng214. Zhou Zheng[11] claims this dialect is close to Cheng Yu dialect, and has four tone categories, but the tone values are not the same as Wang Li's, they are 34, 21, 53, 213 respectively. The tone based on the above discussion about Ziyang dialect

according to a certain character or average value is also doubtful. At the moment we apply the above method to determine the tone value of Ziyang dialect. First choose about 10 characters, then get fundamental frequency value through Praat software (see Appendix 2).

From table 3, we can see that the highest value is 233 Hz, the lowest is 83, the remainder is 150. 150 divided by 5 is 30. According to five degree system, the numbers should be 83, 113, 143, 173, 203, 233. Using the simple method, determine the tones based on which area the fundamental frequency value belongs to. Five areas are: 83-113, 114-143, 144-173, 174-203, 204-233. Hence, the tones of these characters in yinping group are: east 东 24, wave 波 34, light 灯 25, ladder 梯 34, through 通 24, open 开 24, sky 天 35, spring 春 35. Since no tone is in the majority, we should choose more characters. Now we get five more characters in yinping group from the word list in sequence, they are: song 歌 35, boot 靴 34, should 该 23, west 西 34, dust 灰 34. In light of majority rule, yinping is 34. Let's turn to yangping: door 门 31, dragon 龙 31, cow 牛 31, oil 油 31, copper 铜 41, skin 皮 31, joint 节 41, late 迟 31. Most of tones are 31, therefore yangping is 31. Shangsheng: ghost 鬼 51, understand 懂 51, ancient 古 53, nine 九 51, bitter 苦 51, unite 统 51, buy 买 51, have 有 51. Most of tones are 51, so shangsheng is 51. Qusheng: move 动 225, crime 罪 325, odd 怪 224, ache 痛 225, back 后 225, near 近 225, four 四 325, half 半 114, qusheng should be 225. Till now we can determine the four tones of Ziyang dialect: yinping 34, yangping 31, shangsheng 51, qusheng 225.

Look at Zhenba dialect. Both Wang Li[10] and Huang Houju[12] regard it has four tone categories, yinping 45, yangping 31, shangsheng 54, qusheng 324. If we use the fundamental frequency graph, what's the tone value?

First, choose some characters from the word list. Here we choose 8 characters in each group (see Appendix 3).

The highest value in Zhenba dialect is 172Hz, the lowest is 68, the remainder is 103, 103 divided by 5 equals to 20.6, we get 21, then the numbers from the lowest one are as follows: 69, 90, 111, 132, 153, 174, five areas are 69-90, 91-111, 112-132, 133-153, 154-174 respectively. Let's look at the tone values of each character. Yin ping group: more 多 23, song 歌 34, wave 波 34, boot 靴 34, cock 鸡 35, open 开 35, district 区 45, spring 春 34, 34 is in the majority, thus accordingly, yinping should be 34, of course we can find more characters to validate. Yangping: eggplant 茄 32, except 除 41, tea 茶 21, grandma 婆 31, row 排 31, mud 泥 31, complete 齐 31, compensate 赔 31, hence yangping must be 31. Shangsheng: fire 火 53, tile 瓦 32, ancient 古 42, bitter 苦 43, five 五 43, tiger 虎 53, host 主 51, buy 买 21, we have the same numbers of tone 53 and tone 43, but for the purpose of balance, we had better choose 53. Consequently, shangsheng is 53. Qusheng: mill

磨 113, sit 坐 214, pass 过 14, thank 谢 113, anger 怒 224, do 做 224, saw 锯 225, half 半 224, here qusheng should be 224. Now we get the tone values of Zhenba dialect: yinping 34, yangping 31, shangsheng 53, qusheng 224.

By using this method, we can easily figure out each person's tone value and learn all nuances of tone and could avoid subjectivity.

V. CONCLUSION

This article discusses the determination of tones by means of fundamental frequency graphs and South Shaanxi dialects. The method by this way is not only simple and reliable, but also convenient and practical. Traditional method of determining tones by ear is not accurate enough, and the experimental phonetic method is much more difficult and not necessarily so exact. Yet comparing with these two methods, the method mentioned in this paper only needs Praat software and fundamental frequency value. At the same time, the value got by computer is much more objective than by brain. The tone determined by majority rule may avoid instability because one person's articulation could be high and low, long and short which interferes the precision of tone's determination. If the tone is determined by any one character, there will appear countless tone values. And for the complex tone, especial 2-contour tone, if we just rely on our ears, it's hard to decide the exact tone value, but by this method we can distinguish it explicitly.

REFERENCES

- [1] S. B. Davis and P. Mermelstein, "Comparison of parametric representation for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357-366, 1980.
- [2] Z. Bao, "The Structure of Tone," Oxford University Press, 1999.
- [3] C. M. Y., "Tone Sandhi: patterns across Chinese dialects," Cambridge University Press, 2000.
- [4] Y. Li and Z. Xiaonong, "The Tonal Types of Mandarin in the Northwest," *Chinese Study*, 2015. [5] R. Phil. "Considerations in the normalization of the Fundamental Frequency of Linguistic Tone," *Speech Communication*, 1987.
- [6] Shi, Feng and Liang Rongrong, "Collections of Phonetics," Beijing Language Institute Press, 1994. [7] Lin, Tao and Wang Lijia, "The Course of Phonetics," Beijing University Press, 1999.
- [8] Liang, Lei and Meng Xiaolin, "Statistical Analysis of monosyllabic tone of Chongqing dialect," *Proceedings of the 9th Chinese phonetics Conference*, 2009.
- [9] Zeng, Xiaoyu, "Comparative Analysis of 70 Years' Research on Chongqing Dialect Phonetics," *Journal of Chongqing Radio and TV University*, 2013.
- [10] Li, Wang, The Study on Southwest Mandarin at the Junction of Sichuan and Shaanxi. Beijing Language and Culture University Master's thesis, 2008.
- [11] Zhou, Zheng. *The Investigation of Ankang Dialects*. Shaanxi People's Educational Publishing House, 2015.
- [12] Huang, Houji. The Comparative Study on Hanzhong Phonology. Beijing Language and Culture University Master's thesis, 2008.

Appendix 1: 8 Representative Characters and Fundamental Frequency from Huaping Dialect

yinping		yangping		shangsheng		qusheng	
low 低	167-219	compensate 赔	179-118	ghost 鬼	263-251	move 动	148-142-209
open 开	161-179	complete 齐	164-118	understand 懂	243-227	crime 罪	135-135-212
should 该	141-173	mud 泥	150-112	ancient 古	272-259	odd 怪	136-122-160
book 书	149-165	firewood 柴	161-100	nine 九	252-246	ache 痛	148-148-203
primary 初	176-201	hoe 锄	174-115	bitter 苦	272-242	inch 寸	156-155-242
pig 猪	158-204	eggplant 茄	172-148	grass 草	213-190	near 近	129-129-201
boot 靴	162-193	nose 鼻	167-101	buy 买	204-204	hole 洞	151-144-239
song 歌	149-173	late 迟	166-115	old 老	237-232	tree 树	149-149-227

Appendix 2: 10 Representative Characters and Fundamental Frequency from Ziyang Dialect

yinping		yangping		shangsheng		qusheng	
east 东	138-199	door 门	178-91	ghost 鬼	212-102	move 动	138-130-204
wave 波	144-187	dragon 龙	162-94	understand 懂	207-109	crime 罪	150-122-210
light 灯	143-211	cow 牛	167-91	ancient 古	214-161	odd 怪	140-122-200
ladder 梯	158-191	oil 油	161-100	nine 九	215-104	ache 痛	134-128-205
through 通	139-202	copper 铜	184-91	bitter 苦	224-94	back 后	136-135-204
open 开	136-200	skin 皮	173-83	unite 统	218-114	near 近	141-126-203
sky 天	145-213	joint 节	180-89	buy 买	233-118	four 四	146-121-196
spring 春	146-218	late 迟	170-96	have 有	232-107	half 半	117-109-188

Appendix 3: 8 Representative Characters and Fundamental Frequency from Zhenba Dialect

yinping		yangping		shangsheng		qusheng	
more 多	104-111	eggplant 茄	128-91	fire 火	169-126	mill 磨	89-86-113
song 歌	113-133	except 除	136-70	tile 瓦	127-106	sit 坐	92-89-137
wave 波	124-142	tea 茶	90-73	ancient 古	135-96	pass 过	86-134
boot 靴	132-145	grandma 婆	131-77	bitter 苦	146-127	thank 谢	88-80-122
cock 鸡	128-160	row 排	112-78	five 五	150-128	anger 怒	102-100-153
open 开	125-156	mud 泥	121-71	tiger 虎	155-129	do 做	115-98-126
district 区	134-172	complete 齐	112-69	host 主	160-89	saw 锯	101-93-165
spring 春	112-146	compensate 赔	132-80	buy 买	104-79	half 半	109-103-135

Duplicate Question Detection based on Neural Networks and Multi-head Attention

Heng Zhang

*Shanghai Key Laboratory of Trustworthy Computing
East China Normal University
Shanghai, China
51174500154@stu.ecnu.edu.cn*

Liangyu Chen

*Shanghai Key Laboratory of Trustworthy Computing
East China Normal University
Shanghai, China
lychen@sei.ecnu.edu.cn*

Abstract—It is well known that using only one neural network can not get a satisfied accuracy for the problem of Duplicate Question Detection. In order to break through this dilemma, different neural networks are ensembled serially to strive for better accuracy. However, many problems, such as vanishing gradient or exploding gradient, will be encountered if the depth of neural network is blindly increased. Worse, the serial integration may be poor in computational performance since it is less parallelizable and needs more time to train. To solve these problems, we use ensemble learning with treating different neural networks as individual learners, calculating in parallel, and proposing a new voting mechanism to get better detection accuracy. In addition to the classical models based on recurrent or convolutional neural network, Multi-Head Attention is also integrated to reduce the correlation and the performance gap between different models. The experimental results in Quora question pairs dataset show that the accuracy of our method can reach 89.3%.

Keywords-deep learning; multi-head attention; ensemble learning;

I. INTRODUCTION

Duplicate Question Detection(DQD) is an important task of Semantic Textual Similarity(STS) in Natural Language Processing(NLP). It is mainly used to judge whether two problems are similar, so duplicate question detection is a binary classification problem. It is also the basis of many NLP tasks. For example, in automatic Question Answering(QA), by retrieving questions that are semantically equivalent to a question presented by a user, an automatic QA system can answer the user's question with answers of the retrieved questions.

About duplicate question detection, researchers proposed a variety of topic modeling[1], syntactic structure[2] and designed various similarity features based on word embeddings[3][4]. Another approach is to use neural networks such as Feed-Forward Neural Networks(FNN)[5], Convolutional Neural Networks(CNN)[6], Long Short-Term Memory(LSTM)[7], and more complex models[8]. Although the above methods have achieved good results on DQD, each single model has one limitation, that is, it does not match and perform well for all types of texts. Take RNN as an example, RNN is suitable for sequence prediction tasks for texts with arbitrary length, since it uses cyclic joins. However, as the sequence length increases, the more information of the header is forgotten. For example, "Why is argon used in welding?" and "How

is argon used in welding?" are two problems with different words in the header of sentence. Although many words are same, the meanings of two sentences are not equal for the reason that the meanings of "Why" and "How" are different. Therefore, when the RNN is applied, it uses fewer header information and more tail information, which may lead to erroneous prediction.

In this paper, we use ensemble learning with treating different neural networks as individual learners and proposing an integrated mechanism based on a new voting algorithm: Credible Voting. Our voting method can escape from the unstable classification in neural network, that is, ensure classification results same after several rounds of training. In addition to the classical models based on recurrent or convolutional neural network, Multi-Head Attention is also integrated to reduce the correlation and the performance gap between different models. For Quora question pairs dataset, the experimental results show that the classification accuracy of our method can reach 89.3%.

The rest of the paper is structured as follows. In Mixed Multi-head Attention section, we introduce the network structure based on Multi-head Attention. Then in the third section, we describe the common neural network structure in the semantic textual similarity task, which is introduced by Bi-LSTM example. Later we present our new voting algorithm in the fourth section. In the fifth section, we show the experimental process, and the results were analyzed. Finally, we summarize in the sixth section.

II. MIXED MULTI-HEAD ATTENTION

In this section, we describe our mixed multi-head attention model in detail. The framework of the model is illustrated in Figure 1. The model mainly contains three components: encoder layer based on the word embedding, hidden layer based on self-attention, mutual-attention and ResNet[9], and decoder layer based on GlobalMaxPooling[10] and Multi-Layer Perceptron(MLP)[11].

A. Encoding Layer

Like to other textual similarity tasks, we use learned embeddings to convert the input words to vectors. Then we use linear module to convert the matrix into a matrix of dimension d_{model} as the input to the entire neural networks.

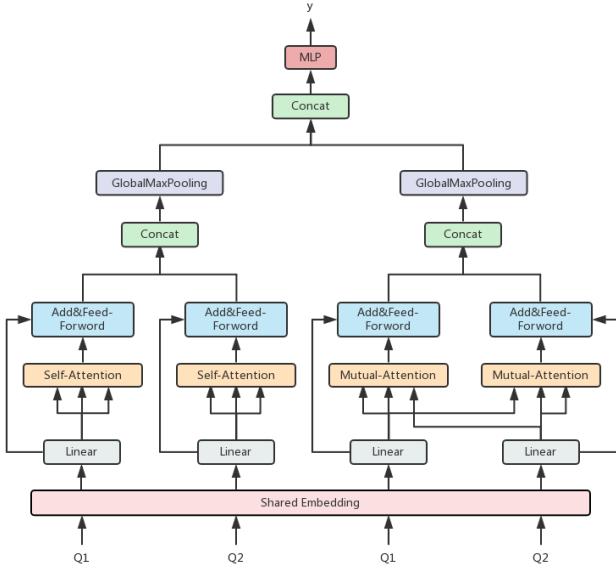


Figure 1: The structure of Mixed Multi-Head Attention

B. Hidden Layer

Hidden layer has two sub-layers. The first is self-attention and mutual-attention, and the second is a simple, ResNet and feed-forward network.

Self-attention and mutual-attention are two variants of multi-head attention, where the difference between them is input vector. Figure 2 depicts the computation graph of multi-head attention mechanism. The attention function can be described as follows:

$$\begin{aligned} Att(\mathbf{Q}, \mathbf{K}, \mathbf{V}) &= softmax\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}, \\ head_i &= Att(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V), \\ \mathbf{Q} &\in \mathbb{R}^{n \times d_k}, \mathbf{K} \in \mathbb{R}^{n \times d_k}, \mathbf{V} \in \mathbb{R}^{n \times d_k}, i \in h \end{aligned} \quad (1)$$

where \mathbf{Q} , \mathbf{K} and \mathbf{V} represent the attention query matrix, key matrix and value matrix respectively, n and d_k represent the number of matrices and hidden units of our network respectively. For each head of attention, we denote the learned linear maps by $\mathbf{W}_i^Q \in \mathbb{R}^{d_{model} \times d_k/h}$, $\mathbf{W}_i^K \in \mathbb{R}^{d_{model} \times d_k/h}$ and $\mathbf{W}_i^V \in \mathbb{R}^{d_{model} \times d_k/h}$, which denote to queries, keys and values respectively.

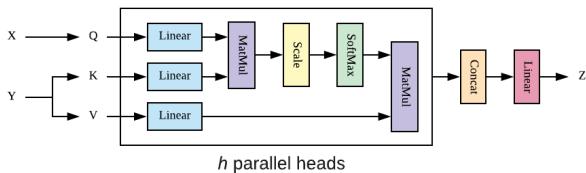


Figure 2: The graph of Multi-Head Attention computation mechanism.

Finally, all vectors produced by parallel heads are concatenated to form a single vector. Again, a linear map is used to combine different channels from different heads:

$$\mathbf{Z} = \text{Concat}(head_1, \dots, head_h)\mathbf{W}^O, \quad (2)$$

where $\mathbf{W}^O \in \mathbb{R}^{d_{model} \times d_k}$.

In this work, we employ $h = 8$ parallel attention layers, or heads. For each of these, we use $d_k = d_{model}/h = 32$. We use two separate self-attentions on $Q1$ and $Q2$, respectively, and two separate mutual-attentions between $Q1$ and $Q2$.

We attach a ResNet and a feed-forward neural network behind each multi-head attention. In the ResNet, we add the multi-head attention input and its output as an composite input to the feed-forward neural network. There are two linear variations in a full connected neural network, one of which contains a ReLU activation function. The expression is listed as follows.

$$\text{ReLU}(x) = \max(x, 0), \quad (3)$$

$$FNN(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2, \quad (4)$$

where $W_1 \in \mathbb{R}^{d_{model} \times d_k \times 2}$ and $W_2 \in \mathbb{R}^{d_{model} \times d_k}$.

C. Decoder Layer

For better results, we used GlobalMaxPooling and MLP in the decoder layer. GlobalMaxPooling can reduce the number of parameters to avoid overfitting. MLP is a forward-structured artificial neural network that maps a set of input vectors to a set of output vectors, so MLP can learn in high-dimensional data. The input layer uses ReLU layer as activation function, while the output layer uses sigmoid function. The sigmoid function is listed as follows:

$$\text{sigmod}(x) = \frac{1}{1 + e^{-x}}. \quad (5)$$

III. NEURAL NETWORKS MODEL

Because our focus is on how to integrate different neural networks rather than proposing a novel NNM, we directly use some classic models that perform well on semantic textual similarity tasks. In experiments, many different kinds of neural networks are tried. Here we only introduce a basic one as an example, that is a Siamese network[12] consists of two bidirectional long short-term memory (Bi-LSTM) networks. The structure is shown in Figure 3.

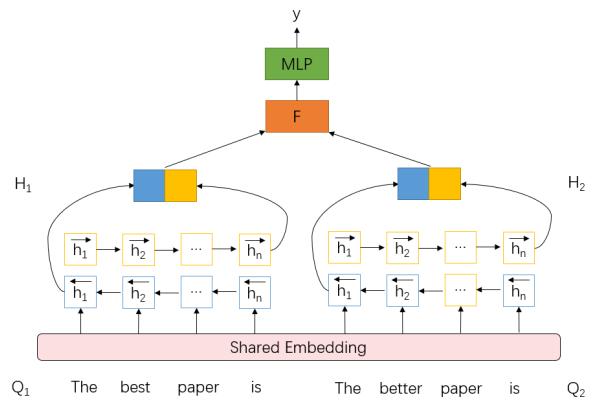


Figure 3: The structure of Bi-LSTM neural network.

At first, as a question $Q = [w_1, w_2, \dots, w_n]$ can be represented as a sequence of word embeddings which construct a matrix $E = [e_{w_1}, e_{w_2}, \dots, e_{w_n}]$, where e_{w_n}

is the word embedding of w_n . Existing works use this question matrix as the input of Bi-LSTM. For the t-th word in a question, the LSTM takes the word embedding x_t as input. The detailed computation process is listed as follows:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \quad (6)$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \quad (7)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \quad (8)$$

$$\hat{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c), \quad (9)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t, \quad (10)$$

$$h_t = o_t \odot \tanh(c_t), \quad (11)$$

where i_t, f_t, o_t, c_t are input gate, forget gate, output gate and memory cell respectively, σ denotes the sigmoid function, W_i, W_f, W_o, W_c denote weight matrices, b_i, b_f, b_o, b_c denote biases, \odot denotes the element-wise vector product, and h_{t-1} denotes the output vector at the time step $t-1$.

A Bi-LSTM consists of a forward and a backward LSTM. The hidden states of them are denoted as $(\overrightarrow{h}_1, \dots, \overrightarrow{h}_n)$, $(\overleftarrow{h}_1, \dots, \overleftarrow{h}_n)$ respectively. By connecting the last forward and backward hidden state as an output, we obtain the representation.

$$H = [\overrightarrow{h}_n; \overleftarrow{h}_1]. \quad (12)$$

For duplicate question detection we need to build a connection between the two questions, so we use the function F to reflect the connection between the two questions.

$$F = [|H_1 - H_2|; |H_1 \odot H_2|]. \quad (13)$$

Finally, we use the MLP output prediction value P . The output layer of the MLP uses the sigmoid function.

IV. ENSEMBLE APPROACH

In this section, we describe how to integrate multiple neural networks. It is well known that CNN generates separate hyperplanes by extracting adjacent words from multiple convolution kernels. RNN generates separated hyperplanes by recording the last state, and Attention is to generate separated hyperplanes by global weighted summation. Obviously, the way MHA generates the separated hyperplane is different from the above three.

Since the optimizer in the neural network mostly uses random sampling, the predicted values of the same data set on different training batches are different. That is, the separated hyperplanes used in different batches of training are fluctuated and not same. It is obviously observed that when the sample distance is very close to the separation hyperplane, the classification result obtained by the separation hyperplane is not credible, because it is likely to be classified into other categories in the next training. In order to solve this problem, this paper proposes a voting algorithm based on credibility: Credible Voting.

In this algorithm, if the predicted value of the model M is closer to 0.5, the predicted value is considered to be incredible. Based on this, we set the incredible offset $\Delta \in [0, 0.5]$ for the array of n model M predicted values

consisting of $P = [p_1, p_2, \dots, p_n]$. If the predicted value $p_i \in [0.5 - \Delta, 0.5 + \Delta]$, the result of the model M_i is considered incredible and is considered as a discard, otherwise can join voting. Finally, if the voting number of positive or negative classification is more than half, the category with the highest vote is selected, otherwise the category of all prediction values farthest from 0.5 is outputed. The description of the algorithm 1 is presented as follows.

Algorithm 1: The description of the Credible Voting algorithm

```

Input: incredible offset  $\Delta$ , model number  $n$ , model predictors set  $P = [p_1, p_2, \dots, p_n]$ 
Output: classification  $y$ 
 $zore \leftarrow 0;$ 
 $one \leftarrow 0;$ 
 $half \leftarrow \lfloor n/2 \rfloor + 1;$ 
for  $i = 1, \dots, n$  do
    if  $P[i] < 0.5 - \Delta$  then
         $zore \leftarrow zore + 1$ 
    else if  $P[i] > 0.5 + \Delta$  then
         $one \leftarrow one + 1$ 
    if  $y == 2$  then
         $closer0 \leftarrow min(P)$ 
         $closer1 \leftarrow 1 - max(P)$ 
         $y \leftarrow argmin([closer0, closer1])$ 
    return  $y$ 
```

We found that when $\Delta = 0$, there would be no discards, so CV algorithm is equivalent to absolute voting. When $\Delta = 0.5$, all predictions are considered as discards, so the algorithm selects the category farthest from 0.5. We call this with Maximum Credible Voting (MCV).

For example, assume there are five models, who present a set of predicted values of $P = [0.58, 0.1, 0.87, 0.89, 0.22]$ respectively and $\Delta = 0.2$. Then for each $p_i \in P$, if $p_i \in [0.3, 0.7]$, we deem it as a discard, so the final voting result is $negative = 2, positive = 2$, which shows that the number of votes on both sides is not more than half. Therefore, according to the algorithm, we choose the prediction value farthest from 0.5. Finally we output category negative.

CV is based on Voting to increase the discarding mechanism. When there is a rejection prediction, the sample farthest from the separation hyperplane will be selected. The MCV considers all voting as a discard on the basis of CV, and the sample farthest from the separation hyperplane will be directly selected.

V. EXPERIMENTS

A. Datasets

Quora question pairs (QQP) dataset¹ is adopted in our experiments. It consists of over 400, 000 lines of potential

¹<https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs>

question duplicate pairs. Each line contains two IDs and corresponding questions and a label. Then we shuffle the dataset randomly and split into train/dev/test set with a proportion of 8:1:1. The statistics of the question pairs are listed in Table I.

Table I: Statistics of QQP dataset

	Positive	Negative	Total
Train(80%)	119282	204150	323432
Dev(10%)	15010	25419	40429
Test(10%)	14971	25458	40429
Total	149263	255027	404290

B. Experimental Models

In this experiment, we selected some classical neural networks in text similarity tasks. These models have similar overall structures and all use QQP data sets. The main difference between these neural networks is the structure of the neurons. The neurons used in the experiment are as follows.

- **LSTM** has been specifically introduced in the "Neural Networks Model" section.
- **GRU[13]** is similar to LSTM neurons, mainly combining the forgotten gate and the input gate into an update gate. The GRU neurons are described below.

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]), \quad (14)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]), \quad (15)$$

$$\tilde{h}_t = \tanh(W_{\tilde{h}} \cdot [r_t \odot h_{t-1}, x_t]), \quad (16)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t, \quad (17)$$

where z_t, r_t are respectively represented as update gates and reset gates.

- **CNN** [14] consists of some fixed-size convolution kernels. The specific formula is described below.

$$h_t = f(W_h \odot x_{t:t+w-1} + b_h), \quad (18)$$

where f represents a nonlinear activation function, and in the experiment we use the ReLU function. w represents the convolution kernel window size and $x_{t:t+w-1}$ represents a feature matrix consisting of t to $(t + w - 1)$ input eigenvectors.

- **Attention** [15] can give weight values to the hidden state sequences (h_1, h_2, \dots, h_n) of the above three neurons. The specific expression is listed as follows:

$$e_t = \tanh(W_c \cdot h_t + b_c), \quad (19)$$

$$a_t = \frac{\exp(e_t)}{\sum_{k=1}^T \exp(e_k)}, \quad (20)$$

$$c = \sum_{t=1}^T a_t \cdot h_t, \quad (21)$$

where e_t is the hidden state of h_t after the tanh activation function, a_t is the weight value of the t time hidden state normalized by the softmax function and c is the new eigenvector obtained after weighting.

We use the Attention mechanism on Bi-LSTM, Bi-GRU and CNN to generate three new models, namely:

AT-BLSTM, AT-BGRU and AT-CNN. In the experiments, we use spaCy² as the word segmentation tool, and word embeddings³ are pre-trained using GloVe[16] with a dimension of 300. All models are trained using Adam[17] optimization method with the learning rate set to 0.001 and the batch size is set to 512.

C. Experimental Results

We set up two groups of experiments. The first group used Accuracy (Acc), Precision (P), Recall (R) and F1 Score (F1) to measure each model. The evaluation indices are defined as follows:

$$Acc = \frac{TP + TN}{TP + FP + FN + TN}, \quad (22)$$

$$P = \frac{TP}{TP + FP}, \quad (23)$$

$$R = \frac{TP}{TP + FN}, \quad (24)$$

$$F1 = \frac{2 \times P \times R}{P + R}, \quad (25)$$

where TP is short for true positive, FP represents false positive, FN stands for false negative and TN is true negative. Table II shows the results of each model under four evaluation indices.

Table II: The indicators for all models

#	Neural Network Model	Acc	P	R	F1
1	Bi-LSTM	85.69	79.87	81.18	80.52
2	AT-BLSTM	86.54	83.15	81.00	82.06
3	Bi-GRU	85.98	81.97	80.52	81.24
4	AT-BGRU	86.65	82.08	81.91	82.00
5	CNN	85.21	78.23	81.14	79.66
6	AT-CNN	85.18	81.84	78.92	80.35
7	Multi-Head Attention	86.83	84.07	81.06	82.54

Through Table II, we found that all accuracies of these seven models are around 86%. At the same time, we found that Multi-Head Attention performed best on Acc, P, and R. Additionally, we found that AT-BGRU performs best on R.

The second group of experiments combines Mixed Multi-Head Attention with six neural network models, and compares the difference in accuracy among the three methods of voting, Maximum Credible Voting (MCV) and Credible Voting (CV). In the case of the same number of votes in each voting method, this experiments use negative as default setting. Table III shows the accuracy between the different combinations under the three integration strategies, where each of combinations has Multi-Head Attention.

In the first six rows of Table III, the MCV and CV have higher increasing rate. Mainly because the number of neural networks in the first six rows is even, the same number of votes will occur. In this case, the experiments use negative as default setting.

²<https://spacy.io/>

³<http://nlp.stanford.edu/data/glove.840B.300d.zip>

Table III: Results of the ensemble method. The second column is the accuracy of Voting. The third and fourth columns are the accuracy of our proposed ensemble methods, i.e., Maximum Credible Voting (MCV) and Credible Voting (CV). The improvements are listed in the parentheses.

#	Combined models	Voting	MCV	CV
1	Bi-LSTM	86.6	87.6(+1.0)	87.6(+1.0)
2	AT-BLSTM	87.1	88.2(+1.1)	88.2(+1.1)
3	Bi-GRU	86.8	88.1(+1.3)	88.1(+1.3)
4	AT-BGRU	87.0	88.2(+1.2)	88.2(+1.2)
5	CNN	86.1	87.3(+1.2)	87.3(+1.2)
6	AT-CNN	86.6	87.9(+1.3)	87.9(+1.3)
7	AT-BLSTM&Bi-LSTM	88.4	88.4(+0.0)	88.6(+0.2)
8	AT-BGRU&Bi-GRU	88.3	88.4(+0.1)	88.6(+0.3)
9	AT-CNN&CNN	88.1	87.6(-0.4)	88.2(+0.2)
10	Bi-LSTM&CNN	88.3	88.0(-0.3)	88.5(+0.2)
11	Bi-LSTM&Bi-GRU	88.1	87.8(-0.3)	88.2(+0.1)
12	Bi-GRU&CNN	88.2	87.6(-0.6)	88.4(+0.1)
13	AT-BLSTM&AT-BGRU	88.5	88.8(+0.3)	88.9(+0.5)
14	AT-BLSTM&AT-CNN	88.2	88.4(+0.2)	88.6(+0.4)
15	AT-BGRU&AT-CNN	88.3	88.4(+0.2)	88.6(+0.3)
16	All models except: AT-CNN&CNN	88.9	88.7(-0.2)	89.0(+0.1)
17	All models except: AT-BGRU&Bi-GRU	88.7	88.5(-0.2)	89.0(+0.2)
18	All models except: AT-BLSTM&Bi-LSTM	88.8	88.4(-0.4)	89.0(+0.2)
19	All models except:MHA	88.7	88.6(-0.1)	89.1(+0.4)
20	All models	89.1	88.7(-0.4)	89.3(+0.2)

As can be seen from the table III, after the 6 lines, the MCV has a lower accuracy than the voting. The reason for this is: As the number of models increases, the greater the probability of false predictions, the lower the accuracy than Voting. At the same time, the accuracy of CV is always higher than voting, indicating that the CV algorithm is effective.

It can be seen from the results of the last two lines that after adding the MHA model, the three voting algorithms have improved to different degrees, indicating that the MHA model can improve the accuracy of the voting method.

As can be seen from Table III, the accuracy of CV is higher than Voting and MCV. The main reason is that when deep learning models cannot resolve the text, the classification result will be very close to the separation hyperplane.

D. Further Analysis of the Ensemble Approach

In this subsection, we provide some further analysis of the ensemble approach, mainly about the detailed statistics on the 19th line of the Table III.

Figure 4 shows the effect of different Δ on accuracy, where the step size of Δ is 0.05. We found that when $\Delta = 0$, its result is the same as Voting, when $\Delta = 0.5$, the result is the same as MCV and peaks at $\Delta = 0.2$. We perform a detailed analysis of $\Delta = 0$ and $\Delta = 0.2$, and count how many neural networks can correctly predict each problem.

From (a) of Figure 5 we can see that the proportion of four and above models successfully predicted is 89.1%, which is the same as the Voting result. In (b) of Figure 5, we found that due to the discarded votes, the number that

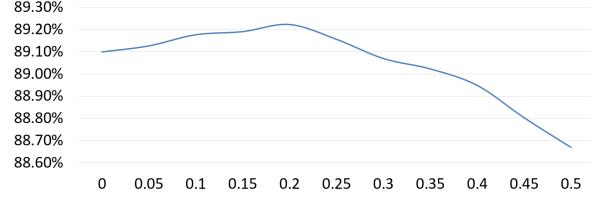


Figure 4: The influence of Δ in accuracy

can be successfully predicted by seven neural networks has experienced a large decline(from 63.51% to 58.48%), and the rest also occurs to varying degrees. According to the description of the algorithm 1, we find that our improvement in accuracy mainly comes from the part of the model whose number is less than or equal to three. From (a) we can see that the proportion of this part is small (only 10.75%), so the overall rate of increase is also relatively small (only 0.2%).

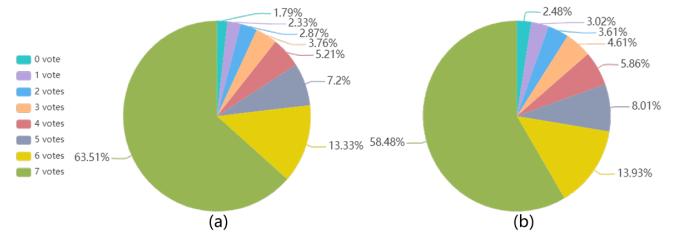


Figure 5: (a) is the number of correct prediction of Voting and (b) is the number of correct prediction of CV

VI. CONCLUSION

Duplicate question detection is a classic and important binary classification problem. In this paper, we use multi-head attention to process duplicate question detection task and propose the Credible Voting algorithm. The algorithm eliminates the separation hyperplane which is too close to the sample, obtains stable classification results in several rounds of training, and improves the accuracy. Finally, the algorithm can achieve an accuracy of 89.3% on this task.

REFERENCES

- [1] K. Zhang, W. Wu, H. Wu, Z. Li, and M. Zhou, “Question retrieval with high quality answers in community question answering,” in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 2014, pp. 371–380.
- [2] S. Filice, G. Da San Martino, and A. Moschitti, “Kelp at semeval-2017 task 3: Learning pairwise patterns in community question answering,” in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 2017, pp. 326–333.
- [3] M. Salvador, S. Kar, T. Solorio, and P. Rosso, “Combining lexical and semantic-based features for community question answering,” *Proceedings of SemEval*, pp. 814–821, 2016.
- [4] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *nature*, vol. 323, no. 6088, p. 533, 1986.

- [5] P. Nakov, L. Márquez, and F. Guzmán, “It takes three to tango: Triangulation approach to answer ranking in community question answering,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 1586–1597.
- [6] P. Wang, L. Ji, J. Yan, D. Dou, N. D. Silva, Y. Zhang, and L. Jin, “Concept and attention-based cnn for question retrieval in multi-view learning,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 9, no. 4, p. 41, 2018.
- [7] S. Romeo, G. Da San Martino, A. Barrón-Cedeno, A. Moschitti, Y. Belinkov, W.-N. Hsu, Y. Zhang, M. Mohtarami, and J. Glass, “Neural attention for learning to rank questions in community question answering,” in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 1734–1745.
- [8] M. Zhang and Y. Wu, “An unsupervised model with attention autoencoders for question retrieval,” *arXiv preprint arXiv:1803.03476*, 2018.
- [9] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [10] M. Lin, Q. Chen, and S. Yan, “Network in network,” *arXiv preprint arXiv:1312.4400*, 2013.
- [11] C. M. Bishop *et al.*, *Neural networks for pattern recognition*. Oxford university press, 1995.
- [12] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, “Signature verification using a” siamese” time delay neural network,” in *Advances in neural information processing systems*, 1994, pp. 737–744.
- [13] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.
- [14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [15] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *Computer Science*, 2014.
- [16] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.
- [17] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

Acoustic Scene Classification Using Deep Convolutional Neural Network via Transfer Learning

Min Ye^{1,2}, Hong Zhong¹, Xiao Song², Shilei Huang³, Gang Cheng^{2,*}

¹School of Computer Science and Technology, Anhui University, Hefei, China

²PKU Shenzhen Institute, Shenzhen, China

³Shenzhen Raisound Technologies, Co., Ltd, Shenzhen, China

{min.ye,xiao.song}@imsl.org.cn; zhongh@ahu.edu.cn; {shilei.huang,gang.cheng}@raisound.com

We use deep convolutional neural network via transfer learning for Acoustic Scene Classification (ASC). For this purpose, a powerful and popular deep learning architecture — Residual Neural Network (Resnet) is adopted. Transfer learning is used to fine-tune the pre-trained Resnet model on the TUT Urban Acoustic Scenes 2018 dataset. Furthermore, the focal loss is used to improve overall performance. In order to reduce the chance of overfitting, data augmentation technique is applied based on mixup. Our best system has achieved an improvement of more than 10% in terms of class-wise accuracy with respect to the Detection and classification of acoustic scenes and events (DCASE) 2018 baseline system on the TUT Urban Acoustic Scenes 2018 dataset.

Keywords—transfer learning; Acoustic Scene Classification; focal loss; mixup

I. INTRODUCTION

With the increase of the amount of data, the neural network with many layers shows superior performance. But if the network is deeper, the gradient exploding of the back propagation becomes a problem. ResNet [1] solved this problem and achieved state-of-the-art performance in the fields of image classification, object detection, instance segmentation, semantic segmentation. However, training Resnet from scratch requires large amounts of data and high computational resources. Transfer learning [2] save computational resources by applying knowledge learned from a problem to another different but related problem.

Acoustic Scene Classification (ASC) is the task to identify audio recordings that recorded in a public area into one of several predefined acoustic scene classes, such as “park”, “pedestrian street” and “metro station”. There have been several attempts to use transfer learning for ASC. For example, deep neural network based learning and transferring mid-level audio features for ASC [3], a study on transfer learning for Acoustic Event Detection (AED) in a real life scenario [4] and an investigation of transfer learning mechanism for ASC [5].

In this paper, we describe ASC system based on transfer learning for identifying audio that recorded in a public area into one of several predefined acoustic scene classes. For this purpose, we fine-tune pre-trained deep convolutional neural network of Resnet on the TUT Urban Acoustic Scenes 2018 dataset. We also adopt the focal loss [6] to further improve

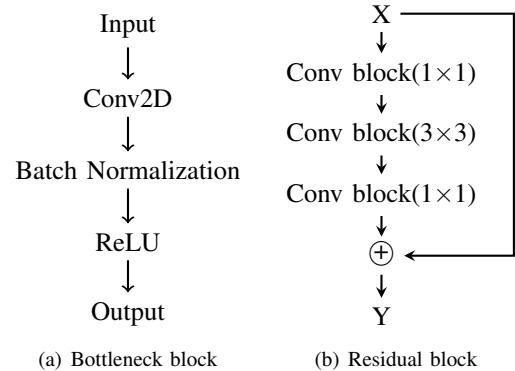


Fig. 1. Detail of blocks.

classification performance, data augmentation using the mixup [7] is introduced to diversify the training data.

The remaining content of this paper is organized as follows: Section II describes the transfer learning ASC system. Our experiments are described in Section III. Section IV is the conclusion.

II. TRANSFER LEARNING ASC SYSTEM

This section describes the transfer learning ASC system. The following subsections give more description on transfer learning and model fine-tuning, Resnet, focal loss and data augmentation using mixup.

A. Transfer learning and model fine-tuning

Transfer learning is a machine learning technique in which a model trained on one task is reused for another related task. If there is not enough data to train large networks from scratch, transfer learning can be used to avoid overfitting [8]. Transfer learning also saves computational resources.

In this paper, we train the pre-trained Resnet model and fine-tune the pre-training Resnet model weight [9] on the TUT Urban Acoustic Scenes 2018 dataset. In this case, the number of nodes in the output layer must be modified to match the number of classes for ASC. In addition, the data must match the input size of the pre-training Resnet model.

B. Resnet

As we all know, network depth is the factor that determines network performance. Deeper and deeper networks are used in

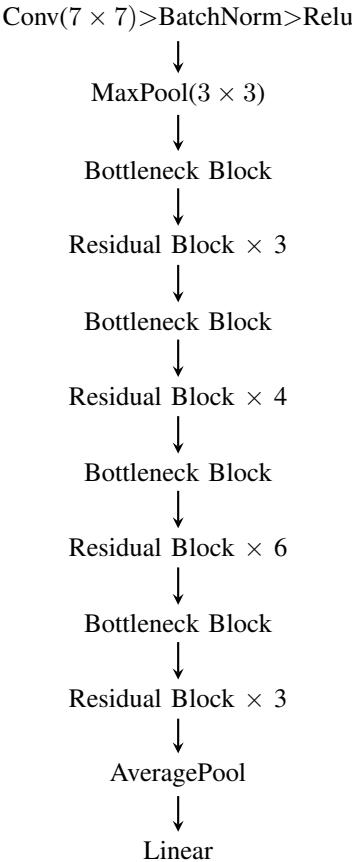


Fig. 2. Resnet architecture.

computer vision area. However, due to the gradient vanishing problem, deep network training is not easy. ResNet solved this problem and provide a training framework to ease the training of networks that are substantially deeper than those used previously.

The detail of blocks in this paper are shown in Fig. 1. Fig. 1 (a) shows the bottleneck block and Fig. 1 (b) shows the residual block. For bottleneck block, Batch Normalization (BN) [10] is added after convolution. The ReLU function is used as a non-linearity of Resnet after BN. Residual block have three bottleneck blocks. The convolution kernels of the three bottleneck blocks are 1×1 , 3×3 , 1×1 , respectively. Residual block first reduces the input channel count with 1×1 kernels, applies 3×3 convolutions in that reduced channel size and then restores to the input channel size back with 1×1 convolutional layer.

The Resnet architecture in this paper are shown in Fig. 2. As show in Fig. 2, the layers of Residual block is 3, 4, 6, 3, respectively. Resnet is based on 50 convolutional layers in this paper.

C. Focal loss

We train the network with the extension of the focal loss [6], which was previously used for object detection in images of natural scenes using the stochastic gradient descent

optimizer. Focal loss is an extension of Cross Entropy (CE) loss, which solves very large class imbalance problems and performs implicit negative mining by imposing higher losses on uncertain prediction. There may have the sample imbalance problem in ASC.

The normal CE loss for object detection is showed below:

$$CE(p, y) = -\sum_{j=0}^c y_j \log(p_j) \quad (1)$$

where y specifies the ground-truth class and $p \in \{0, 1\}$ is the models estimated probability, j represents the j -th class. The CE can solve the problem that the weight update too slow. When the error is large, the weight update fast, so CE is widely used. The focal loss for object detection is showed below:

$$FL(p, y) = -\alpha_t (1 - p_t)^\gamma \log(p_t) \quad (2)$$

In this paper, ASC is multiple classification but object detection is binary classification, therefore we modify the focal loss as follows:

$$Label(p, y) = -\frac{1}{n} \sum_{i=0}^n \alpha_t (1 - p_t^i)^\gamma \log(p_t^i) \quad (3)$$

For focal loss, we only need to select the appropriate hyper-parameters α , γ .

D. Data Augmentation using Mixup

We uses mixup as a method of data augmentation. This method can improve the generalization ability of the model and construct a virtual training sample. The mathematical expression for the mixup is as follows:

$$\hat{x} = \lambda x_i + (1 - \lambda)x_j \quad (4)$$

$$\hat{y} = \lambda y_i + (1 - \lambda)y_j \quad (5)$$

Where (x_i, y_i) and (x_j, y_j) are two samples randomly extracted from the training data, and $\lambda \in [0, 1]$, $\lambda \sim Beta(\alpha, \alpha)$, $\alpha \in (0, \infty)$. This mixture extends the features of training set and label distribution through linear interpolation of eigenvectors and linear interpolation of corresponding labels. The super-parameter α of the mixup controls the interpolation strength of the features and labels.

III. EXPERIMENTAL

A. Datasets

TUT Urban Acoustic Scenes 2018 dataset [11] includes 10 categories of acoustic scenes such as “bus”, “airport”, “park” and “metro”. The dataset was recorded in six large European cities, and in different locations for each scene class. For each location there is one 5-6 minute audio recordings. The original recordings were split into segments with a length of 10 seconds that are provided in individual files. The dataset consists of 10-second audio segments from 10 acoustic scenes. TUT Urban Acoustic Scenes 2018 development dataset contains in total 24 hours of audio and each acoustic scene has 864 segments (144 minutes of audio). TUT Urban Acoustic Scenes 2018

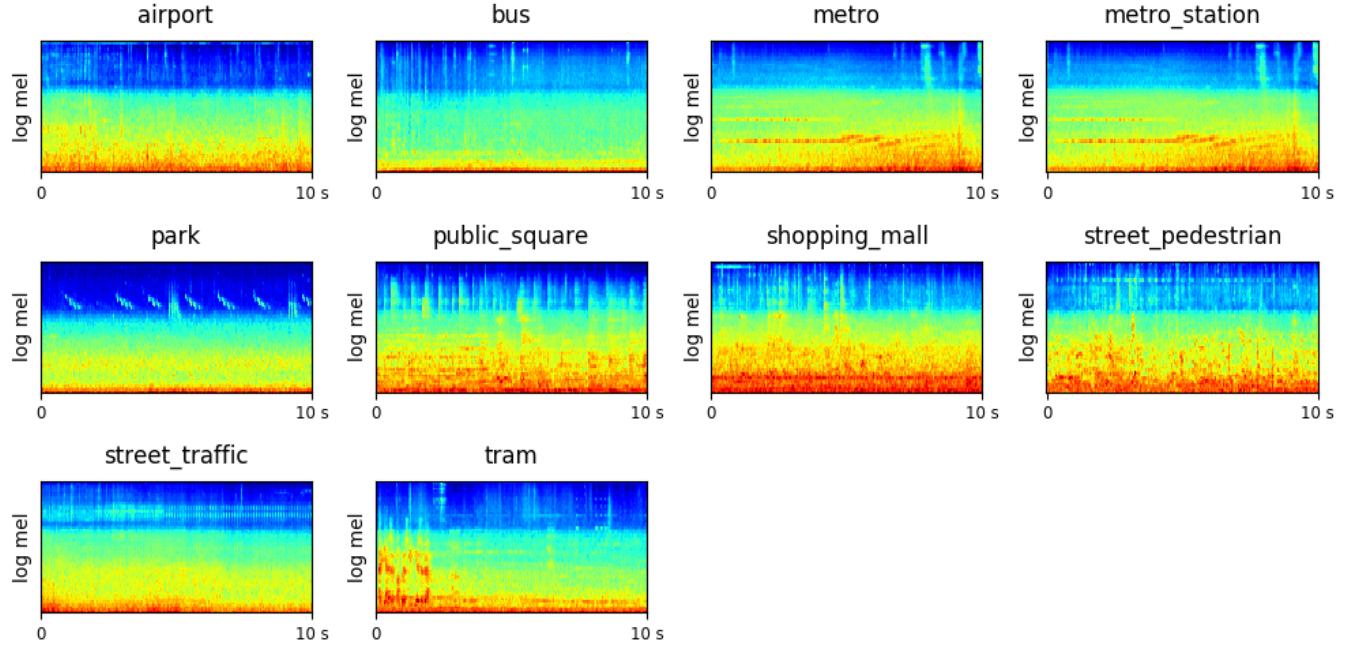


Fig. 3. The log-mel spectrograms of the scenes.

leaderboard dataset contains in total about 3.3 hours of audio and each acoustic scene has 120 segments (20 minutes of audio).

In this paper, we choose TUT Urban Acoustic Scenes 2018 development dataset as the training subset, which all contains 8,640 segments. Also, we choose TUT Urban Acoustic Scenes 2018 leaderboard dataset as the test subset, which all contains 1,200 segments.

B. Evaluation

The scoring of ASC will be based on classification accuracy: the number of correctly classified segments among the total number of segments. Accuracy will be calculated as average of the class-wise accuracy.

C. Feature extraction

The log-mel spectrograms of the scenes are shown in Fig. 3. We extract the spectrograms and apply log-mel filter banks on the spectrograms followed by logarithm operation where the sampling rate is set to be 48 kHz. A short-time Fourier transform (STFT) [12] with a Hanning window size of 2048 samples and a hop size of 1500 samples is used to extract the spectrogram, so that there are 320 frames in an audio clip. The mel filter bank has a cut-off frequency of 50 Hz. Mel filter banks with 64 bins and cut-off frequencies of 50 Hz to 24 kHz are applied on the spectrogram.

D. Setup

We use AdaBound [13] optimizer for gradient-based optimization [14] with learning rate of 0.001. The learning rate

is reduced by multiplying 0.9 after every 200 iterations of training. As described in reference [13], AdaBound is an optimizer that behaves like Adam at the beginning of training, and gradually transforms to Stochastic Gradient Descent (SGD) at the end. The final learning rate parameter indicates AdaBound would transform to SGD with this learning rate. In common cases, a default final learning rate of 0.1 can achieve relatively good and stable results on unseen data. It is not very sensitive to its hyper-parameters. The transfer learning ASC system are optimized during 5000 maximum iteration steps, which are empirically set. The transfer learning ASC system are implemented with Python and Pytorch. The hyper-parameters α of mixup is 0.3.

TABLE I
CLASS-WISE ACCURACY OF THE ASC SYSTEM.

Model	Accuracy
Baseline [11]	0.625
SE-Resnet [15]	0.725
NLL loss	0.725
focal loss	0.733
focal loss + mixup	0.747

E. Results and discussion

TABLE I shows the class-wise accuracy of our systems on the TUT Urban Acoustic Scenes 2018 dataset. To confirm the performance of our ASC systems, we compared it with two conventional methods, The baseline [11] of the DCASE 2018 Challenge and SE-Resnet [15].

a) *Compare our systems and baseline:* The baseline of the DCASE 2018 Challenge is based on two convolutional layers. Compared with the 62.5% class-wise accuracy of the baseline system, the best class-wise accuracy achieved by our approach was 74.7% and improves class-wise accuracy by almost 10%.

b) *Compare our systems and SE-Resnet:* SE-Resnet is based on 152 convolutional layers of squeeze-and-excitation [16] Resnet. Our system is based on 50 convolutional layers of pre-trained Resnet. Compared with the 72.5% class-wise accuracy of the SE-Resnet, our best method achieves a relative improvement of 2.2% on the TUT Urban Acoustic Scenes 2018 dataset.

c) *On the effect of focal loss:* Our experiment analyze the effect of focal loss. NLL loss is CE loss. The class-wise accuracy of our system with NLL loss was 72.5% and the class-wise accuracy of our system with focal loss was 73.3%, respectively. The focal loss could solve the problem that some samples are difficult to recognize, the system with focal loss achieves 0.8% relative improvement for ASC.

d) *On the effect of mixup:* Our experiment analyze the effect of mixup. We use focal loss. The class-wise accuracy of our system without mixup was 73.3% and the class-wise accuracy of our system with mixup was 74.7%, respectively. The system with mixup achieves 1.4% relative improvement compared with the method without mixup.

IV. CONCLUSION

We applied transfer learning to identify acoustic scenes using deep convolutional neural network. The utilized network is based on pre-trained model of Resnet. We evaluated the network on the TUT Urban Acoustic Scenes 2018 dataset by fine-tuning the pre-trained model of Resnet. We use focal loss solve very large class imbalance problems and mixup as data augmentation. Our best model obtained more than 10% improvement over the baseline system of DCASE 2018 challenge on the TUT Urban Acoustic Scenes 2018 dataset.

ACKNOWLEDGMENT

This work was supported by Shenzhen Basic Research Program (JCYJ20170817155939233) and Shenzhen Technology Project (JSGG20170822105644555).

REFERENCES

- [1] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [2] K. Weiss, T. M. Khoshgoftaar, and D. Wang, “A survey of transfer learning,” *Journal of Big data*, vol. 3, no. 1, p. 9, 2016.
- [3] S. Mun, S. Shon, W. Kim, D. K. Han, and H. Ko, “Deep neural network based learning and transferring mid-level audio features for acoustic scene classification,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 796–800.
- [4] P. Arora and R. Haeb-Umbach, “A study on transfer learning for acoustic event detection in a real life scenario,” in *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2017, pp. 1–6.
- [5] H. Zhou, X. Bai, and J. Du, “An investigation of transfer learning mechanism for acoustic scene classification,” in *2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, 2018, pp. 404–408.
- [6] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.
- [7] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” *arXiv preprint arXiv:1710.09412*, 2017.
- [8] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *Advances in neural information processing systems*, 2014, pp. 3320–3328.
- [9] O. Lucena, A. Junior, V. Moia, R. Souza, E. Valle, and R. Lotufo, “Transfer learning using convolutional neural networks for face anti-spoofing,” in *International Conference Image Analysis and Recognition*. Springer, 2017, pp. 27–34.
- [10] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.
- [11] A. Mesaros, T. Heittola, and T. Virtanen, “A multi-device dataset for urban acoustic scene classification,” *arXiv preprint arXiv:1807.09840*, 2018.
- [12] A. Nuruzzaman, O. Boyraz, and B. Jalali, “Time-stretched short-time fourier transform,” *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 2, pp. 598–602, 2006.
- [13] L. Luo, Y. Xiong, Y. Liu, and X. Sun, “Adaptive gradient methods with dynamic bound of learning rate,” in *Proceedings of the 7th International Conference on Learning Representations*, May 2019.
- [14] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [15] J. H. Yang, N. K. Kim, and H. K. Kim, “Se-resnet with gan-based data augmentation applied to acoustic scene classification,” DCASE2018 Challenge, Tech. Rep., September 2018.
- [16] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

Japanese Particle Error Correction employing Classification Model

Youichiro Ogawa and Kazuhide Yamamoto

Nagaoka University of Technology

Nagaoka, Niigata, Japan

{ogawa, yamamoto}@jnlp.org

Abstract—We present a grammatical error correction system for Japanese particles based on the classification method. We define a confusion set of the particles for detection of particle errors and prediction of the correct word. Our method can handle not only substitutions but also insertions and deletions. For building the training data, we used two datasets: a large amount of native language data and corrected learners' sentences. That is, we did not require a parallel corpus of learners. We show the results for Japanese particle error correction on the NAIST Goyo corpus, evaluated by the MaxMatch (M^2) score. In addition, we analyze the effect of percentage changes in deletion labels while building the training data and analyze the prediction probability threshold at correction. Our best model achieved 46.4 $F_{0.5}$.

Keywords-grammatical error correction; Japanese particle errors; classification model;

I. INTRODUCTION

With the increasing number of non-native Japanese learners, the demand for the development of writing assistance tools and error correction systems is growing. Grammatical error correction (GEC) system automatically corrects various types of errors in text written by non-native learners. The GEC system can be used to support the language teachers' evaluation of the learners' texts and support the learners' language acquisition from e-learning resources.

The Japanese particle is the largest obstacle for the Japanese language learners. Japanese particles are suffixes or short words in Japanese grammar that immediately follow the modified noun, verb, adjective, or sentence. They indicate various meanings and functions. Among the many types of errors committed by the learners, the Japanese particle error is the most frequent type (23% of all error types), as shown by the analysis of the NAIST Goyo Corpus that consists of Japanese learners' texts and their annotations with the error types.

For example, a Japanese learner may write:

私は車 ϕ 買った。
(I bought a car.)

This sentence has the grammatical error (ϕ) of a missing “を,” which determines the relationship between the verb and the object. This error corresponds to a preposition/article error in English.

We present a method to correct Japanese particle errors by using a classification model. For this task, we define a confusion set, which is the Japanese particle word list used for detection of particle errors or selection of the

correct particle word. For handling the insert operation, we also define a complemental condition to judge whether to insert the word. The training data are built from “correct” sentences, such as from the native language data or the corrected sentences of the learner. Therefore, we do not require a parallel corpus of learners and we can use a large amount of data. Our correction method can handle not only word substitutions but also insertions and deletions. Our best model achieved 46.4 $F_{0.5}$ on the NAIST Goyo Corpus.

II. RELATED WORKS

Many promising approaches have been proposed for grammatical error correction. They can be categorized into two types: machine translation (MT) and classification. MT-based methods aim to directly translate an incorrect sentence to a correct one. MT-based methods have been actively studied [2] [3] [4]. Junczys-Dowmunt and Grundkiewicz [5] used a statistical machine translation (SMT) trained with parallel corpus containing learner sentences and the corresponding correct sentences. Recent developments in neural machine translation (NMT) have also been successful. Yuan and Briscoe [6] are the first to apply NMT for GEC. Chollampatt and Ng [7] proposed a multilayer convolutional encoder-decoder neural network. Grundkiewicz and Junczys-Dowmunt [8] developed a hybrid system by combining SMT and NMT, resulting in an exceptionally good performance. Zhao et al. [9] used the copy mechanism for GEC. Stahlberg et al. [10] adapted the finite state transducer for GEC.

Classification-based methods treat GEC as a classification task and predict the correct word from a confusion set which includes lists of confusable words for each error types, for example, {a, an, the} for articles [11] [12] [13]. Bryant and Briscoe [14] used an N-gram language model trained with a large monolingual corpus for classification. Rozovskaya and Roth [15] constructed Naive Bayes classifiers from native data and injected learner error patterns into them. Wang et al. [16] proposed a deep-context model that used a recurrent neural network (RNN) to extract contextual information from an input sentence. By adding an attention mechanism, Kaili et al. [17] outperformed other classification approaches and showed comparable performance to highly effective MT methods. In contrast to the MT methods, classification methods do not always require a learner-annotated or parallel corpus. Classifiers are trained individually for specific error types. Because the error type needs to be defined, only well-defined

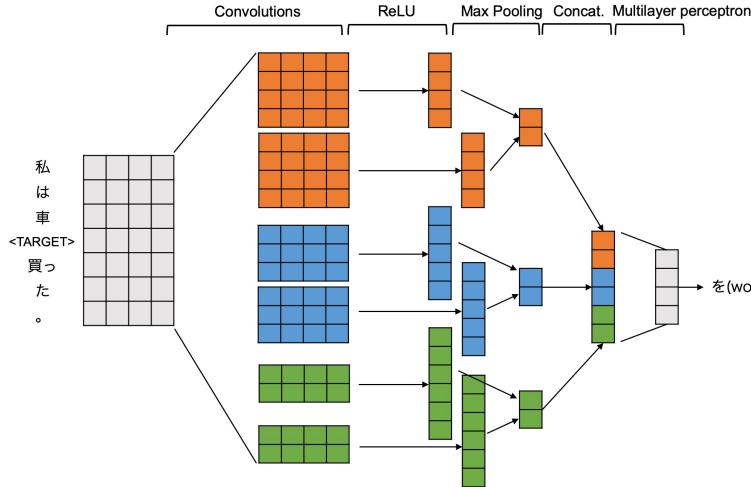


Figure 1. Shallow-and-wide convolutional networks [1]: Three convolutional layers with respective kernel window sizes 3,4,5 are used. A global max-pooling is then applied to the whole sequence on each filter. Finally, the outputs of each kernel are concatenated to a unique vector and provided to a fully connected layer. An example of the input and output is shown.

mistakes can be typically addressed in a straightforward way.

As for the Japanese language, Mizumoto et al. [18] proposed a GEC model for Japanese learners’ text using SMT. This approach can handle all error types but is challenging because there is a limit to the correction patterns that can be trained. Imamura et al. [19] studied Japanese particle error correction using conditional random fields that allow insertion and deletion. However, their system uses features surrounding the target words; therefore, it could not consider far away words, which is important for correction.

In the MT approach, the output vocabulary space is huge because the model predicts the next word from the whole vocabulary (usually limited to 40,000 to 50,000) in order. This approach is powerful and used in many studies because it can handle various error types such as phrase-level errors and word order errors. In this study, however, the output candidates are very limited since we focus on only particle errors. Also, non-particle words should not be changed.

Therefore, we propose a correction method specialized on Japanese particle errors, which predicts the correct particle from limited candidates for the detected positions, while preserving non-particle words.

III. JAPANESE PARTICLE ERROR CORRECTION

A. Learner corpus analysis

The NAIST Goyo corpus was created by [20] in which learner sentences are annotated with the error type by native speakers. We focus on particles because these errors are the most frequent in the corpus, accounting for 23% of all error tags (2,739/14,207). The 2,739 sentences include particle error tags and 373 patterns of particle errors.

Table I
DISTRIBUTION OF PARTICLE ERRORS

Incorrect particle	Correct particle	Total	%
ϕ	no	231	6.85
ga	wa	187	5.54
ϕ	wa	154	4.57
wa	ga	150	4.45
ϕ	ni	135	4.00
ϕ	wo	114	3.38
no	ϕ	106	3.14
ga	wo	95	2.85
wo	ga	95	2.82
ni	de	87	2.58

Frequencies for the top 10 Japanese particle errors are shown in Table I.

We decided on the following confusion set of Japanese particles: { ga no wo ni he to yori kara de ya wa niwa karawa towa deva hewa madewa yoriwa made }. There are 3,207 error tags for the target particles, accounting for 95.08% of all particle error tags, giving this confusion set enough coverage. Note that missing errors account for 28.50%, and unnecessary errors account for 10.76% of the target particle errors.

B. Classification model

Figure 1 illustrates the classification model. We employed the shallow-and-wide convolutional neural network (CNN) model [1], which is an ensemble of convolutional kernels trained in a single layer. Consequently, the network is wide and only has a small number of hidden layers. Let $\mathbf{x}_i \in R^d$ be an input token. The input h -gram $\mathbf{x}_{i:i+h-1}$ is transformed through a convolution filter $\mathbf{w}_c \in R^{hd}$:

$$c_i = f(\mathbf{w}_c \cdot \mathbf{x}_{i:i+h-1} + b_c) \quad (1)$$

Here, $b_c \in R$ is a bias term and f the non-linear ReLU function [21]. This produces a feature map $\mathbf{c} \in R^{n-h+1}$,

where n is the number of tokens in the sentence. Then, we apply a global max-over-time pooling over the feature map:

$$\hat{c} = \max(\mathbf{c}) \in R \quad (2)$$

This process is repeated for one feature to obtain m filters with different window sizes h . The resulting filters are concatenated to form a shallow-and-wide network:

$$\mathbf{g} = [\hat{c}_1, \hat{c}_2, \dots, \hat{c}_m] \quad (3)$$

Finally, a fully connected layer is applied:

$$\hat{y} = f(\mathbf{w}_y \cdot \mathbf{g} + b_y) \quad (4)$$

For particle error correction, we input a sentence with a *TARGET* label to the model and it predicts a particle word included in the confusion set or a *DEL* label. *TARGET* represents the position of the word to be predicted by the model. In other words, this is a MASK prediction task. The label *DEL* indicates that the predicted result is deletion. Our model not only replaces the word but also inserts and deletes it.

C. Correction method

We employed a classification model that predicts the correct word from a given confusion set. Details of the classifier are described in section III-B. Our system corrects the particle errors in an input sentence as follows.

- 1) Words and the parts-of-speech (POS) they belong to are obtained from the input sentence by the morphological analyzer. We use MeCab¹ with UniDic² for the Japanese morphological analyzer.
- 2) Positions of the target particles (target positions) are detected. It is repeatedly determined whether the POS is a particle and the word is included in the confusion set.
- 3) The positions that satisfy the *complemental conditions* (complemental positions) are detected. After analyzing the appearance pattern of particles from a native corpus, we set the complementary conditions such that the POS of the previous word is either noun, pronoun, auxiliary verb, particle, noun-like suffix, or verb, and the POS of the next word is not particle or auxiliary verb.
- 4) Our classifier predicts from the beginning repeatedly for the positions detected in 2 and 3. If this position is in the target positions, its word is replaced with *TARGET* label and the classifier predicts a substitution or deletion. If this position is in the complementary positions, the *TARGET* label is inserted into this position and the classifier predicts an insertion or nothing.
- 5) Steps 1 to 4 are repeated to the end of the sentence.

Table II
TRAINING AND EVALUATION CORPUS

	Corpus	Sentences
Training	BCCWJ	5,384,131
	Lang-8	1,574,343
Evaluation	NAIST Goyo Corpus	2,076

D. Building training data

For training the classification model, we built the training data from “correct” sentences such as from native speakers’ data. The classification model predicts a word or deletes it according to the position of *TARGET* label in the input. We prepared *TARGET*-labeled sentences intentionally from correct sentences. Note that there is only one label per sentence.

There are two ways of preparing labels. One is to replace one of the target particle words with the label. It makes the model learn word substitution and insertion. The other is to insert the label at one of the complementary positions. It makes the model learn word deletion. Comparing the number of detections, we can see more complementary positions than target positions because the complementary condition is lenient. Therefore, if we choose randomly from among the target and complementary positions, deletion will be learned well, but the learning is not balanced because of fewer missing errors (described in section III-A). Therefore, we introduced a deletion rate D to control the number of insertions of the *TARGET* label.

When a sentence is given for training, it is labeled as follows.

- 1) By performing steps 1, 2, and 3 described in section III-C, the target and complementary positions are obtained.
- 2) Whether one of the target particle words is replaced with *TARGET* label or *TARGET* is inserted into one of the complementary positions is chosen with the deletion rate D .
- 3) If it is a replaced *TARGET* label, the original word is marked as the answer label. The word to be replaced is randomly chosen.
- 4) If it is an inserted *TARGET* label, the *DEL* label is marked as the answer label. The position to be inserted is randomly chosen.

IV. EXPERIMENTS

A. Datasets

For training, we used the Balanced Corpus of Contemporary Written Japanese (BCCWJ)³. BCCWJ is a corpus created for comprehending the breadth of contemporary written Japanese. The data comprise 104.3 million words and cover multiple genres and fields including general books, magazines, newspapers, business reports, blogs, internet forums, textbooks, and legal documents. In addition, the corrected sentences of Lang-8 Learner Corpora⁴ were combined to BCCWJ because the domain knowledge of

¹<http://taku910.github.io/mecab/>

²<https://unidic.ninjal.ac.jp/>

³http://pj.ninjal.ac.jp/corpus_center/bccwj/en/

⁴<http://lang-8.com>

Table III
COMPARISON OF GEC SYSTEMS ON THE NAIST GOYO CORPUS

Model	Dataset	Precision(%)	Recall(%)	$F_{0.5}$
5-gram LM	BCCWJ+Lang8	13.3	29.5	14.9
CNN	BCCWJ	28.9	52.4	31.7
CNN	BCCWJ+Lang-8	30.9	52.5	33.7
+kana	BCCWJ+Lang-8	30.8	53.2	33.6
+emb	BCCWJ+Lang-8	33.0	54.9	35.8
+kana+emb	BCCWJ+Lang-8	33.3	55.3	36.1

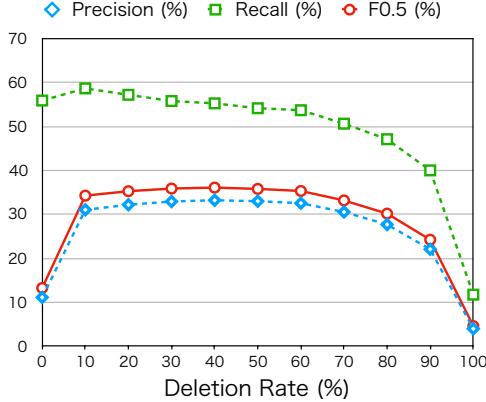


Figure 2. The effect of changing the deletion rate D on M^2 score. The model is CNN+kana+emb trained with BCCWJ+Lang8.

learner writing is useful in correction methods. We only used sentences composed of Japanese characters and symbols, and 10,000 sentences were used for the development set. Deletion rate D was set to 0.4 for building training data (discussed in section VI-A).

For evaluation, we used the NAIST Goyo corpus as test data. We extracted sentences that contained particle errors and corrected the non-particle errors in advance. Table II shows the training/evaluation corpus size.

Japanese learners use Hiragana more frequently than Kanji⁵ because there are several types of Kanji, and many learners are unfamiliar with them. Many Hiragana words are not included in the native data, and the model treats them as unknown words. To avoid this problem, we converted all the words to Hiragana as pre-processing.

B. Model settings

For our models, we used a token hidden size of dimension 500. The CNN had one layer. We set the dropout [22] to 0.1. The vocabulary consisted of the 40,000 most common words, and out-of-vocabulary words were replaced with a *UNK* token. Word embeddings were initialized by NWJC2Vec [23], which was pre-trained using word2vec [24] with the NINJAL Web Japanese Corpus. The models were optimized with Adam.

C. Baseline

We built a 5-gram language model (LM) using the KenLM toolkit [25]. The TARGET label in the input was

⁵Japanese language has two kinds of characters: Hiragana and Kanji. Hiragana is a phonogram similar to the English alphabet and the phonemes alone do not express meaning. Kanji is an ideogram where each character has a specific meaning.

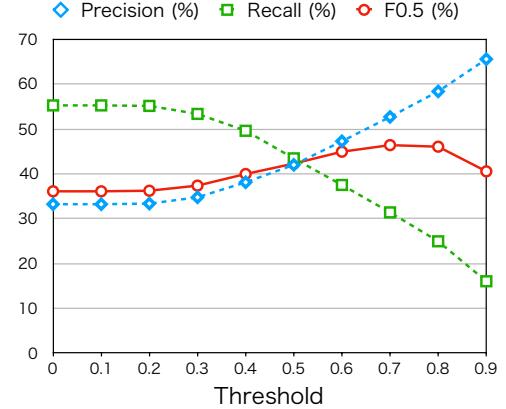


Figure 3. The effect of changing the prediction probability threshold on M^2 score. The model is CNN+kana+emb trained with BCCWJ+Lang8. The best $F_{0.5}$ is 46.4, precision is 52.7, and recall is 31.4 when threshold is 0.7.

applied to the words in the confusion set in order, and the LM selected the most probable word.

V. RESULTS

For evaluation, We used the MaxMatch score (M^2) which is used commonly in GEC task. Table III shows the results of particle error correction on the NAIST Goyo corpus. “+emb” means that the model was initialized with pre-trained word embeddings. Further, “+kana” means that the training and validation data were converted into Hiragana.

The CNN classification model trained with BCCWJ outperformed the language model and achieved 31.7 $F_{0.5}$. After combining Lang-8 with BCCWJ, our model achieved 33.7 $F_{0.5}$. After applying “+kana,” Recall improved but Precision decreased slightly. After applying “+emb,” Recall and Precision improved and our model achieved 35.8 $F_{0.5}$. Finally, when both “+kana” and “+emb” were applied, our model achieved 36.1 $F_{0.5}$.

Imamura et al. [19] showed the results of Japanese particle error correction from their study, and their best score was 28.5 F_1 . Although our model achieved 41.5 F_1 , it is not correct to compare it with their system because the test data are different and their learner corpus is not publicly available. Mizumoto et al. [18] presented the results of a Japanese learners’ error correction study, with no limitation of error types. They also presented the results of Recall by error type, and it was 6.7% for particle errors. In comparison, we have demonstrated better performance in the limitation of the particles.

Table IV
OUTPUT EXAMPLES

No.	Source sentence	Predict	Answer
1	私にとって、日本に来てから、たばこのことが一つ<φ>問題である。 (ref: For me, smoking has been a problem since coming to Japan.)	の	の
2	だから、たばこ<の>吸う人がたくさんいる。 (ref: So there are a lot of people who smoke.)	を	を
3	賛成の理由<は>次に述べさせていただく。 (ref: We explain the following reasons for agreeing.)	は	を
4	他の人の気持ち<は>悪くなった。 (ref: Other people's feelings got worse.)	は	が
5	楽しい時、たばこを一服すったら、気分がもっとよく<に>なります。 (ref: I feel better if I smoke a cigarette when I have fun.)	に	φ
6	以上<φ>私の意見です。 (ref: The above is my opinion.)	φ	が

VI. DISCUSSIONS

A. Deletion rate

As we have described in section III-D, the deletion rate D decides the quantity of TARGET label that will be inserted while building the training data. Figure 2 shows the effect of changing D values on the M^2 score. D = 0.0 means the model does not learn deletion, and D = 1.0 means the model does not learn substitution and insertion. When D = 0.4, it results in the best $F_{0.5}$ score (the score is shown in table III, at the last row).

After analyzing the training data, we found that actual D was 0.69. Therefore, if there is no restriction, the rate is high at which labels are inserted. When D = 0.7, the score decreased by 2.9 $F_{0.5}$ (33.2 $F_{0.5}$). Therefore, it is better to consider the balance of labeling in the building of training data.

B. Prediction probability threshold

The results of table III show that Recall is higher than Precision, but $F_{0.5}$ gives more weight to Precision than Recall. To enhance $F_{0.5}$, we suppressed the corrections with low likelihood. Figure 3 shows the effect of changing the prediction probability threshold on the M^2 score. When the threshold is 1.0, the model does not change at all times because it does not exceed that threshold. The higher the threshold, the higher the Precision and the lower the Recall. This is a trade-off, and it can be seen that the threshold can control this balance. When the threshold is 0.7, our model achieved **52.7** Precision, **31.4** Recall and **46.4** $F_{0.5}$.

C. Output examples

Table IV shows our output examples with setting on VI-B. The target of the correction is enclosed in brackets. Examples 1 and 2 illustrate successful insertions/substitutions. Example 3 illustrates incorrect prediction, but it is acceptable because of being grammatically correct. Examples 4, 5, and 6 illustrate incorrect predictions that are not changed. We need to make the model more aware of errors. For example, it is possible to include error words without masking in the input or to consider the error pattern of the learner.

VII. CONCLUSION

We present a grammatical error correction system for Japanese particles based on the classification method. The CNN classification model is built by training native data and corrected learner sentences that are labeled according to our defined method. In the input sentence, the positions to be predicted are detected first, and the model predicts iteratively. Our best model that was set on a high prediction probability threshold achieved 46.4 $F_{0.5}$ on the NAIST Goyo corpus. Our method repeats prediction in order from the front, however considering multiple predictions at once may improve the performance. Furthermore, our method can be extended to other error types.

ACKNOWLEDGMENT

This work was supported in part by JSPS KAKENHI Grants-in-Aid for Scientific Research (B) Grant ID 15H03216.

REFERENCES

- [1] Y. Zhang and B. Wallace, “A sensitivity analysis of (and practitioners’ guide to) convolutional neural networks for sentence classification,” in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Taipei, Taiwan: Asian Federation of Natural Language Processing, Nov. 2017, pp. 253–263.
- [2] Z. Xie, G. Genthial, S. Xie, A. Ng, and D. Jurafsky, “Noising and denoising natural language: Diverse back-translation for grammar correction,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 619–628.
- [3] S. Chollampatt and H. T. Ng, “Neural quality estimation of grammatical error correction,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, Oct.-Nov. 2018, pp. 2528–2539.
- [4] M. Junczys-Dowmunt, R. Grundkiewicz, S. Guha, and K. Heafield, “Approaching neural grammatical error correction as a low-resource machine translation task,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*:

- Human Language Technologies, Volume 1 (Long Papers).* New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 595–606.
- [5] M. Junczys-Dowmunt and R. Grundkiewicz, “Phrase-based machine translation is state-of-the-art for automatic grammatical error correction,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, November 2016, pp. 1546–1556.
- [6] Z. Yuan and T. Briscoe, “Grammatical error correction using neural machine translation,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2016, pp. 380–386.
- [7] S. Chollampatt and H. T. Ng, “A multilayer convolutional encoder-decoder neural network for grammatical error correction,” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, February 2018.
- [8] R. Grundkiewicz and M. Junczys-Dowmunt, “Near human-level performance in grammatical error correction with hybrid machine translation,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics, 2018, pp. 284–290.
- [9] W. Zhao, L. Wang, K. Shen, R. Jia, and J. Liu, “Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 156–165.
- [10] F. Stahlberg, C. Bryant, and B. Byrne, “Neural grammatical error correction with finite state transducers,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4033–4039.
- [11] M. Junczys-Dowmunt and R. Grundkiewicz, “The AMU system in the CoNLL-2014 shared task: Grammatical error correction by data-intensive and feature-rich statistical machine translation,” in *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*. Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 25–33.
- [12] A. Rozovskaya, K.-W. Chang, M. Sammons, D. Roth, and N. Habash, “The Illinois-Columbia system in the ConLL-2014 shared task,” in *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*. Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 34–42.
- [13] M. Felice, Z. Yuan, Ø. E. Andersen, H. Yannakoudakis, and E. Kochmar, “Grammatical error correction using hybrid systems and type filtering,” in *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*. Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 15–24.
- [14] C. Bryant and T. Briscoe, “Language model based grammatical error correction without annotated training data,” in *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, 2018, pp. 247–253.
- [15] A. Rozovskaya and D. Roth, “Grammatical error correction: Machine translation and classifiers,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 2205–2215.
- [16] C. Wang, R. Li, and H. Lin, “Deep context model for grammatical error correction,” in *Proc. 7th ISCA Workshop on Speech and Language Technology in Education*, 2017, pp. 167–171.
- [17] Z. Kaili, C. Wang, R. Li, Y. Liu, T. Hu, and H. Lin, “A simple but effective classification model for grammatical error correction,” *arXiv preprint arXiv:1807.00488*, 2018.
- [18] T. Mizumoto, M. Komachi, M. Nagata, and Y. Matsumoto, “Mining revision log of language learning SNS for automated Japanese error correction of second language learners,” in *Proceedings of 5th International Joint Conference on Natural Language Processing*. Chiang Mai, Thailand: Asian Federation of Natural Language Processing, Nov. 2011, pp. 147–155.
- [19] K. Imamura, K. Saito, K. Sadamitsu, and H. Nishikawa, “Grammar error correction using pseudo-error sentences and domain adaptation,” in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Jeju Island, Korea: Association for Computational Linguistics, Jul. 2012, pp. 388–392.
- [20] H. Oyama, M. Komachi, and Y. Matsumoto, “Hierarchical annotation and automatic error-type classification of Japanese language learners’ writing,” *Journal of Natural Language Processing*, vol. 23, no. 2, pp. 195–225, 2016.
- [21] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML’10. USA: Omnipress, 2010, pp. 807–814.
- [22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.
- [23] H. Shinnou, M. Asahara, K. Komiya, and M. Sasaki, “nwjc2vec: Word embedding data constructed from ninjal web Japanese corpus,” *Journal of Natural Language Processing*, vol. 24, no. 5, pp. 705–720, 2017.
- [24] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 3111–3119.
- [25] K. Heafield, “KenLM: faster and smaller language model queries,” in *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland, United Kingdom, July 2011, pp. 187–197.

Effect of Preprocessing for Distributed Representations: Case Study of Japanese Radiology Reports

Taro Tada and Kazuhide Yamamoto
Nagaoka University of Technology
Nagaoka, Niigata, Japan
{tada, yamamoto}@jnlp.org

Abstract—A radiology report is a medical document based on an examination image in a hospital. However, the preparation of this report is a burden on busy physicians. To support them, a retrieval system of past documents to prepare radiology reports is required. In recent years, distributed representation has been used in various NLP tasks and its usefulness has been demonstrated. However, there is not much research about Japanese medical documents that use distributed representations. In this study, we investigate preprocessing on a retrieval system with a distributed representation of the radiology report, as a first step. As a result, we confirmed that in word segmentation using Morphological analyzer and dictionaries, medical terms in radiology reports are not handled as long nouns, but are more effective as shorter nouns like subwords. We also confirmed that text segmentation by SentencePiece to obtain sentence distributed representation reflects more sentence characteristics. Furthermore, by removing some phrases from the radiology report based on frequency, we were able to reflect the characteristics of the document and avoid unnecessary high similarity between documents. It was confirmed that preprocessing was effective in this task.

Keywords-Japanese; Medical document; Radiology report; Distributed representation; Pre-processing;

I. INTRODUCTION

Distributed representation has shown its usefulness in various tasks in the natural language processing field. Moreover, there are many studies on distributed representations in the biomedical domain [1] [2] [3]. However, in Japanese medical documents, distributed representations are used for some individual tasks, and there are not a very large number of studies as a whole. In Japan, attempts to digitize medical documents and their secondary use are underway; access to medical documents is generally difficult. Moreover, there are almost no medical domain data sets or corpus in the Japanese language that can be easily accessed by the general public, such as PubMed¹.

In this study, we used anonymous radiology reports. A radiology report is a report created after image examinations at hospitals. However, preparing this report is a burden for doctors, and they require similar past reports as a reference that include images and diagnosis.

Although there are similar document retrieval research works in the medical domain in Japanese, they have not been actively reported. Furthermore, there are only a few reports using distributed representation. Therefore, it is not known how to handle medical terms in Japanese,

where it is necessary to perform word segmentation for tokenization. In addition, it is also unclear what kind of preprocessing is suitable because there are few studies on the basis of Japanese radiology reports.

In this study, we investigate the preprocessing for a similar document search in radiology reports using a simple method. We observed a tendency to obtain a better distributed representation in medical documents, especially in interpretation reports, by generating each document vector with the distributed representation method and evaluating the similarity between the documents.

臨床診断	胃癌。	
検査目的	上記精査。	
臨床情報	胃癌で化学療法を予定。胃体部～噴門、前壁～小弯に癌腫あり。 単純 CT で LN 転移。	
部位/モルヒネ	胸部～骨盤／ CT	検査日時 20XX 年 XX 月 XX 日 XX:XX
検査実施 医療機関	○○ 病院	造影剤 オイバロミン
依頼医/依頼科	△△ △△ / 外科	病棟 外来
希望対応	通常	
読影担当医	○○ ○○	
	前回 CT 検査(20XX/XX/XX)と比較しました。	
	胃噴門部から胃体 上部にかけてほぼ全周性に胃壁の肥厚を認めます。明らかな周囲組織への直接浸潤を疑う所見は認めません。T2-3 相当と思われます。胃吻合部から小弯部にかけて認める多発リンパ節腫大(#1～#3)は最大のもので径 42×52mm とサイズに変化はありません。7 個以上は認めます(N3a、UICC 7th)。	
	肝外側区には囊胞を疑う低吸収域を認めます。明らかな肝転移を疑う腫瘍は認めません。 その他、腹腔内臟器に転移を疑う所見はありません。	
	胆囊底部には萎縮、壁肥厚を認めます。局所的慢性炎症を疑います。内部に結石を認めます。 上行結腸には多発憩室を認めます。	
	単純 CT 上、明らかな転移を疑う所見はありません。播種を疑う結節および腹水は認めません。 肺野に明らかな結節、浸潤影は指摘できません。	
	有意なサイズの綱隔リンパ節腫大、その他の綱隔器質病変はみられません。胸水ありません。	
	骨粗大病変は認めません。	
	その他、有意な所見はありません。	
読影診断	胃癌、多発リンパ節転移疑い(T2-3, N3a, M0:UICC 7th)	
読影医:	○○ ○○	



Figure 1. An example of Japanese radiology report [4]

II. JAPANESE RADIOLOGY REPORTS

The medical documents dealt with in this study are anonymized radiology reports and examination order documents shared from Y's Reading inc. An examination order document is a document in which a doctor who examined a patient describes the findings and the purpose

¹<https://www.ncbi.nlm.nih.gov/pubmed/>

of an examination of the patient's symptoms. An examination department, such as the radiology department, in a hospital, performs an examination based on the examination order document, and a doctor from the examination department describes the findings and diagnoses based on the photographed image. [4, Figure 1] shows an example of a radiology report.

There are two types of radiology report: primary and secondary radiology reports. The primary radiology report is written in reference to the examination image. The secondary report is created with reference to the primary report and the examination image. The secondary report, therefore, provides double-checking of radiology images together with the primary report. After the secondary report is prepared, it is submitted to the patient's doctor, and the doctor makes the final diagnosis. These three types of documents are written based on free description.

In departments such as radiology department that perform image examinations, the number of examinations per day is large, and in the case of examinations through CT (computed tomography) scan, the number of examination images per process may be very large, even several hundred. The preparation of such radiology reports is a burden on the doctors [5]. Therefore, in order to reduce the burden on the preparation of Japanese radiology reports, we investigate the preprocessing of a similar document retrieval system using distributed representations.

III. RELATED WORKS

Research on natural language processing in medical domain documents, such as electronic medical records, there are many available resources, and many studies are presented at various conferences in English [6]. Moreover, there are many studies on radiology reports in English [7] [8]. However, studies using Japanese medical domain documents are not as frequent as those in English documents.

Research on retrieval systems for Japanese medical domain documents has been conducted before, however, there are not many. Ono et al. [9], Okamoto et al. [10], Doi et al. [11], Aramaki et al. [12] are representative. This is attributed to the fact that medical document data itself is difficult to obtain, and it is not so long after the secondary usage activities began [13]. In addition, the need for data anonymization, understanding the purpose and features of documents, and the appearance of domain-specific terminology also makes it difficult to handle documents without the cooperation of medical workers. For these reasons, the treatment of medical documents is complicated, which is considered to be a barrier in its research.

As prior research of similar document search in the medical domain, Okamoto et al. and Doi et al. searched for similar documents using TF-IDF. Okamoto et al. classified information of Observation, Diagnosis, and Treatment that are added to the sentences in medical records and are applied for similar document search. Doi et al. reported searching similar documents using a document vector space method based on TF-IDF. Ono et al. created a

dendrogram of a maximum spanning tree from word weighting by TF-IDF and the ICD (International Classification of Diseases / International Statistical Classification of Diseases and Related Health Problems) code applied for classification of hospital discharge summaries. Aramaki et al. has been operating and reporting a disease case search system that performs a full-text search based on GETA(Generic Engine for Transposable Association), which is an associative search engine, for seven years.

Siddharth et al. [14] proposed a text preprocessing, that reduces the complexity of sentences in biomedical abstracts to improve the performance of syntactic parsers in English.

IV. DATA

The data used are text data of 181,875 examinations. A examination order document, a primary radiology report, and a secondary radiology report are contained for each examination. Table I shows the statistical information of the radiology report and examination order document.

Table I
STATISTICAL INFORMATION OF RADIOLOGY REPORTS AND EXAMINATION ORDER

Type of document	Average document length (char.)
examination order	99.1
primary radiology report	299.3
secondary radiology report	332.3

In the radiology reports, the diagnosis is described for all the parts appearing in the photographed examination image. Suspected positive diseases and injuries will be described even if it is not the original purpose of the imaging examination. Radiology reports with single diseases are few, and those with multiple diseases appear in combination under findings and diagnoses in the report. Even if observing different diseases are the main purpose of the examination, there are cases wherein the same diseases are observed in reports. Furthermore, it is difficult to determine the degree of similarity between each disease. Therefore, it is difficult to judge the superiority or inferiority of the document similarity unless it is handled by a person with advanced medical knowledge, such as a doctor. Moreover, as per our knowledge, there is no data set on the Japanese radiology report that can be used to evaluate document similarity.

In this study, as a first step, we created a data set to simplify the problem. We investigated the effects of pre-processing by checking whether radiology reports included the same disease can be obtained as similar documents. The evaluation data set was created with reference to the study by Okamoto et al. [10].

First, we selected eight diseases (Alzheimer's disease, lung cancer, myocardial infarction, fatty liver, disc herniation, medial collateral ligament injury, Elbow fracture, Achilles tendon injury) as classes. Next, we selected radiology reports with selected the diseases in the above-mentioned. In selecting documents, different classes do not contain the same disease.

A total of 80 documents were selected, 10 documents in each disease class, as data for evaluation, although it is a small sample set. The documents for the 181, 795 examinations were used as training data, excluding the examination documents used for the evaluation data.

V. PREPROCESSING

We investigated the effect of preprocessing on the similarity between the documents. This process involved, first, word segmentation with different granularity for radiology reports; second, cleaning by removing frequently appearing phrases from the evaluation data.

Because the Japanese language is not described with a space between words, word segmentation is required as preprocessing for input to various models. These segmented words become system inputs, and the granularity of the segmentation may affect subsequent processes. In addition, as mentioned above, there are few studies on Japanese radiology reports; therefore, we investigated the effects of phrase frequency-based cleaning. The neologdn² was used to normalize characters before performing the above procedure.

A. Word segmentation

To investigate the effect of word segmentation with different granularities, word segmentation was performed using Japanese morpheme analyzers MeCab [15] and SentencePiece [16]. MeCab³ can select a dictionary for use in word segmentation.

In this article, we used five dictionaries, including two medical terminology dictionaries. We used UniDic⁴, IPADic⁵, and ipadic-neologd⁶ as general domain dictionaries, and ComeJisyo⁷ and MANBYO Dictionary⁸ as medical terminology dictionaries. These medical terminology dictionaries are combined with IPADic to supplement non-medical terminologies. Table II lists the examples of diseases that have been word segmented using MeCab with each dictionary. The medical terminology dictionaries can medical domain terminology to be treated as words with a relatively longer granularity than general domain dictionaries. The general domain dictionaries divide the medical terms into shorter granularities.

B. Word segmentation using SentencePiece

SentencePiece⁹ can perform frequency-based segmentation, which is different from the dictionary-based word segmentation in terms of granularity. Therefore, we investigated the effect of segmentation using SentencePiece. Because SentencePiece can split text by specifying the number of unique tokens, we performed experiments on multiple unique token sizes. Other SentencePiece parameters were used as default.

²<https://github.com/ikegami-yukino/neologdn>

³<https://taku910.github.io/mecab/>

⁴<https://unidic.ninjal.ac.jp/>

⁵<https://taku910.github.io/mecab/>

⁶<https://github.com/neologd/mecab-ipadic-neologd>

⁷<https://ja.osdn.net/projects/comedic/>

⁸<http://sociocom.jp/~data/2018-manbyo/index.html>

⁹<https://github.com/google/sentencepiece>

Table II
EXAMPLE OF WORD SEGMENTATION DIFFERENCE BY MECAB WITH DICTIONARIES

dictionary	segmentation results
example1 : 右肺上葉切除術後	(After upper right lung lobectomy)
UniDic	右肺 上葉 切除 術後
IPADic,IPADic-NEologd	右 肺 上 葉 切除 術 後
MANBYO Dictionary, ComeJisyo	右 肺 上葉 切除 術 後
example2 : 陳旧性心筋梗塞	(old myocardial infarction)
UniDic	陳 旧 性 心 筋 梗 塞
IPADic,IPADic-NEologd	陳 旧 性 心 筋 梗 塞
MANBYO Dictionary, ComeJisyo	陳 旧 性 心 筋 梗 塞

C. Removing high-frequency phrases

As described in Section II, as a feature of the radiology report, the sentences of the findings are freely described in the radiology report for all parts that appear in examination images, even if they are not required for the original examination purposes.

In the radiology report, most of the findings and diagnoses of the areas not intended for examination are negative. These are written as sentences with body parts appear continuously (examples: no significant findings are found in liver, bile, pancreas, spleen) or semi-fixed sentences (example: no other significant findings). Depending on the body parts of the examination target and examination method, the sentences will appear in reports. In addition, because the radiology report is a report of an imaging examination, fixed sentences (such as greetings to the report receiver on the beginning or end of the finding) that are not related to the findings and diagnoses also appear.

As a result, it is assumed that the similarity unnecessarily increases when calculating the similarity between documents. Therefore, we checked for high-frequency phrases from the training data and manually selected phrases that were noisy and removed them from the evaluation data. Table III shows an example of the phrases removed from the evaluation data set.

Table III
EXAMPLE OF REMOVED PHRASES

胆、(bile,) / 肝、(liver,) / その他、(other,) /
明らかな異常を認めません。(There is no obvious abnormality.) /
胸水みられません。(no pleural effusion) /
有意な所見はありません。(no significant findings.) /
ご報告申し上げます。(I'd like to inform you.) /
前回同様です。(same as last time.) /

VI. EXPERIMENTS

Apply the preprocessing described in Section V to the training data and the collected evaluation data. A document vector is generated from each document of the evaluation data, and the similarity between each document is calculated to check the effect of preprocessing.

For each document vector of the evaluation data, the degree of similarity between the document vectors of the

remaining 79 documents in the evaluation data is calculated. Then, it is determined how many of the remaining nine documents in the same disease class are included in the top nine similar documents. This is performed for all the documents of the evaluation data. As a distributed representation method, a document vector is created using fastText [17], [18] with skip-gram, and the degree of similarity between the documents is calculated using cosine similarity.

fastText is the original implementation¹⁰ by Facebook Inc. for investigating the effect of word segmentation by MeCab with different dictionaries; Doc2Vec [19] with DBow was also employed. Doc2Vec is implemented with gensim¹¹. The hyperparameter of Doc2Vec was determined with reference to Lau et al. [20] [21]. Table IV shows the hyperparameters during learning.

Table IV
HYPER PARAMETERS

Hyper parameters (other parameters are default)	
Doc2Vec	Method : DBow, Dim : 300 Window Size : 30, Min Count : 5 Sub-sampling : 10^{-5} Negative sampling : 10, Epoch : 1000
fastText	Method : skipgram, Dim : 300 Window Size : 30, Min Count : 1 Negative sampling : 10, Epoch : 30 Loss : hs

VII. RESULTS AND DISCUSSION

Table V shows the statistical information of the radiology report and examination order document after word segmentation, and Table VI shows the experimental results using MeCab and five different dictionaries. Unexpectedly, the accuracy of the results was improved by segmenting long medical terms into shorter words. This can be assumed because the short granularity of words reduced the number of vocabularies and increased the frequency of words by treating the medical terms as a group of words with finer granularity, like English subwords. However, single-character segmentation results in lower accuracy. In this case, even if each Kanji character has a meaning, it is likely that the meaning can be retained more by learning with short words. A similar trend was observed in fastText and Doc2Vec. These results may depend on the amount of data, but the data used in this experiment are relatively large compared to the actual Japanese medical texts available at present. Therefore, similar trends are likely to be obtained in many cases. On this result, the accuracy of fastText being lower than the that of Doc2Vec, because there was not any hyper-parameter tuning done of fastText. therefore accuracies of between methods are not comparable.

The effect of text segmentation by SentencePiece is shown in Table VII. The number of unique tokens of SentencePiece was the highest result in 4000, exceeding the accuracy of experimental results with MeCab and

dictionaries. Some of the tokens in the documents that were divided using SentencePiece are phrases. Therefore, if medical terms are required to, for example, confirm the similarities between diseases during postprocessing, it is better to divide using MeCab with dictionaries. However, for obtaining a vector of a document, a better distributed representation can be obtained through dictionary-based word segmentation.

Table VIII shows the effect of cleaning, in which high-frequency phrases obtained from the learning data are removed from the evaluation data. As a result, it was confirmed that cleaning by phrase removing based on frequency and little domain knowledge was effective. In the case of medical documents with some fixed elements, high-frequency phrases unnecessarily increase the similarity between documents. We confirmed that removing of these phrases led to the acquisition of distributed representations that better reflect the characteristics of the documents.

In the case where documents of the same disease class were not obtained as similar documents in the experiment, some diseases were not included in the disease classes in the evaluation data and they unintentionally appeared over multiple classes. However, a high degree of similarity between the same disease documents is desirable, and in this respect, it can be said that the document vector reflects the contents of the document more accurately. In addition, based on the fact that they are influenced by words such as "soft tissue" and "fracture" that commonly appear when describing the findings of injuries and diseases, these documents can be said to be similar in a broad sense but require a different approach.

VIII. CONCLUSION

In this study, we investigated the preprocessing method for using distributed representations method for Japanese radiology reports, the effect of word segmentation granularity and cleaning based on frequently appearing phrases.

Preprocessing can have a great effect on obtaining distributed representations from radiology reports. Unexpectedly, it was more effective to divide the medical terms appearing in the radiology report into short words rather than being treated as long words, using a medical terminology dictionary. Moreover, the accuracy in the experimental result with word segmentation by SentencePiece is higher than morphological analyzer. It can be lead to consider that frequencies of words in the training data are very important for the learning of distributed representations better.

For cleaning the evaluation data, we used high-frequency phrases in the training data and selected high-frequency phrases based on a little domain knowledge and a little manual labor. We confirmed that this method could remove factors that unnecessarily increase the degree of similarity between documents, and reflect more document characteristics to the distributed representations.

The preprocessing in this study is common in many tasks using distributed representations, therefore it can be

¹⁰<https://fasttext.cc/>

¹¹<https://radimrehurek.com/gensim/index.html>

Table V
STATISTICAL INFORMATION OF RADIOLOGY REPORTS AND EXAMINATION ORDER

Segmentation method	Average document length (word)			Average token length (character)		
	primary reports	secondary reports	examination orders	primary reports	secondary reports	examination orders
character segmentation	299.3	332.3	99.1	1	1	1
UniDic	212.2	237.0	68.2	1.41	1.40	1.45
IPADic	210.5	234.9	63.7	1.42	1.41	1.56
IPADic-neologd	202.9	226.7	60.2	1.48	1.47	1.65
MANBYO Dictionary	195.9	219.3	60.2	1.53	1.52	1.65
ComeJisyo	188.9	211.9	57.5	1.58	1.57	1.72
SentencePiece(4000)	132.8	147.6	59.4	2.25	2.25	1.67

Table VI
ACCURACIES OF FASTTEXT AND DOC2VEC IN CHANGING SEGMENTATION METHODS

segmentation method	fastText	Doc2Vec
character segmentation	0.685	0.799
UniDic	0.708	0.894
IPADic	0.704	0.908
ipadic-neologd	0.663	0.896
MANBYO Dictionary	0.644	0.828
ComeJisyo	0.613	0.869

Table VII
ACCURACIES OF FASTTEXT WITH SENTENCEPIECE SEGMENTATION

unique tokens	fastText	unique tokens	fastText
3000	0.836	16000	0.786
4000	0.847	18000	0.797
5000	0.824	20000	0.797
6000	0.832	25000	0.792
8000	0.819	30000	0.799
10000	0.772	40000	0.807
12000	0.769	100000	0.814
14000	0.781		

applied to various tasks.

IX. FUTURE WORK

Medical documents we used in this work were examination order documents, primary radiology reports, and secondary radiology reports. We confirmed a problem in a part of the word-level similarity ("handball" as upper similar words of "metastasis", etc.) that due to the influence of examination order documents. Probably, better distributed representations can be obtained by filtering the documents used for learning. Besides, it might be possible to learn relationships between medical terms using external documents.

In this study, we simplified the problem as the first step toward the similar document retrieval task of radiology reports. However, owing to the time and cost, the scale of the evaluation data was small. Because larger-scale evaluation data are required for considering overfitting risk. therefore we would like to create a new data set. In addition, we intend to collaborate with medical professionals with advanced medical knowledge in order to set up tasks closer to medical sites.

ACKNOWLEDGMENT

The authors are grateful to Y's Reading inc. for giving incisive comments and sharing anonymized text data of examination documents.

Table VIII
EFFECT OF PHRASE REMOVING (ACCURACY)

Method	fastText	Doc2Vec	Avg. length (words)
IPADic	0.704	0.908	151.4
IPADic with removing	0.781	0.922	130.7

This work was supported in part by JSPS KAKENHI Grants-in-Aid for Challenging Research (Exploratory) Grant ID 17K18481.

REFERENCES

- [1] Dieter Galea,Ivan Laponogov,Kirill Veselkov,Sub-word information in pre-trained biomedical word representations: evaluation and hyper-parameter optimization,Proceedings of the BioNLP 2018 workshop, pp.56-66 2018
- [2] Chen, Qingyu and Peng, Yifan and lu, Zhiyong BioSentVec: creating sentence embeddings for biomedical texts, he Seventh IEEE International Conference on Healthcare Informatics (ICHI 2019),2019
- [3] Alsentzer Emily, Murphy John, Boag William, Weng Wei-Hung, Jindi Di, Naumann Tristan, McDermott Matthew, Publicly Available Clinical BERT Embeddings, Proceedings of the 2nd Clinical Natural Language Processing Workshop, pp.72-78 2019
- [4] Y's Reading, Our reports, Accessed on: Jul. 1, 2019. [Online]. Available <https://ys-reporting.com/reports/>
- [5] Kyoko Makino, Rumi Hayakawa, Koichi Terai, Hiroshi Fukatsu, Development and Evaluation of a Diagnostic Documentation Support System using Knowledge Processing, Transactions of the Japanese Society for Artificial Intelligence, Vol.23 No.6 pp.463-472 2008
- [6] Pranjul Yadav, Michael Steinbach, Vipin Kumar, Gyorgy Simon, Mining Electronic Health Records (EHR): A Survey, ACM Computing Surveys, Vol. 1, No. 1, Article 1, Publication date: April 2016.
- [7] Banerjee I,Madhavan S,Goldman RE,Rubin DL,Intelligent Word Embeddings of Free-Text Radiology Reports, AMIA Annu Symp Proc. pp.411420 2017,Published 2018 Apr 16.
- [8] Yifan Peng, Ke Yan, Veit Sandfort, Ronald M. Summers, Zhiyong Lu, "A self-attention based deep learning method for lesion attribute detection from CT reports," IEEE International Conference on Healthcare Informatics (ICHI), 2019
- [9] Hiroki ONO, Katsuhiko TAKABAYASHI, Takahiro SUZUKI, Hideto YOKOI, Atsushi IMIYA, Youichi SATOMURA, Classification of Discharge Summaries by Text Mining, Japan Journal of Medical Informatics, Vol.24 No.1 pp.35-44 2004

- [10] Kazuya OKAMOTO and Tadamasa TAKEMURA and Tomohiro KURODA, Keisuke NAGASE and Hiroyuki YOSHIIHARA, Context-based Retrieval System for Similar Medical Practice Documents, Transactions of Japanese Society for Medical and Biological Engineering, Transactions of Japanese Society for Medical and Biological Engineering, Vol.49 No.6 pp.199-206 2006
- [11] Shunsuke DOI, Takashi KIMURA, Masaki SEKINE, Takahiro SUZUKI, Katsuhiko TAKABAYASHI, Toshiyo TAMURA, Management and Evaluation of Similar Case Searching System in Homepage of Medical Society, Transactions of Japanese Society for Medical and Biological Engineering, Vol.49 No. 6 pp. 870-876 2011
- [12] Aramaki E, Iwao T, Wakamiya S, Ito K, Yano K, Ohe K, A Fundamental Study on User Utilization Based on a Trial Operation of the Medical Case Retrieval System, Japan Journal of Medical Informatics, Vol.38 No.4 pp. 245-256 2018
- [13] Toshihiro TAKEDA and Shirou MANABE and Yasushi MATSUMURA, The Current Situation and Issues of the Secondary Use of Electronic Medical Record Data, Transactions of Japanese Society for Medical and Biological Engineering, Vol.55 No.4 pp. 151-158 2017
- [14] Jonnalagadda Siddhartha, Tari Luis, Hakenberg Jörg, Baral Chitta, Gonzalez Graciela, Towards Effective Sentence Simplification for Automatic Processing of Biomedical Text, Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers, pp.177-180 2009
- [15] Taku Kudo, Kaoru Yamamoto, Yuji Matsumoto, Applying Conditional Random Fields to Japanese Morphological Analysis, Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pp.230-237 2004
- [16] Taku Kudo and John Richardson, SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing, Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (System Demonstrations), pp.66-71 2018
- [17] Piotr Bojanowski, Edouard Grave, Armand Joulin and Tomas Mikolov, Enriching Word Vectors with Subword Information, Transactions of the Association for Computational Linguistics, vol.5 pp. 135-146 2017
- [18] Armand Joulin, Edouard Grave, Piotr Bojanowski, Tomas Mikolov, Bag of Tricks for Efficient Text Classification, Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Volume 2, Short Papers, pp. 427-431 2017
- [19] Quoc Le, Tomas Mikolov, Distributed Representations of Sentences and Documents, Proceedings of the 31 st International Conference on Machine Learning, JMLR: W&CP volume 32, pp. 1188-1196 ,2014
- [20] Jey Han Lau and Timothy Baldwin, An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation, Proceedings of the 1st Workshop on Representation Learning for NLP, pp. 78-86 2016
- [21] Billy Chiu,Gamal Crichton,Anna Korhonen,Sampo Pyysalo, How to Train Good Word Embeddings for Biomedical NLP, Proceedings of the 15th Workshop on Biomedical Natural Language Processing, pp.166174 2016

Character Decomposition for Japanese-Chinese Character-Level Neural Machine Translation

Jinyi Zhang
*Graduate School of Engineering
 Gifu University
 Gifu, Japan*
Email: zhang@mat.info.gifu-u.ac.jp

Tadahiro Matsumoto
*Faculty of Engineering
 Gifu University
 Gifu, Japan*
Email: tad@gifu-u.ac.jp

Abstract—After years of development, Neural Machine Translation (NMT) has produced richer translation results than ever over various language pairs, becoming a new machine translation model with great potential. For the NMT model, it can only translate words/characters contained in the training data. One problem on NMT is handling of the low-frequency words/characters in the training data. In this paper, we propose a method for removing characters whose frequencies of appearance are less than a given minimum threshold by decomposing such characters into their components and/or pseudo-characters, using the Chinese character decomposition table we made. Experiments of Japanese-to-Chinese and Chinese-to-Japanese NMT with ASPEC-JC (Asian Scientific Paper Excerpt Corpus, Japanese-Chinese) corpus show that the BLEU scores, the training time and the number of parameters are varied with the number of the given minimum thresholds of decomposed characters.

Keywords-character decomposition; neural machine translation; Japanese-Chinese; character-level; LSTM; encoder; decoder;

I. INTRODUCTION

Machine translation's performance has greatly improved from Statistical Machine Translation (SMT), due to the appearance of Neural Machine Translation (NMT). For NMT, one problem is handling of the low-frequency words/characters in the vocabulary of the training data [1]. For the NMT models, as the vocabulary size increases, the computational complexity becomes enormous. Therefore, in a general word-level NMT model, the vocabulary size (the number of different characters) is usually limited to about tens of thousands of words, and the remaining low-frequency words are uniformly treated as unknown words. The increasing of unknown words leads to reduce translation performance, therefore the handling of low frequency words is a big problem in NMT.

Byte Pair Encoding (BPE) made the NMT model capable of open-vocabulary translation by encoding low-frequency and unknown words as sequences of subword units, was proposed by Sennrich et al. [2], to be used to solve the low frequency words' problem.

However, Chinese mainly uses Chinese characters (Hanzi) which are logograms. Many Chinese words are written with one or two Chinese characters, as a result, it is difficult to divide a Chinese word into high-frequency subword units. Therefore, it is considered that the character-level is suitable for NMT between Japanese and Chinese.

For character-level NMT, there is also an advantage that errors and fluctuations do not occur in the process of dividing sentences into words (word segmentation).

Compared with the word-level NMT, the vocabulary size is kept small in character-level NMT, but there are still many characters of extremely low-frequency in the vocabulary. At word-level, the method of replacing a low-frequency word having low statistical reliability with another word of related high-frequency has been attempted [3], but such a substitution is difficult for characters. Therefore, we devised a method for reducing low-frequency characters for character-level NMT between Japanese and Chinese by dividing low-frequency Chinese characters into constituent elements of the character (radicals: traditionally recognized components of Chinese characters) and pseudo partial characters. We investigated the effects of the method on translation results, and the number of the parameters of the model by experiments.

We used Luong's NMT system as the base system [4], which follows an encoder-decoder architecture with global attention at the character level. In our case, we chose the character-level NMT as the baseline, because the character-level NMT between Japanese and Chinese has better translation performance than the word-level NMT.

The main contributions of this paper are the following. We created a Chinese character composition table for finding its constituent elements. We demonstrate the possibility to improve the translation performance of NMT systems by dividing the Chinese and Japanese characters into constituent elements and share them with the other characters in the vocabulary, without changing the neural network architecture. We believe this capability makes our approach applicable to different NMT architectures.

In the remainder of this paper, Section II presents the related work of this paper. Section III gives a brief explanation of the architecture of the NMT that we used as the base system and ASPEC-JC (Asian Scientific Paper Excerpt Corpus, Japanese-Chinese) corpus. Section IV describes the proposed method, how to divide the Chinese and Japanese characters into constituent elements and share them with the other characters in the vocabulary. Section V reports the experimental framework and the results obtained in the Japanese-Chinese and Chinese-Japanese translation (with ASPEC-JC [5]). Finally, Section VI concludes with the contributions of this paper and

further work.

II. RELATED WORK

The characters used in a language are usually much fewer than the words of the language. Character-level neural language models [6] and MT are explored and achieved respective results. Previous works, such as POS tagging [7], name entity recognition [8], parsing [9], learning word representations [10], and character embeddings [11], shown different advantages of using character-level information in Natural Language Processing (NLP).

Besides, subword-based representations (the middle of word-based and character-based representations) have been explored in NMT [2], and are applied to English and other western languages, where most of the words consist of several or a dozen characters. Contrastingly, Chinese characters, which are used in Chinese, Japanese and some other Asian languages, are typical logograms. A logogram is a character that represents a concept or thing, namely a word; and thus, it is difficult to split those words into subwords. Recently, Meng et al. [12] found that character-based models consistently outperform subword-based and word-based models for deep learning-based Chinese NLP works.

For Chinese-Japanese NMT, the sub-character level information improved the translation performance [13], by using sub-character sequences on either the source or target side. However, about their character decomposition, it still needs to be explored. Du and Way [14] trained factored NMT models using “Pinyin” sequences on the source side. Pinyin, is the official romanization system for Chinese. This work only applied to Chinese source-side NMT. Zhang and Matsumoto [15] also attempted to use a factored encoder for Japanese-Chinese NMT system using radical information. They did not achieve good results in Chinese-to-Japanese NMT. Wang et al. [16] directly applied a BPE algorithm to sequences before building NMT models. This method has only been tested in the Chinese-English direction and is not comprehensive enough.

III. NEURAL MACHINE TRANSLATION AND ASPEC-JC CORPUS

A. Neural Machine Translation

NMT completely adopts the neural network approach to compute the conditional probability $p(y|x)$ of the target sentence y for the given source sentence x . We follow the NMT architecture by Luong et al. [4], which we will briefly describe here. This NMT system is implemented as a global attentional encoder-decoder neural network with Long Short-Term Memory (LSTM), and we simply use it at the character level.

The encoder is a bi-directional neural network with LSTM units that reads an input sequence $x = (x_1, \dots, x_m)$ and calculates a forward sequence of hidden states $(\vec{h}_1, \dots, \vec{h}_m)$ and a backward sequence $(\overleftarrow{h}_1, \dots, \overleftarrow{h}_m)$. The hidden states \vec{h}_j and \overleftarrow{h}_j are concatenated to obtain the annotation vector h_j .

The decoder is a recurrent neural network with LSTM units that predicts a target sequence $y = (y_1, \dots, y_n)$. Every word (or character in case of character-level NMT) y_i is predicted based on a recurrent hidden state s_i , the previously predicted word (or character) y_{i-1} , and a context vector c_i . c_i is computed as the weighted sum of the annotations h_j . Finally, the weight of each annotation h_j is computed through an alignment (or attention) model α_{ij} , which models the probability that y_i is aligned to x_j . The forward states of the encoder is expressed as below:

$$\vec{h}_j = \tanh(\vec{W}Ex_j + \vec{U}\vec{h}_{j-1}) \quad (1)$$

where $E \in \mathbb{R}^{m \times V_x}$ is a word embedding matrix, $\vec{W} \in \mathbb{R}^{n \times m}$ and $\vec{U} \in \mathbb{R}^{n \times n}$ are weight matrices; m , n and V_x are the word embedding size, the number of hidden units, and the vocabulary size of the source language, respectively.

B. ASPEC-JC Corpus

We implement our system with the ASPEC-JC corpus, which was constructed by manually translating Japanese scientific papers into Chinese [5]. The Japanese scientific papers are either the property of the Japan Science and Technology Agency (JST) or stored in Japan’s Largest Electronic Journal Platform for Academic Societies (J-STAGE).

ASPEC-JC is composed of three parts: training data (672,315 sentence pairs), development data (2,090 sentence pairs), development-test data (2,148 sentence pairs) and test data (2,107 sentence pairs) on the assumption that it would be used for machine translation research.

ASPEC-JC contains both abstracts and some parts of the body texts. ASPEC-JC only includes “Medicine”, “Information”, “Biology”, “Environmentology”, “Chemistry”, “Materials”, “Agriculture” and “Energy” 8 fields because it was difficult to include all the scientific fields. These fields were selected by investigating the important scientific fields in China and the use tendency of literature databases by researchers and engineers in Japan. In these fields, sentences belonging to the same article are not included.

Compared with other language pairs such as English-French, which usually comprises millions of parallel sentences. ASPEC-JC corpus only has about 672k sentences. Moreover, LSTMs+attention model is usually more robust than the transformer model [17] on smaller datasets, due to the smaller number of parameters [12].

IV. REDUCTION OF LOW-FREQUENCY CHARACTERS BY CHARACTER DECOMPOSITION

During the training and translation process, the training data contains many low-frequency characters that the NMT model cannot translate. The low-frequency characters affect translation performance.

In this research, we decomposed low-frequency characters (mainly Chinese characters) by using high-frequency characters and pseudo-characters, and sharing pseudo-characters among multiple low-frequency characters. We

devised a method to remove characters below a certain frequency and checked the effect on translation performance in the experiment.

The method of decomposing low-frequency characters will be described below.

A. Character Decomposition

Chinese characters are logograms, but some different types could be identified, based on the manner in which they are formed.

They include:

- pictographs: 日 (sun), 月 (moon), 人 (person), 木 (tree),
- simple ideograms: 一 (one), 二 (two), 上 (up), 下 (down),
- compound ideographs: 林 (woods ← tree+tree), 休 (rest ← person+tree), and
- phono-semantic compounds: 銅 (copper ← semantic 金 (metal) + phonetic 同), 河 (river ← semantic 水 (water) + phonetic 可).

Phono-semantic compounds, together with compound ideographs, form over 90% of Chinese characters; accordingly, most Chinese characters consist of two or more (sub-)characters.

Even if a character is rare, its component may be a high-frequency character. For example, 榆 (elm) appears only 16 times in the Japanese sentences of the ASPEC-JC training data, whereas its component 木 (tree) appears 7780 times. If there are other low-frequency characters that have 木 as their components, the frequency of 木 increases more by decomposing the low-frequency characters. In most cases, the higher frequency component (such as 木) of a compound character is a radical, which is related to the meaning of the character.

Our method decomposes low-frequency characters into two partial characters by using the Chinese character decomposition table (Section IV-B). If a character has three or more parts, the method decomposes it into the first part and the rest. Comparing the two components of a character, the component appearing less frequently in the training data is replaced with a pseudo-character, such as s_1, s_2, \dots, s_n .

The appearance frequency of the pseudo-partial characters are increased by sharing them among low-frequency characters as follows:

- | | |
|--|----------------------|
| 榆 (elm tree) | → [木 (wood), s_1] |
| 柂 (a square wooden box used to measure rice) | → [木 (wood), s_2] |
| 炒 (fry) | → [火 (fire), s_1] |
| 焰 (flame) | → [火 (fire), s_2] |

To balance the frequency of the pseudo-characters, we set an upper limit of the number of pseudo-partial characters that are paired with each genuine character component.

If the number exceeds the limit, the method decomposes the character into two pseudo-characters as follow:

- | | |
|-----------------|------------------------|
| 榦 (sakaki tree) | → [s_{13}, s_{16}] |
|-----------------|------------------------|

枷 (cangue) → [s_{19}, s_{22}]

If a low-frequency character cannot be decomposed, it is replaced with a pair of a pseudo-character and 漢 (han) (for Chinese characters), 仮 (assumed) (for Japanese Kana) or 符 (symbol) (for symbols and other characters).

In this way, the method replaces every low-frequency ($\leq k$) character with a pair of a high-frequency character and a pseudo-character or pairs of two pseudo-characters, in order to eliminate such low-frequency characters in the training data.

The mappings from the low-frequency characters to the character pairs are separately created for Japanese and Chinese training data. The training is conducted with the decomposed data. In testing time, low-frequency characters in the source sentences are decomposed with the mapping for the source language before translation. The translated sentences are reconstructed (decoded) with the mapping for the target language. If the reconstruction of a character is failed, the character pair is replaced with the space character.

B. Creation of Chinese Character Composition Table

We created a table for decomposing Chinese characters, based on the Chinese character decomposition table of cjklib¹, the Kanji structure information table of the CHISE project², Jigen³ and the distribution data of the Kanji database project⁴. Our table was created manually.

If there are multiple Chinese characters having the same constituent elements as shown in Table I and Table II, they are distinguished by numbering as follows:

- | | |
|---------------------|-------|
| Examples: 量 (dizzy) | → 日軍1 |
| 暉 (sunshine) | → 日軍2 |
| 柰 (crab-apple) | → 木示1 |
| 标 (label) | → 木示2 |

As the Table III and Table IV show, if the components are decomposed in a simple form, the meaning becomes weak, they are excluded from the table so that decomposition is not performed. In the experiments with the ASPEC-JC corpus, we excluded 102 Kanji out of 3,802 Kanji contained in Japanese sentences, and 204 Hanzi out of 5,576 Hanzi contained in Chinese sentences.

We have manually confirmed the Chinese character composition table, and also uploaded this table to github, hoping that interested people can come up with suggestions for improvement⁵.

V. EVALUATION AND TRANSLATION RESULTS

A. Experiment Settings

We implemented our system using the OpenNMT toolkit [18] with the ASPEC-JC corpus which had already introduced in Section III-B.

¹<http://cjklib.org>

²<http://www.chise.org>

³<http://jigen.net>

⁴<http://kanji-database.sourceforge.net>

⁵<https://github.com/zhang-jinyi/Chinese-Character-Composition-Table>

Table I
EXAMPLES OF JAPANESE KANJI THAT HAVING THE SAME CONSTITUENT ELEMENTS.

Constituent elements	Japanese Kanji
弓 丨	引 (draw), 弓 (hang)
束 束	棘 (sour jujube), 穉 (jujube)
日 軍	暈 (dizzy), 晖 (sunshine)
木 口	束 (bind), 杏 (apricot)
土 襷	壤 (soil), 壤 (soil)
口 貝	唄 (song), 員 (member)
山 峯	峰 (peak), 峯 (peak)

Table II
EXAMPLES OF CHINESE HANZI THAT HAVING THE SAME CONSTITUENT ELEMENTS.

Constituent elements	Chinese Hanzi
𠂔 土	由 (by), 田 (field), 冂 (tender)
木 示	标 (label), 奈 (crab-apple)
𠂔 人	贝 (shellfish), 內 (inside)
口 八	𠂔 (horn), 只 (only)
亼 直	值 (value), 值 (value)
口 贝	唄 (to chant), 員 (member)
𠂔 日	暉 (sunshine), 晕 (dizzy)

Our models have one LSTM layer, with 512 cells, and embedding size is 512. The parameters are uniformly initialized in $(-0.1, 0.1)$, using plain SGD, starting with a learning rate of 1 until epoch 6, and after that, 0.5 times for each epoch. The max-batch size is 100. The normalized gradient is rescaled whenever its norm exceeds 1. The dropout probability is set to 0.5 to avoid overfitting. Decoding is performed by beam search with a beam size of five. The maximum length of a sentence is 250 by default, but it is set to 500 because it becomes much longer at the character level.

We segment the Chinese and Japanese sentences into

Table III
EXAMPLES OF JAPANESE KANJI THAT EXCLUDED FROM THE TABLE.

Constituent elements	Japanese Kanji
𠂔	七 (seven)
一 乂	丈 (measure)
一 卜	下 (below)
丂 𠂔	儿 (son)
月 一	且 (even)
丨 丶	卜 (divination)
十一	土 (soil)

Table IV
EXAMPLES OF CHINESE HANZI THAT EXCLUDED FROM THE TABLE.

Constituent elements	Chinese Hanzi
丂 𠂔	么 (for interrogatives and adverbs)
一 夕	歹 (bad)
𠂔 月 匕	能 (ability)
七十	车 (vehicle)

words by Jieba⁶ and Mecab⁷, respectively.

BiLingual Evaluation Understudy (BLEU) is an algorithm for evaluating the quality of text that has been machine-translated from one natural language to another [19]. BLEU score is calculated with multi-bleu.perl attached to OpenNMT after the word segmentation. In other words, we took the word-level evaluation.

In many cases, validation perplexity (perplexity with dev data) stopped declining in epoch 10 or 11. The average of BLEU scores from that point to epoch 16 was taken as the evaluation BLEU value. The baseline is the character-level translation with the raw training data that does not process anything.

B. Experiment Results and Discussion

Variation of BLEU scores: The low-frequency characters are deleted from the training data by the character decomposition method described in Section IV-A. Figure 1 shows the variation in BLEU scores per epoch.

The least frequency of occurrence of the baseline is 1. The upper limit of the subscript of pseudo-characters was basically set to 55 in both languages, but it was set to 60 when setting the least frequency of occurrence as 7000 to more in Chinese language data, because of the lack of pseudo-characters .

In the Japanese-to-Chinese translation, when the least frequency of occurrence was between 10 and 120, the translation results often exceeded the baseline. Improved about 0.5% when setting the least frequency of occurrence to 20. On the other hand, in the case of Chinese-to-Japanese translation, the translation result was less likely to exceed the baseline, but it improved by about 0.3% when setting the least frequency of occurrence to 150. The results above are not as good as we expected.

The type of decomposed characters are 78% of Chinese Hanzi in Chinese sentences, 47% of Japanese Kana in Japanese sentences, 36% of Japanese Kanji in Japanese sentences in the training data, respectively. It is conceivable that this difference affects the translation results. Unlike Chinese, Japanese Kanji only account for 36% in Japanese, and our method only decomposes Japanese Kanji in Japanese, which caused Japanese to be not fully decomposed. This will result in a certain decline in Japanese translation results in the direction of Chinese-to-Japanese. The decomposed sentences we used as training data became longer than before. This may be a factor that affects the translation results. There is also a possibility that the number of characters contained in the training data (the vocabulary size of the NMT system) also has a huge difference from the 6,088 characters of Chinese to the 4,249 characters of Japanese, on the ASPEC-JC corpus.

Variation of training time: Reducing low-frequency characters decreases the vocabulary size, so the number of parameters to be trained also decreases. As a result, it is expected that the amount of memory used during

⁶<http://github.com/fxsjy/jieba>

⁷<http://taku910.github.io/mecab>

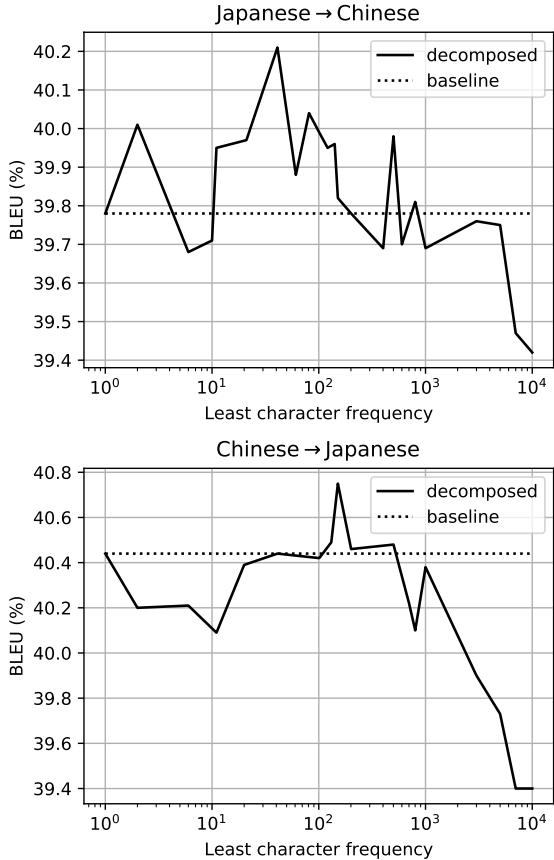


Figure 1. Variation of the BLEU scores

training and the training time will be reduced. Figure 2 shows the number of parameters obtained from the log data in training and the variation in the average training time per epoch. Because experiments were conducted on multiple systems with different configurations of CPU and GPU, the values are relative to the results of the baseline training on each system.

The character decomposition reduces the number of parameters of NMT models, but increases the number of characters. In the Japanese-to-Chinese translation, the training time was always shorter than the baseline until the least frequency of 1000, and the average was 3.56% shorter in the range of the least frequency of 10 to 1000. On the other hand, the effect of shortening the training time was not seen much in Chinese-to-Japanese translation. The decomposed sentences become longer than before. This makes such a result that taking more time to calculate the models.

The results obtained above are dependent on the ASPEC-JC corpus. For different corpora, there should be different threshold (least frequency of occurrence) choices and different translation results.

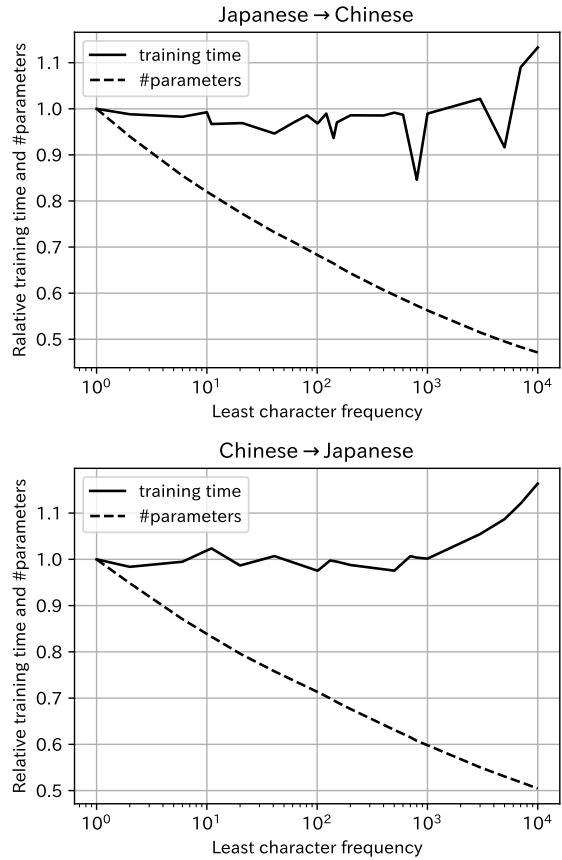


Figure 2. Variation of the training time and the number of parameters

VI. CONCLUSION

In this research, we created a Chinese character composition table and proposed a method to reduce low-frequency characters by decomposing low-frequency characters into Chinese characters' constituent elements and pseudo-characters for NMT between Japanese and Chinese.

Experiments of Japanese-to-Chinese and Chinese-to-Japanese NMT systems showed that the BLEU scores and the training time varied with the number of least frequency of decomposed characters. As a result, compared to the baseline, the BLEU value was about 0.5% higher in Japanese-to-Chinese and 0.3% higher in Chinese-to-Japanese. However, especially in the Chinese-to-Japanese, in most cases, the BLEU scores were lower than the baseline. The translation results are not very well overall. The training time was generally shorter than the baseline when the least frequency of occurrence was less than 1000 in the Japanese-to-Chinese translation experiment.

Because the decomposition of the Chinese characters causes the sentence to grow longer, we should increase the NMT model's support for long sentences, such as using the long sentence segmentation method for NMT [20].

Further, we should use the popular models to train, such as the transformer model [17].

In the future, we should improve or find a better character decomposition method to choose the appropriate least frequency of occurrence for different corpus, even at the character level translation from Chinese to other languages, or from Japanese to other languages. The use of Chinese characters' constituent elements may lead to an improvement for translation performance with fewer parameters and shorter training time.

ACKNOWLEDGMENT

J. Zhang was supported by China Scholarship Council (CSC) under the CSC Grant (No.201708050078). All authors wish to thank the anonymous reviewers and editors for their insightful comments and assistance.

REFERENCES

- [1] P. Koehn and R. Knowles, “Six Challenges for Neural Machine Translation,” in *Proc. of the First Workshop on Neural Machine Translation*, ACL, Vancouver, Canada, 2017, pp 28–39.
- [2] R. Sennrich, B. Haddow, and A. Birch, “Neural machine translation of rare words with subword units,” in *Proc. 54th Annual Meeting of the Assoc. for Computational Linguistics*, Berlin, Germany, 2016, pp 1715-1725.
- [3] R. Chitnis, and J. DeNero, “Variable-Length Word Encodings for Neural Translation Models,” in *Proc. 2015 Conf. on Empirical Methods in Natural Language Processing*, ACL, Lisbon, Portugal, 2015, pp. 2088–2093.
- [4] M. T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation,” in *Proc. 2015 Conf. on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 2015, pp 1412-1421.
- [5] T. Nakazawa, M. Yaguchi, K. Uchimoto, M. Utayama, E. Sumita, S. Kurohashi, and H. Isahara, “ASPEC: Asian scientific paper excerpt corpus,” in *Proc. 9th Int. Conf. on Language Resources and Evaluation (LREC 2016)*, European Language Resources Association (ELRA), Portorož, Slovenia, 2016, pp 2204-2208.
- [6] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, “Character aware neural language models,” in *Proc. 30th AAAI Conf. on Artificial Intelligence*, Phoenix, Arizona, USA, 2016. pp. 2741-2749.
- [7] C. D. Santos and B. Zadrozny, “Learning character-level representations for part-of-speech tagging,” in *The 31st International Conference on Machine Learning*, Beijing, China, 2014, pp. 1818-1826.
- [8] C. D. Santos and V. Guimaraes, “Boosting named entity recognition with neural character embeddings,” in *Proc. Fifth Named Entity Workshop*, ACL, Beijing, China, 2015, pp. 25-33.
- [9] M. Ballesteros, C. Dyer, and N. A. Smith, “Improved transition-based parsing by modeling characters instead of words with lstms,” in *Proc. 2015 Conf. on Empirical Methods in Natural Language Processing*, ACL, Lisbon, Portugal, 2015, pp. 349-359.
- [10] X. Chen, L. Xu, Z. Liu, M. Sun, and H. B. Luan, “Joint learning of character and word embeddings,” in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, AAAI Press, Buenos Aires, Argentina, 2015, pp. 1236-1242.
- [11] Y. Li, W. Li, F. Sun, and S. Li, “Component-enhanced Chinese character Embeddings,” in *Proc. 2015 Conf. on Empirical Methods in Natural Language Processing*, ACL, Lisbon, Portugal, 2015, pp. 829-834.
- [12] Y. Meng, X. Li, X. Sun, Q. Han, A. Yuan and J.Li, “Is Word Segmentation Necessary for Deep Learning of Chinese Representations?” in *Proc. 57th Annual Meeting of the Assoc. for Computational Linguistics*, Florence, Italy, 2019, [Online]. arXiv:1905.05526.
- [13] L. Zhang and M. Komachi, “Neural Machine Translation of Logographic Languages Using Sub-character Level Information,” in *Proceedings of the 3rd Conference on Machine Translation (WMT)*, Brussels, Belgium, 2018, pp. 17–25.
- [14] J. Du and A. Way, “Pinyin as Subword Unit for Chinese-Sourced Neural Machine Translation,” in *Irish Conference on Artificial Intelligence and Cognitive Science*, Dublin, Ireland, 2017.
- [15] J. Zhang and T. Matsumoto, “Improving Character-level Japanese-Chinese Neural Machine Translation with Radicals as an Additional Input Feature,” in *the 21st International Conference on Asian Language Processing (IALP)*, Singapore, 2017, pp. 172-175.
- [16] Y. Wang, L. Zhou, J. Zhang, and C. Zong, “Word, subword or character? an empirical study of granularity in Chinese-English NMT,” in *China Workshop on Machine Translation*, Springer, Dalian, China, 2017, pp. 30–42.
- [17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention Is All You Need,” in *the 31st Conference on Neural Information Processing Systems*, Long Beach, CA, USA, 2017, pp. 5998-6008.
- [18] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, “OpenNMT: Open-source toolkit for neural machine translation,” in *Proc. 55th Annual Meeting of the Assoc. for Computational Linguistics*, Vancouver, Canada, 2017, pp. 67–72.
- [19] K. Papineni, S. Roukos, T. Ward and W. J. Zhu, “BLEU: A Method for Automatic Evaluation of Machine Translation,” in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Philadelphia, PA, USA, 2002, pp. 311-318.
- [20] J. Pouget-Abadie, D. Bahdanau, B. V. Merrienboer, K. Cho, and Y. Bengio, “Overcoming the Curse of Sentence Length for Neural Machine Translation using Automatic Segmentation,” in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, Doha, Qatar, 2014, pp. 78–85.

Japanese grammatical simplification with simplified corpus

Yumeto Inaoka and Kazuhide Yamamoto
Nagaoka University of Technology
Nagaoka, Japan
{inaoka, yamamoto}@jnlp.org

Abstract—We construct a Japanese grammatical simplification corpus and established automatic simplification methods. We compare the conventional machine translation approach, our proposed method, and a hybrid method by automatic and manual evaluation. The results of the automatic evaluation show that the proposed method exhibits a lower score than the machine translation approach; however, the hybrid method garners the highest score. According to those results, the machine translation approach and proposed method present different sentences that can be simplified, while the hybrid version is effective in grammatical simplification.

Keywords-text simplification, paraphrasing, controlled languages

I. INTRODUCTION

The number of foreign residents in Japan has reached approximately 2.64 million, increasing over the past six years¹. Recently, in Japan, guide plates and official documents have often been written in both Japanese and English. However, 56% of foreigners in Japan do not understand English [1], which is more than the foreigners who do not understand Japanese (37%). Therefore, Japanese that is easy for foreigners to understand (Easy Japanese) is presented in the field of humanities².

Meanwhile, automatic simplification is studied in the field of natural language processing. This is a task that automatically converts text with complex vocabulary and grammar into simple text. [2] This technology is useful for foreign language learners, children, the elderly, and disabled people. Text simplification can be regarded as a subtask into various language processing applications because the input and output are the same language. It is known to improve performance by using text simplification as a subtask of syntactic parsers [3] and machine translation [4].

First, we constructed a Japanese grammatical simplification corpus by using crowdsourcing. Crowdsourcing is a method of widely recruiting workers using the Internet and outsourcing the work. With this method, a large amount of work can be completed in a short time and inexpensively. For instance, this method is used in the construction of language resources and manual evaluation of machine translation [5], [6].

Second, we compared simplification methods by using the constructed corpus. In conventional text simplification, the machine translation approach is used. Because the input and output languages are the same, it is not necessary to generate an entire sentence from scratch. Therefore, we compared the following three methods.

- Machine translation approach that is often used in conventional research
- Our proposed method by extracting differences in sentence pairs
- Hybrid method combining the above two methods

II. RELATED WORK

Text simplification research is mainly written in English. There is also some research in Japanese that we target. Simple PPDB [7] is a dictionary for simplification, constructed by collecting simplified paraphrases from a large-scale paraphrase dictionary (PPDB) [8] in English, and a Japanese version(Simple PPDB: Japanese [9]) is available as well. Paraphrasing to simplify vocabulary, as in those dictionaries, is called lexical simplification. In sentence-level simplification, a machine learning method using a parallel corpus is applied in which pairs of difficult sentences and plain sentences are constructed. Statistical machine translation (SMT) [10] and neural machine translation (NMT) [11] are used to train parallel corpora. In English sentence simplification, parallel corpora are often constructed by sentence pairs extracted using English Wikipedia³ and Simple English Wikipedia⁴ [12]. In Japanese research, construction of a parallel corpus by hand, construction using crowdsourcing, and simplification by machine translation method using a parallel corpus have been studied [13]–[15].

English used in Simple English Wikipedia is constrained by simple grammar and vocabulary. Meanwhile, Japanese simplified corpora do not consider simple grammar. Therefore, they cannot be used as a parallel corpus for grammatical simplification.

We extracted paraphrases for simplification from parallel corpus. The extraction of paraphrases from parallel corpus by using the alignment method [16] is proposed [17]. We used the method for grammatical simplification by using a manually constructed parallel corpus.

³<https://en.wikipedia.org/>

⁴<https://simple.wikipedia.org>

¹<https://www.e-stat.go.jp/>

²<http://human.cc.hiroaki-u.ac.jp/kokugo/tagengoenglish.html>

Table I
SENTENCES THAT FOLLOW MINIMUM GRAMMAR AND THOSE THAT DO NOT

	Sentence that does NOT follow	Sentence that follows	English translation
1	彼は元気である。	彼は元気です。	He is fine.
2	その動物は滅ぼされた。	その動物は滅びました。	The animal has been destroyed.
3	お好きなだけ取ってください。	好きな分を取ってください。	Take as much as you like.
4	知らないです。	知りません。	I don't know.
5	彼女は必死になって走りました。	彼女は必死に走りました。	She ran desperately.
6	私は次にどうすべきか分かりません。	私は次にどうする方がいいか分かりません。	I do not know what to do next.

Table II
DETAILS OF THE MACHINE TRANSLATION MODEL

Parameter	Value
Architecture	Encoder-Decoder with Attention
Encoder	Bi-directional LSTM
Decoder	LSTM
Number of RNN layers	2
Hidden size	256
Word embedding size	256
Dropout	0.4
Optimizer	Stochastic gradient descent
Learning rate	1.0
Others	Shared embedding layers Copy-mechanism Replace UNK ⁵

III. GRAMMATICALLY SIMPLIFIED CORPUS

A. Easy Japanese grammar checker

One of the examples of simple grammar in Japanese is called “minimum grammar” [18]. It defines the minimum grammar necessary to express one’s own thoughts in Japanese. We constructed a parallel corpus by paraphrasing sentences to satisfy the grammatical constraints. We created a checker that automatically checks whether a sentence follows the grammatical constraints. Input sentences are analyzed using MeCab⁶, the JUMAN dictionary, and CaboCha⁷. MeCab is an engine used for Japanese morphological analysis. The JUMAN dictionary is a dictionary used in morphological analysis by MeCab. CaboCha is a Japanese dependency parser; however, we used it to chunk sentences. The checker checks in chunk units without considering dependencies. Consequently, it has not been possible to implement all of the grammatical constraints. In addition, although the nominal verb is not included in “minimum grammar,” we included it because, otherwise, it would be difficult to paraphrase through crowdsourcing. The paraphrasing of a nominal verb will be an issue for the future. Table I shows examples of sentences that follow minimum grammar and those that do not.

B. Construction of a grammatically simplified corpus

We constructed a grammatically simplified corpus for research into Japanese grammatical simplification. The corpus is constructed by paraphrasing complex sentences to be

simple. We used “small_parallel_enja: 50k En/Ja Parallel Corpus for Testing SMT Methods⁸” as the original text to paraphrase. It is a Japanese-English bilingual corpus constructed by filtering the Tanaka corpus⁹. The workers to paraphrase were recruited using CrowdWorks¹⁰ (Japanese Crowdsourcing service). We assigned 5,000 sentences per person to 13 workers and collected 64,738 paraphrasing pairs. However, 100 of the sentences were given to every worker. Therefore, 100 sentences were paraphrased by all 13 workers. The workers used an easy Japanese grammar checker to paraphrase. Specifically, the sentence is paraphrased by repeating paraphrasing, such that a chunk classified by the checker as not easy is determined to be easy.

IV. METHODS

A. Machine translation approach

In the case of simplification, the machine translation approach is often used, considering before and after simplification as different languages. We trained a neural machine translation model using the parallel corpus explained above and constructed the simplification model. The details of the machine translation model used are shown in Table II.

B. Proposed method

Because text simplification outputs sentences with the same language and meaning as the input, there are many common parts between the input and output. Therefore, there is no need to generate an output sentence from scratch. NMT inputs and outputs entire sentences, but those do not require simplification. Thus, we propose a method to extract and apply the difference in sentence pairs for simplification.

1) *Extraction of paraphrases*: First, a sentence pair is considered before and after simplification. Next, each sentence is separated into chunks with CaboCha. A sentence is expressed by a sequence of chunks, which represent sequences of words. The edit distance from each sentence-pair sequence is calculated. Then, according to the edit distance, the substitution is extracted among the editing

⁸https://github.com/odashi/small_parallel_enja

⁹http://www.edrdg.org/wiki/index.php/Tanaka_Corpus

¹⁰<https://crowdworks.jp/>

¹⁰Replace the generated UNK tokens with the source token that had highest attention weight.

⁶<https://taku910.github.io/mecab/>

⁷<https://taku910.github.io/cabocha/>

Table III
EXTRACTED PARAPHRASES FROM TABLE I

	Source	Target	English Translation
1	元気である。	元気です。	fine.
2	滅ぼされた。	滅びました。	has been destroyed.
3	お好きなだけ	好きな分を	as much as you like
4	知らないです。	知りません。	do not know.
5	必死になつて	必死に	desperately
6	どうすべきか	どうする方がいいか	What shoud I do?

Table IV
EXAMPLE THAT CANNOT BE PARAPHRASED WITHOUT GENERALIZATION

	Surface	English Translation
Chunk1	食べた	ate (plain form)
Chunk2	食べました	ate (polite form)
Chunk1'	飲んだ	drank (plain form)
Chunk2'	飲みました	drank (polite form)

operations (substitution, insertion, and deletion) between the two sequences (implemented using Python difflib¹¹). In this manner, paraphrases between sentences can be extracted. The extracted paraphrases are not necessarily a substitution of one chunk for another chunk. Multiple chunks may be substituted by one chunk or one chunk may be substituted by multiple chunks. The paraphrases extracted from the corpus are shown in Table III.

2) *Generalization of paraphrase by conjugation:* The applicable range of paraphrases extracted using only surface of words is narrow. The example that cannot be paraphrased without generalization is shown in Table IV. Even if it is possible to extract a paraphrase from Chunk1 to Chunk2, it is not possible to paraphrase from Chunk1' to Chunk2'. This means that the applicable range of the extracted paraphrase is narrow. Such examples also appear elsewhere. Therefore, we generalize the verb and adjective words to the form of “part-of-speech + conjugation” without using surface. Thus, even if only one of these paraphrases can be extracted, the other paraphrase is also applicable. The generalization is applied only to the first verb or adjective in a chunk, because the second and subsequent verbs are often non-independent verbs. It is often the same with adjectives. The example of generalization is shown in Table V.

3) *Application to the paraphrases:* The paraphrases are applied using the pair before and after paraphrasing. The applied paraphrase is performed in the following procedure.

- (i) Divide the input sentence into chunk sequences with CaboCha.
- (ii) Search for the part that matches the chunk sequence before paraphrasing from the input. The search is performed in order from the beginning of the input sentence.
- (iii) Substitute the chunk sequence with the paraphrased chunk sequence.

¹¹<https://docs.python.org/3/library/difflib.html>

- (iv) Change to the correct conjugation using the surface of the input word.

If there are multiple matches in step 2, we repeat step 2 and step 3. In addition, the search is sequentially performed from the top of the input chunk sequence and the paraphrase whose source sequence of the chunk is the longest. If there is still more than one paraphrase, the most frequent paraphrase is selected.

4) *Filtering paraphrase by frequency:* Even if there are any applicable paraphrases, they are not always applied. In the proposed method, the frequency of occurrence of the source chunk sequence of the paraphrase in the data set($= n_{all}$) and the frequency of the applied paraphrase($= n_{applied}$) are counted. The ratio($\alpha = n_{applied}/n_{all}$) is determined using those values. Paraphrases are filtered using the ratio α and threshold($= \alpha_{th}$). By applying the paraphrases only when α exceeds the threshold($= \alpha_{th}$), it is possible to adjust how much the paraphrases are applied. In this study, we set α_{th} to 0.4 using validation data.

C. Hybrid method

Our proposed method described above can be hybridized with conventional machine learning methods. In the hybrid method, the proposed method is used as preprocessing for the input of the machine translation model. In other words, simplification is performed in the two steps of the proposed method and the machine translation approach. The hybrid method can simplify sentences that can be simplified only by either of the two methods. In this study, we set α_{th} to 0.6 using validation data.

V. EXPERIMENTS

We performed experiments using the dataset described below to compare the approaches described above. The evaluation was performed by automatic evaluation and manual evaluation described below. In automatic evaluation, a system that outputs without changing the input sentences was set as a baseline.

A. Datasets

The dataset used for the experiments is the grammatically simplified corpus described above. The sentence pairs with the common source sentence paraphrased by the 13 workers are used as test data, and 58,438 sentence pairs are used for

Table V
GENERALIZED PARAPHRASES OF TABLE III

	Before	After
1	元気である。	Adjective(dearu-lemma form)
2	滅ぼされた。	Verb(irrealis form) れた
3	お好きなだけ	お Adjective(adverbial form) だけ
4	知らないです。	Verb(irrealis form) ないです
5	必死になつて	Adjective(adverbial form) なつて
6	どうすべきか	どう Verb(literally lemma form) べきか

Table VI
CRITERIA FOR HUMAN EVALUATION

Fluency	
4	It is a grammatically correct sentence.
3	It has some grammatical mistakes, but the meaning of the sentence can be understood.
2	The grammar is incorrect, but you can guess the meaning.
1	It has many grammatical mistakes, and the meaning cannot be understood.
Meaning preservation	
4	The meanings of the two sentences are the same.
3	The specific meanings of the two sentences are different, but the overall meaning is the same.
2	The specific meanings of the two sentences are different, but the meanings of parts are the same.
1	The meanings of the two sentences are quite different.

the extraction of paraphrases while 5,000 sentence pairs are used to determine hyperparameters.

B. Automatic evaluation

We evaluate each method with BLEU, SARI [19], and the simplification rate. Although BLEU is a metric originally used to evaluate machine translation, it is also used to evaluate text simplification [10], [11], [13], [15]. SARI is a metric for simplicity that evaluates n-gram F-measure (addition, keeping) and relevance in the operation of simplification. We used multi-bleu.perl (Moses)¹² to calculate BLEU and SARI.py¹³ to calculate SARI. We define the simplification rate as follows. First, the number of not-simple chunks ($= N_{input}$) included in the input sentence is counted. The same is done to output sentences ($= N_{output}$). Then, $(N_{input} - N_{output})/N_{input}$ is calculated and defined as the simplification rate. The simplification rate has a value between 0 and 1. If the number of not-simple chunks does not change between the input and output, the value is 0. If the output sentence does not contain not-simple chunks, the value is 1.

C. Human evaluation

We manually evaluated the “fluency” and “meaning preservation.” The criteria for manual evaluation are shown in Table VI, which is the same as that used in [15]. Manual evaluation was performed by the first author of this paper. The scores are published to confirm whether the evaluation is reasonable¹⁴.

Table VII
AUTOMATIC EVALUATION

Methods	BLEU	SARI	Simplicity [%]
Baseline (same as input)	50.78	21.98	0
MT approach	72.59	67.65	77.8
Proposed method	67.12	55.98	54.5
Hybrid method	73.63	68.84	82.8

Table VIII
HUMAN EVALUATION

Methods	Fluency	Meaning preservation
MT approach	3.69	3.62
Proposed method	3.87	3.90
Hybrid method	3.75	3.70

VI. RESULTS

Some of the input and output results are shown in the appendix. This section shows the results of automatic evaluation and manual evaluation.

A. Automatic evaluation

The automatic evaluation results of each method are shown in Table VII. As can be seen from the baseline’s high BLEU, input sentences and references are similar. Therefore, BLEU of each method is higher than a general machine translation task. The results show that the proposed method is lower in both BLEU, SARI, and simplification rates than the machine translation approach. However, the hybrid approach attains the highest score.

B. Human evaluation

The human evaluation results of each method are shown in Table VIII. When the input sentence and output one are the same, it is excluded from the result. Because, in

¹²<http://www.statmt.org/moses/>

¹³<https://github.com/cocoxu/simplification>

¹⁴Hidden for anonymization now

that case, the fluency and the meaning preservation always have a score of 4. The machine translation approach has low fluency and meaning preservation, while those of the proposed method are relatively high. The hybrid method has a score between the two methods'. The hybrid method has higher fluency and meaning preservation than that of the machine translation approach, which indicates that the sentences that can be simplified are different in the machine translation approach and the proposed method. Therefore, the hybrid method works effectively.

VII. CONCLUSION

We constructed a Japanese grammatical simplification corpus and compared conventional machine translation approach, proposed method by extracting differences between sentence pairs, and hybrid method of them as simplification methods. We obtained the result that the hybrid method is effective in both automatic evaluation and manual evaluation. This means that sentences that can be simplified are different in the machine translation approach and proposed method. In grammatical simplification, it is important to take advantage of the fact that input and output sentences change only partially. In this study, focus was placed only on grammatical simplification; however, general simplification may have the same tendency.

Meanwhile, the proposed method still requires improvement. For example, when multiple paraphrases can be selected, selecting the most frequent paraphrase may not be the most suitable for the context. Therefore, it is necessary to use more information than frequency to find the most suitable paraphrase. In this experiment, paraphrasing is applied sequentially from the top of the input sentence; however, the optimal order is not clear. If there are more than one paraphrasing orders, more contextual ones need to be selected.

Future study will involve finding a method to select the optimal paraphrase by considering the context when multiple paraphrases can be applied. If it is possible, both the fluency and meaning preservation will be improved.

ACKNOWLEDGMENT

This work was supported in part by JSPS KAKENHI Grants-in-Aid for Challenging Research (Exploratory) Grant ID 17K18481.

REFERENCES

- [1] K. Iwata, “The preference for English in linguistic services: ‘Japanese for living: Countrywide survey’ and Hiroshima (<special issue> changing Japanese society and language issues),” in *The Japanese Journal of Language in Society*, vol. 13, no. 1, 2010, pp. 81–94.
- [2] H. Saggion, “Automatic text simplification. synthesis lectures on human language technologies.” Morgan & Claypool Publishers, 2017.
- [3] R. Chandrasekar, C. Doran, and B. Srinivas, “Motivations and methods for text simplification,” in *COLING*, 1996, pp. 1041–1044.
- [4] S. Štajner and M. Popovic, “Can text simplification help machine translation?” in *EAMT*, 2016, pp. 230–242.
- [5] R. Snow, B. O’Connor, D. Jurafsky, and A. Ng, “Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks,” in *EMNLP*, 2008, pp. 254–263.
- [6] C. Callison-Burch, “Fast, cheap, and creative: Evaluating translation quality using Amazon’s Mechanical Turk,” in *EMNLP*, 2009, pp. 286–295.
- [7] E. Pavlick and C. Callison-Burch, “Simple PPDB: A paraphrase database for simplification,” in *ACL*, 2016, pp. 143–148.
- [8] J. Ganitkevitch, B. Van Durme, and C. Callison-Burch, “PPDB: The paraphrase database,” in *NAACL-HLT*, 2013, pp. 758–764.
- [9] T. Kajiwara and M. Komachi, “Simple PPDB: Japanese,” in *Proc. of the 23rd Natural Language Processing of Japan*, 2017, pp. 529–532.
- [10] S. Štajner, H. Bechara, and H. Saggion, “A deeper exploration of the standard PB-SMT approach to text simplification and its evaluation,” in *ACL-IJCNLP*, 2015, pp. 823–828.
- [11] S. Nisioi, S. Štajner, S. P. Ponzetto, and L. P. Dinu, “Exploring neural text simplification models,” in *ACL*, 2017, pp. 85–91.
- [12] W. Hwang, H. Hajishirzi, M. Ostendorf, and W. Wu, “Aligning sentences from standard Wikipedia to simple Wikipedia,” in *NAACL-HLT*, 2015, pp. 211–217.
- [13] T. Maruyama and K. Yamamoto, “Simplified corpus with core vocabulary,” in *LREC*, 2018, pp. 1153–1160.
- [14] A. Katsuta and K. Yamamoto, “Crowdsourced corpus of sentence simplification with core vocabulary,” in *LREC*, 2018, pp. 461–466.
- [15] T. Maruyama and K. Yamamoto, “Sentence simplification with core vocabulary,” in *IALP*, 2017, pp. 363–366.
- [16] W. A. Gale and K. W. Church, “A program for aligning sentences in bilingual corpora,” in *ACL*, 1991, pp. 177–184.
- [17] R. Barzilay and K. R. McKeown, “Extracting paraphrases from a parallel corpus,” in *ACL*, 2001, pp. 50–57.
- [18] I. Iori, “Issues on the study of “Yasashii-nihongo”: Today and tomorrow,” in *The Hitotsubashi journal for Japanese language education*, 2014, pp. 1–12.
- [19] W. Xu, C. Napoles, E. Pavlick, Q. Chen, and C. Callison-Burch, “Optimizing statistical machine translation for text simplification,” *TACL*, vol. 4, pp. 401–415, 2016.

APPENDIX: SOME SIMPLIFIED OUTPUTS AND THE REFERENCE

	Japanese	English
Input	私は 午後 ずっと その 本 を 読み 続けて いる。	I have been reading the book all afternoon.
Our proposed method	私は 午後 ずっと その 本 を 読み 続けて いる。	I have been reading the book all afternoon.
MT approach	私は 午後 ずっと その 本 を 読む 続けて います。	I have been read the book all afternoon.
Hybrid method	私は 午後 ずっと その 本 を 読んで います。	I have been reading the book all afternoon.
Reference	私は 午後 ずっと その 本 の 読書 を 続けて います。	I have been reading the book all afternoon.
Input	すぐ に 出て 行け !	Get out soon!
Our proposed method	すぐ に 出て 行け !	Get out soon!
MT approach	すぐ に 出て ください !	Please get out soon!
Hybrid method	すぐ に 出て ください !	Please get out soon!
Reference	すぐ に 出て いき なさい !	Get out soon!
Input	君 に 会い たかっ た よ。	I wanted to see you.
Our proposed method	君 に 会い たかっ た よ。	I wanted to see you.
MT approach	君 に 会い たい と 思い まし た よ。	I wanted to see you.
Hybrid method	君 に 会い たい です よ。	I wanted to see you.
Reference	君 に 会い たい と 思っ て い まし た。	I wanted to see you.
Input	すべて は 結局 同じ よ。	Everything is the same after all.
Our proposed method	すべて は 結局 同じ よ。	Everything is the same after all.
MT approach	すべて は 結局 です よ。	Everything is after all.
Hybrid method	すべて は 結局 続き まし た よ。	Everything went on eventually.
Reference	すべて は 結局 同じ です よ。	Everything is the same after all.
Input	私が 外出 し て いる あいだ、 犬 の 面倒 を み て くれ ない。	Will you take care of my dog while I'm out?
Our proposed method	私が 外出 中 は、 犬 の 面倒 を み て く だ さ い。	Please take care of my dog while I'm out.
MT approach	私が 外出 し た とき、 犬 の 面倒 を み て く だ さ い。	Please take care of my dog when I'm out.
Hybrid method	私が 外出 中 は、 犬 の 面倒 を み て く だ さ い。	Please take care of my dog while I'm out.
Reference	私が 外出 す る 間 に、 犬 の 面倒 を み て く だ さ い。	Please take care of my dog while I'm out.
Input	私は 先生 に 叱 ら れ た。	I was scolded by my teacher.
Our proposed method	私は 先生 に 叱 り ま し た。	I scolded my teacher.
MT approach	先 生 を 私 は 叱 り ま し た。	I scolded my teacher.
Hybrid method	私 は 先 生 を 叱 り ま し た。	I scolded my teacher.
Reference	私 は 先 生 に 叱 ら れ ま し た。	I was scolded by my teacher.
Input	この 価 格 に は、 運 費 は 含ま れ て い ま せ ん。	This price does not include fares.
Our proposed method	この 価 格 は、 運 費 を 含 ん で い ま せ ん。	This price does not include fares.
MT approach	この 価 格 に は、 運 費 は 含ま い ま せ ん。	This price does not <broken> fares.
Hybrid method	この 価 格 は、 運 費 を 含 ん で い ま せ ん。	This price does not include fares.
Reference	この 価 格 は、 運 費 を 含み ま せ ん。	This price does not include fares.
Input	小 錢 を お 持ち で す か。	Do you have change?
Our proposed method	小 錢 を 持 つ て い ま す か。	Do you have change?
MT approach	小 錢 を 教え て く だ さ い。	Please tell me your change.
Hybrid method	小 錢 を 持 つ て い ま す か。	Do you have change?
Reference	小 錢 を 持 つ て い ま す か。	Do you have change?

Effective Data Augmentation Approaches to End-to-End Task-Oriented Dialogue

Jun Quan and Deyi Xiong*

School of Computer Science and Technology

Soochow University

Suzhou, China

terryqj0107@gmail.com; dyxiong@suda.edu.cn

Abstract—The training of task-oriented dialogue systems is often confronted with the lack of annotated data. In contrast to previous work which augments training data through expensive crowd-sourcing efforts, we propose four different automatic approaches to data augmentation at both the word and sentence level for end-to-end task-oriented dialogue and conduct an empirical study on their impact. Experimental results on the CamRest676 and KVRET datasets demonstrate that each of the four data augmentation approaches is able to obtain a significant improvement over a strong baseline in terms of Success F₁ score and that the ensemble of the four approaches achieves the state-of-the-art results in the two datasets. In-depth analyses further confirm that our methods adequately increase the diversity of user utterances, which enables the end-to-end model to learn features robustly.

Keywords-effective; data augmentation; end-to-end; task-oriented dialogue; state-of-the-art; robust

I. INTRODUCTION

Task-oriented dialogue systems have evolved from traditional modularized pipeline architectures [1], [2], [3] to recent end-to-end trainable frameworks [4], [5], [6]. However, the major data challenge for both formalisms remains the same: the lack of annotated dialogue dataset in specific tasks or domains. Various slots and values in dialogue utterances need to be manually labeled for use in supervised learning. As the process of manual annotation is time-consuming and expensive, publicly available task-oriented dialogue datasets normally contain only a few thousand dialogues. For data-driven dialogue systems, especially neural dialogue systems which are more data-hungry, insufficient training data will substantially limit their power to learn from data, resulting in poor robustness and performance.

In this paper, we are interested in handling such a data scarce problem via automatic and cheap data augmentation methods. We propose four different data augmentation approaches: synonym substitution and stop-word deletion at the word level, translation and paraphrasing at the sentence level. We only apply these approaches to rephrase user utterances while keeping machine utterances intact on the training data. For user utterances, we leave slots and corresponding values unchanged and reword the remaining parts, keeping the meanings of user utterances as much the same as possible. In doing so, we hope to diversify user utterances so that our dialogue system can learn to deal with language variabilities in a robust way.

We use TSCP, an end-to-end dialogue system, recently proposed by [6] to validate the efficacy of our methods. We conduct experiments on two public datasets, CamRest676 and KVRET. The combination of the four data augmentation methods can collectively outperform the basic TSCP model by 4.5 points in terms of F₁ score, the TSCP model with reinforcement learning (RL) by 2.5 points on the CamRest676 dataset. Higher improvements are achieved on the KVRET dataset, 7.8 points and 4.1 points in terms of F₁ over the basic TSCP model and TSCP+RL respectively.

The contributions of the paper are threefold:

- First, we present and empirically investigate four different approaches to data augmentation for end-to-end task-oriented dialogue, which, to the best of our knowledge, is the first attempt in automatic data augmentation for task-oriented dialogue.
- Second, we achieve the state-of-the-art performance on the two datasets with the proposed methods.
- Third, our analyses further display that data augmentation on user utterances is better than augmentation on machine utterances. Details on how the proposed methods improve the performance are also provided.

II. BACKGROUND: END-TO-END TASK-ORIENTED DIALOGUE

Task-oriented dialogue systems that can be trained end-to-end have been studied in recent years as alternatives to traditional pipeline-style dialogue systems. Without loss of generality, we use Sequicity [6] as our baseline system to evaluate our data augmentation methods. It significantly outperforms state-of-the-art pipeline-based methods and obtains a satisfactory entity match rate on out-of-vocabulary (OOV) cases where pipeline-designed competitors totally fail. Sequicity handles both task completion and response generation in a single seq2seq model which can be further optimized with reinforcement learning. It provides a theoretically and aesthetically appealing framework, as it achieves true end-to-end trainability with one single seq2seq model. The key concept introduced in Sequicity is the belief span (bspan), a text span that tracks the dialogue belief states at each turn.

Based on this concept, Sequicity decomposes the task-oriented dialogue problem into the generation of bspans and machine responses in a seq2seq framework. Specifically it decodes in two stages. In the first stage, it generates a bspan to facilitate knowledge base (KB) retrieval. It

*Corresponding author

then generates a machine utterance in the second stage, conditioned on the knowledge base search result and the bspan from the previous stage. Our work is based on an implementation of the Sequicity as a two-stage copynet (TSCP). In the implementation, CopyNet [7] is used to instantiate Sequicity to allow key words from previous utterances to recur in bspans and generated machine responses.

III. DATA AUGMENTATION APPROACHES

In this section, we elaborate the four data augmentation approaches at both the word and sentence level.

A. Word-Level Data Augmentation

We substitute words with their synonyms and delete stop words so as to produce diversity in user utterances at the word level.

In synonym substitution, we first utilize the NLTK toolkit [8] and WordNet [9], [10] to conduct part-of-speech tagging and synonym retrieval respectively. In order to ensure that the meaning of user utterances does not change semantically, we only allow some specific words to be replaced by their synonyms. Proper nouns (e.g., *Africa*, *America*), qualifiers (e.g., *the*, *a*, *some*, *most*, *every*, *no*), personal pronouns (e.g., *hers*, *herself*, *him*, *himself*), and modal verbs (e.g., *can*, *cannot*, *could*, *couldn't*) should not be replaced as the substitution of them can easily result in inconsistent statements or even semantic changes. For notional verbs (e.g., *want*, *like*, *tell*, *find*), adjectives (e.g., *cheap*, *great*, *delicious*) and nouns (e.g., *food*, *restaurant*, *area*, *south*), we look up their synonyms from WordNet and select the candidate synonyms whose part-of-speech tags are consistent with the corresponding words. For each user utterance, we randomly sample one word that satisfies our substitution rules and randomly select a synonym candidate to replace it. In this way, multiple user utterances can be randomly generated for each original utterance in the training data. These generated utterances will be added to the training data to increase diversity at the word level.

Similarly, we can obtain varieties by deleting stop words in user utterances without changing their meaning. It is common for users to ignore stop words, such as articles, prepositions, adverbs and conjunctions. In order to improve the robustness of the task-oriented dialogue system, and to enable it to pay more attention to the key semantic information in user utterances, we propose to discard these high-frequency stop words in user utterances.

B. Sentence-Level Data Augmentation

For data augmentation at the sentence level, we investigate two different approaches: translation and paraphrasing. These two methods will improve the sentence-level variances, not limited to the presence/absence or variety of some specific words.

We use neural machine translation (NMT) models to translate user utterances into other languages and then use reversed NMT systems to translate the generated translations from other languages back to the original

language. In this paper, we use Google online translation engine as our NMT translation system.

For the sentence-level paraphrasing, we use a seq2seq paraphrase model which contains a bidirectional LSTM encoder and LSTM decoder together with an attention network.¹ The model is trained on a mixed data set consisting of paraphrases from para-nmt-5m, Quora question pairs, SNLI and Semeval [11], [12]. In the decoder part, we can either use a greedy search to generate a single unique paraphrase for each entire user utterance, or generate a plenty of different paraphrases via sampling from a distribution.

C. Implementation Details for the four Data Augmentation Approaches

Synonym substitution: we created four different utterances for each user utterance by randomly replacing words with their synonyms. The created data was combined with the original training data. The size of the augmented data in this way was 5 times as large as that of the original training data.

Stop-word deletion: for this augmentation, we utilized the dictionary of stop words from NLTK toolkit and created only one copy for each user utterance and combined the additional copy with the original data.

Translation: user utterances in original English version data were translated into Chinese, Japanese, French, German via Google Translate, and then translated back to English, thus forming four sets of data expressed in different styles.

Paraphrasing: we generated four sets of dialogue data with the seq2seq-based paraphrase generator.

Assembled Augmentation: we combined all data generated by the four methods above. Together, the size of the assembly augmented data is 14 times as large as that of the original data.

The sizes of mini-batch and vocabulary for each data augmentation approach on the two datasets are shown in Table I, which are chosen according to the performance on the development set.

IV. EXPERIMENTS AND ANALYSES

We conducted extensive experiments and analyses on two datasets to validate the effectiveness of the proposed methods in this section.

A. Datasets and Settings

We used two datasets: CamRest676 [13], [14], [15] and KVRET [5], both of which are manually created by crowd-sourcing workers on the Amazon Mechanical Turk platform by a Wizard-of-Oz method [16]. CamRest676 contains 676 dialogues in the restaurant searching domain while KVRET covers three domains: calendar scheduling, weather information and point of interest (POI) navigation.

For TSCP, the dimensionality d for both hidden states and word embeddings was set to 50. Vocabulary size V was 800 for CamRest676 and 1400 for KVRET. The

¹<https://github.com/vsuthichai/paraphraser>

Table I
THE SIZES OF MINI-BATCH AND VOCABULARY FOR THE FOUR DATA AUGMENTATION APPROACHES.

	CamRest676		KVRET	
	Batch size	Vocab size	Batch size	Vocab size
Synonym Substitution	64	800	32	1800
Stop-Word Deletion	32	800	32	1400
Translation	100	800	32	1800
Paraphrasing	64	800	64	1800
Assembled Augmentation	64	800	256	1800

Table II
EXPERIMENT RESULTS ON CAMREST676 AND KVRET.

	CamRest676	KVRET
		Success F ₁
Results from [6]		
TSCP	0.834	0.774
TSCP + RL	0.854	0.811
Our implementation		
TSCP	0.832	0.815
TSCP + RL	0.858	0.831
Results obtained by data augmentation		
Translation	0.869	0.842
Paraphrasing	0.869	0.841
Synonym Substitution	0.871	0.833
Stop-Word Deletion	0.856	0.831
Assembled Augmentation	0.879	0.852
Machine Utterance Augmentation (synonym substitution)	0.775	-
User + Machine Utterance Augmentation (translation)	0.822	-

size of mini-batch for both datasets was set 32. The model was trained with the Adam optimizer [17], with a learning rate of 0.003 and a decay parameter of 0.5. We used a learning rate of 0.0001 and decay of 0.8 for the subsequent reinforcement learning process. We used beam search strategy with a beam size of 10 on CamRest676 and greedy search strategy on KVRET. Early stopping was also performed to improve the training efficiency.

We used the Success F₁ score as the automatic metric for dialogue evaluation. The Success F₁ score estimates both the precision and recall of requested slots being successfully answered in dialogues, a metric proposed by [6].

B. Results

Table II shows the experiment results on the two datasets, from which we have three findings. First, the results demonstrate that all the proposed data augmentation methods contribute to the significant improvements in F₁ over the basic TSCP model. Except for the stop-word deletion method, all other methods perform better than the RL-enhanced TSCP. Second, the sentence-level augmentation methods are better than the word-level methods in most cases as the former provide more variances for user utterances. Third, the assembled augmentation, which combines all data generated by the four data augmentation methods, achieve the new state-of-the-art performance on the two datasets, with more than 2 points higher than the RL-enhanced TSCP model in terms of F₁ score.

C. Effect of Augmentation on Machine Utterances

At each turn in a dialogue from the two datasets, a user utterance triggers some special requests and a machine

response utterance provides answers to these requests. In our previous experiments, we performed data augmentation only on user utterances. In order to study the effect of data augmentation on machine utterances, we further carried out two experiments. One is to generate both user and machine utterances with the translation augmentation method. The other is to create copies only for machine utterances with synonym substitution. Both experiments were carried out on the CamRest676 dataset.

Results are displayed at the bottom of Table II. It is clear to observe that machine utterance augmentation seriously deteriorates the performance. The reason for this may be that data augmentation introduces both variance and noise. The variance and noise in user utterances can prevent the system from over-sensitivity [18], thus making the system more robust. However, the variance and noise in machine utterances will distract the system. This resonates with the back translation that uses real target sentences and translated source sentences, widely used for seq2seq-based neural machine translation [19].

D. Analysis

We took a deep look into the data to investigate how the proposed data augmentation methods improve the Success F₁ score that computes both the precision and recall of requested slots being correctly answered.

The precision and recall in F₁ can be formulated as follows:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (1)$$

Table III
ANALYSIS ON THE PRECISION AND RECALL.

	CamRest676					
	Success F ₁	Precision	Recall	TP	FP	FN
baseline	0.832	0.885	0.786	422	55	115
Assembled Augmentation	0.879	0.883	0.875	467	62	67

Table IV
DIALOGUE EXAMPLES OF THE BASELINE AND DATA AUGMENTATION.

Baseline	Data augmentation
<p>User: Can you tell me if there is a moderately priced restaurant serving spanish food anywhere in town?</p> <p>Machine: la tasca serves spanish food.</p> <p>User: What is the address, phone number, and area?</p> <p>Machine: la tasca's phone number is 01223 464630.</p> <p>User: That is it. Thank you.</p> <p>Machine: I'm sorry , there are no restaurants serving spanish food.</p> <p>User: Thanks again! bye.</p> <p>Machine: Thank you for using the cambridge restaurant system.</p>	<p>User: Can you tell me if there is a moderately priced restaurant serving spanish food anywhere in town?</p> <p>Machine: la tasca serves spanish food in the moderate price range. Would you like their location ?</p> <p>User: What is the address, phone number, and area?</p> <p>Machine: Their address is 14 -16 bridge street and their phone number is 01223 464630.</p> <p>User: That is it. Thank you.</p> <p>Machine: Have a nice day.</p> <p>User: Thanks again! bye.</p> <p>Machine: Have a nice day.</p>
<p>User: I would like a restaurant in the north part of town that serves australasian food</p> <p>Machine: There are no restaurants in the north part of town. Would you like a different type of food ?</p> <p>User: How about asian oriental food?</p> <p>Machine: dojo noodle bar serves asian oriental food.</p> <p>User: Yes, I would like their address and phone number.</p> <p>Machine: dojo noodle bar is located at 40210 millers yard city centre and their phone number is 01223 363471.</p> <p>User: Thank you goodbye.</p> <p>Machine: I'm sorry , but dojo noodle bar is a nice day.</p>	<p>User: I would like a restaurant in the north part of town that serves australasian food.</p> <p>Machine: I'm sorry , there are no restaurants serving australasian food. Would you like me to find a different restaurant?</p> <p>User: How about asian oriental food?</p> <p>Machine: saigon city serves asian oriental food</p> <p>User: Yes, I would like their address and phone number.</p> <p>Machine: Their address is 169 high street chesterton chesterton. Their phone number is 01223 356555.</p> <p>User: Thank you goodbye.</p> <p>Machine: Have a nice day !</p>

$$precision = \frac{TP}{TP + FP} \quad (2)$$

$$recall = \frac{TP}{TP + FN} \quad (3)$$

where TP denotes the number of requested slots that are correctly predicted and do exist in real machine responses, FN the number of slots that exist in real responses but not answered at all, FP the number of slots being predicted but not present in real responses.

We provide the values for the precision, recall, TP, FN and FP in Table III for the assembled augmentation. Obviously, our method can significantly improve the recall by nearly 9 points while keeping the precision basically the same as the baseline. The reason behind the improvement of the recall is that the proposed methods substantially increases TP and decreases FN. This is because the diversity in user utterances created by data augmentation helps the dialogue system recognize more requested slots and further allows the decoder to answer these slots in machine responses. Without data augmentation, some slots are just not detected at all in the baseline (thus a higher FN).

E. Dialogue Samples

Table IV shows some dialogue examples generated by the model with or without data augmentation. The dialogues on the left side of the table is generated by the baseline model, while on the right side is the examples generated by the model with assembled data augmentation.

Obviously, the model after our data augmentation is more robust to understand the user utterances and can produce more appropriate machine responses.

V. RELATED WORK

Data augmentation has achieved great success in various tasks including computer vision [20], speech recognition [21] and text classification [22], but is explored in a very limited way for the natural language understanding (NLU) module of traditional pipeline systems of task-oriented dialogue. [23] propose to augment data for the NLU module by adding noise to one single user utterance without considering its relation with other utterances. [24] introduce a technique to expand the limited in-domain data for a new spoken language understanding task. [25] propose a data-augmentation framework to model relations between utterances of the same semantic frame in the training data. Other researchers present methods for gathering dialogue data through crowd-sourcing, e.g., via talking to myself [26] or MultiWOZ [27]. Different from our methods, these methods either focus solely on the NLU module or rely on expensive human efforts.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have presented four different effective methods of data augmentation for end-to-end task-oriented dialogue systems at both the word and sentence level. Empirical study on two public datasets CamRest676 and KVRET shows that data augmentation can prevent the dialogue system from the omission of key information in

user utterances and significantly improve the F₁ score via effectively solving the problem of data scarcity.

In the future, we intend to apply our data augmentation methods on more datasets and to explore some other efficient ways to increase the diversity of machine responses as well.

ACKNOWLEDGMENT

The present research was supported by the National Natural Science Foundation of China (Grant No. 61622209 and 61861130364). We would like to thank the three anonymous reviewers for their insightful comments.

REFERENCES

- [1] A. I. Rudnicky, E. Thayer, P. Constantinides, C. Tchou, R. Shern, K. Lenzo, W. Xu, and A. Oh, “Creating natural dialogs in the carnegie mellon communicator system,” in *Sixth European Conference on Speech Communication and Technology*, 1999.
- [2] V. Zue, S. Seneff, J. R. Glass, J. Polifroni, C. Pao, T. J. Hazen, and L. Hetherington, “Juplter: a telephone-based conversational interface for weather information,” *IEEE Transactions on speech and audio processing*, vol. 8, no. 1, pp. 85–96, 2000.
- [3] V. W. Zue and J. R. Glass, “Conversational interfaces: Advances and challenges,” *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1166–1180, 2000.
- [4] M. Eric and C. D. Manning, “A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue,” *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 468–473, 2017.
- [5] M. Eric, L. Krishnan, F. Charette, and C. D. Manning, “Key-value retrieval networks for task-oriented dialogue,” in *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, 2017, pp. 37–49.
- [6] W. Lei, X. Jin, M.-Y. Kan, Z. Ren, X. He, and D. Yin, “Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2018, pp. 1437–1447.
- [7] J. Gu, Z. Lu, H. Li, and V. O. Li, “Incorporating copying mechanism in sequence-to-sequence learning,” *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1631–1640, 2016.
- [8] S. Bird and E. Loper, “Nltk: the natural language toolkit,” in *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*. Association for Computational Linguistics, 2004, p. 31.
- [9] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, “Introduction to wordnet: An on-line lexical database,” *International journal of lexicography*, vol. 3, no. 4, pp. 235–244, 1990.
- [10] L. F. D’Haro, S. Kim, and R. E. Banchs, “A robust spoken q&a system with scarce in-domain resources,” in *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE, 2015, pp. 47–53.
- [11] J. M. John Wieting and K. Gimpel, “Learning paraphrastic sentence embeddings from back-translated bitext,” in *Proceedings of Empirical Methods in Natural Language Processing*, 2017.
- [12] J. Wieting and K. Gimpel, “Paramnt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations,” *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, pp. 451–462, 2018.
- [13] T.-H. Wen, Y. Miao, P. Blunsom, and S. Young, “Latent intention dialogue models,” in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR.org, 2017, pp. 3732–3741.
- [14] T.-H. Wen, M. Gasic, N. Mrksic, L. M. Rojas-Barahona, P.-H. Su, S. Ultes, D. Vandyke, and S. Young, “Conditional generation and snapshot learning in neural dialogue systems,” *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2153–2162, 2016.
- [15] T.-H. Wen, D. Vandyke, N. Mrksic, M. Gasic, L. M. Rojas-Barahona, P.-H. Su, S. Ultes, and S. Young, “A network-based end-to-end trainable task-oriented dialogue system,” *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 438–449, 2016.
- [16] J. F. Kelley, “An iterative design methodology for user-friendly natural language office information applications,” *ACM Transactions on Information Systems (TOIS)*, vol. 2, no. 1, pp. 26–41, 1984.
- [17] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *Proceedings of the 3rd International Conference for Learning Representations, San Diego*, 2015.
- [18] T. Niu and M. Bansal, “Adversarial over-sensitivity and over-stability strategies for dialogue models,” *arXiv preprint arXiv:1809.02079*, 2018.
- [19] R. Sennrich, B. Haddow, and A. Birch, “Improving neural machine translation models with monolingual data,” *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 86–96, 2016.
- [20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [21] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, “Deep speech: Scaling up end-to-end speech recognition,” *arXiv preprint arXiv:1412.5567*, 2014.
- [22] X. Zhang, J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” in *Advances in neural information processing systems*, 2015, pp. 649–657.
- [23] G. Kurata, B. Xiang, and B. Zhou, “Labeled data generation with encoder-decoder lstm for semantic slot filling,” in *INTERSPEECH*, 2016, pp. 725–729.
- [24] S. Jalalvand, A. Ljolje, and S. Bangalore, “Automatic data expansion for customer-care spoken language understanding,” *arXiv preprint arXiv:1810.00670*, 2018.

- [25] Y. Hou, Y. Liu, W. Che, and T. Liu, “Sequence-to-sequence data augmentation for dialogue language understanding,” in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 1234–1245.
- [26] J. Fainberg, B. Krause, M. Dobre, M. Damonte, E. Kambembwe, D. Duma, B. Webber, and F. Fancellu, “Talking to myself: self-dialogues as data for conversational agents,” *arXiv preprint arXiv:1809.06641*, 2018.
- [27] P. Budzianowski, T.-H. Wen, B.-H. Tseng, I. Casanueva, S. Ultes, O. Ramadan, and M. Gašić, “Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling,” *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 5016–5026, 2018.

Extremely Low Resource Text simplification with Pre-trained Transformer Language Model

Takumi Maruyama and Kazuhide Yamamoto

Nagaoka University of Technology

1603-1, Kamitomioka Nagaoka, Niigata 940-2188, JAPAN

{maruyama, yamamoto}@jnlp.org

Abstract—Recent text simplification approaches regard the task as a monolingual text-to-text generation inspired by machine translation. In particular, the transformer-based translation model outperform previous methods. Although machine translation approaches need a large-scale parallel corpus, parallel corpora for text simplification are very small compared to machine translation tasks. Therefore, we attempt a simple approach which fine-tunes the pre-trained language model for text simplification with a small parallel corpus. Specifically, we conduct experiments with the following two models: transformer-based encoder-decoder model and a language model that receives a joint input of original and simplified sentences, called TransformerLM. Thus, we show that TransformerLM, which is a simple text generation model, substantially outperforms a strong baseline. In addition, we show that fine-tuned TransformerLM with only 3,000 supervised examples can achieve performance comparable to a strong baseline trained by all supervised data.

Keywords-text simplification; language modeling; transfer-learning;

I. INTRODUCTION

Automatic text simplification is a task that reduces the complexity of vocabulary and expressions while preserving the meaning of the text. This technique can be used to make many text resources available for a wide range of readers including children, nonnative speakers, and the disabled. As a preprocessing step, simplification can improve the performance of natural language processing tasks including parsing [1], summarization [2], [3], semantic role labelling [4], information extraction [5], and machine translation [6], [7].

Over the years, the number of tourists in Japan have increased. Japan hosts around 28 million visitors per year¹. In addition, there are approximately 2.32 million foreign residents in Japan², and this number is on the rise. According to a survey conducted by the National Institute for Japanese Language and Linguistics, the number of people who can understand Japanese is more than the number of people who can understand English [8]. Hence, a simplified text is one of the important ways for providing information to foreigners, and therefore, a practical text simplification system is desired.

Recent approaches regard the simplification process as monolingual text-to-text generation task like machine translation [9], [10], [11], [12], [13], [14], [15]. Simplification rewritings are trained automatically from exam-

ples of original-simplified sentence pairs. Neural-machine-translation-based approaches have greatly improved simplification performance compared to statistical-machine-translation-based models or lexical simplification models. These require a large-scale parallel corpus. However, parallel corpora for text simplification are very few and small compared to machine translation tasks. In Japanese, there is no simplified corpus corresponds to Simple English Wikipedia³ [16], [17], [18].

We focus on pre-training as a way to address a low-resource issue. Language model pre-training [19], [20] has led to impressive results on various tasks such as text classification, question answering, and sequence labeling [21], [22], [23]. Particularly, Shleifer et al. [22] have achieved strikingly performance in spite of slightly small supervised examples.

In this paper, we attempt a simple approach which fine-tunes the pre-trained language model for text simplification using an only small parallel corpus. Specifically, we experiment with the following two models, (1) the transformer-based encoder-decoder model, (2) the language model that receives a joint input of original and simplified sentences, called TransformerLM.

II. RELATED WORKS

Back-translation has substantially improved performance in the machine translation task. It is a method that constructs a synthetic parallel corpus by translating a monolingual corpus of a target language to a source language[24], [25]. In text simplification, Qiang et al. [26] use synthetic parallel corpus generated by back-translating the Simple English Wikipedia according to the method of Sennrich et al.[24]. By adding this synthetic data to training data, even a simple machine translation model can outperform more complex models such as model using reinforcement learning. However, back-translation cannot be applied to text simplification if there is no monolingual simplified corpus.

On the other hand, Kauchak [27] has combined a language model trained with a small simplified corpus and one trained with a large original corpus. The combined model performs as well as a model trained with a large simplified corpus on perplexity and lexical simplification tasks. Motivated by this result, we attempt to improve text simplification model using a large original corpus

¹https://www.jnto.go.jp/jpn/statistics/visitor_trends

²<https://www.e-stat.go.jp>

³<https://dumps.wikimedia.org/simplewiki/>

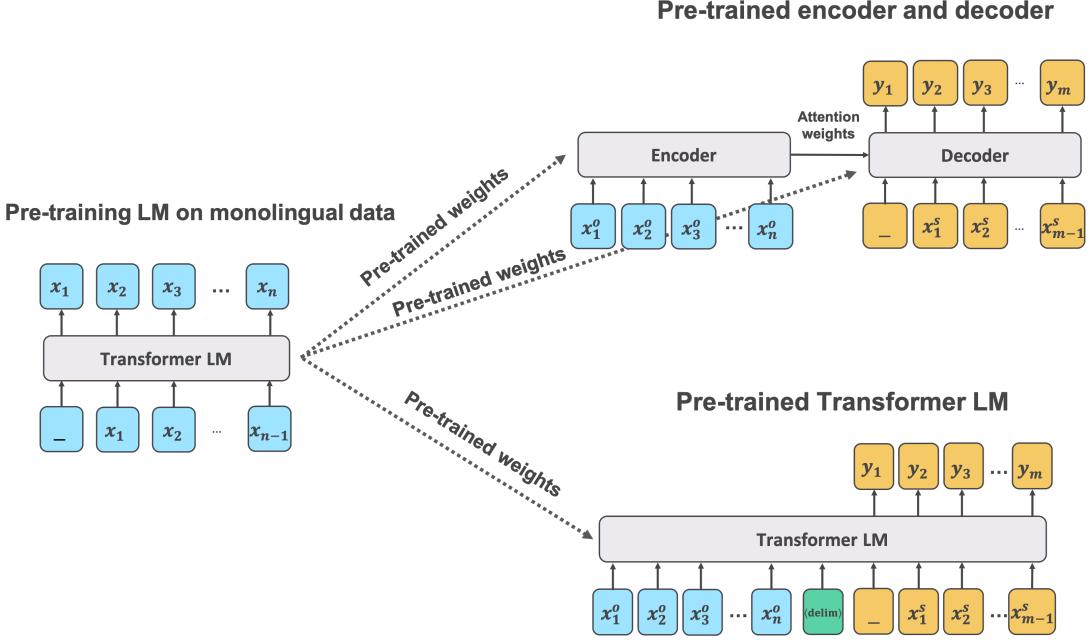


Figure 1. Two fine-tuned models, a transformer-based encoder-decoder model, and a language model that receives a joint input of original and simplified sentences.

instead of a large simplified corpus. Specifically, we train a language model using a large original corpus, and then, fine-tune it with a small parallel corpus for text simplification tasks.

III. METHODS

As shown in Figure 1, we construct a text simplification model by fine-tuning a pre-trained language model. We conducted experiments in two ways: first, using a transformer-based encoder-decoder model; second, using a language model. In this section, we describe the pre-training method of a language model (section III-A). Then, we describe two methods for text generation from an encoder-decoder model (section III-B) and a language model (section III-C).

A. Language Model Pre-training

We use a language model based on transformer [28]. Instead of bidirectional models like ELMo [19] and BERT [20], we use unidirectional models such as GPT [29]. A sentence with N tokens (x_1, x_2, \dots, x_N), our language model trains the parameter θ for maximizing the likelihood $p(x_1, x_2, \dots, x_N; \theta)$.

$$p(x_1, x_2, \dots, x_N; \theta) = \prod_{k=1}^N p(x_k | x_0, x_1, \dots, x_{k-1}; \theta) \quad (1)$$

For pre-training, we use articles extracted from Japanese Wikipedia⁴ by *WikiExtractor*⁵.

⁴<https://dumps.wikimedia.org/jawiki/latest/jawiki-latest-pages-articles.xml.bz>

⁵<https://github.com/attardi/wikiextractor>

B. Text Generation from Pre-trained Encoder-Decoder

We incorporate the weights of pre-training language model into standard encoder-decoder [30] models. The encoder-decoder model consists of a transformer encoder that reads the original sentences, a transformer decoder that generates the simplified sentences, and an attention mechanism [31] that allows the decoder to access encoder states during generation. Both encoder and decoder use the same structure. We compare three ways of incorporating the weights from a pre-trained language model, according to Ramachandran et al. [32]: (1) pre-training the encoder only, (2) pre-training the decoder only, and (3) pre-training both the encoder and decoder. In (3), the parameters of the encoder-decoder attention mechanism initialize randomly.

After the pre-trained weights incorporate into the encoder-decoder model, these are fine-tuned using a parallel corpus. This procedure often leads to catastrophic forgetting where the model's performance on language modeling tasks falls after fine-tuning [33], especially when trained on small supervised datasets. To avoid this problem, we add language modeling loss to translation loss in the fine-tuning step. The translation and language modeling losses are weighted equally.

Instead of a large-scale monolingual corpus, we conduct an experiment pre-training only using parallel corpus similar to Ramachandran et al. [32]. In pre-training using a parallel corpus, the encoder is initialized by a language model pre-trained on the original side, and the decoder is initialized by a language model pre-trained on the simplified side.

Table I
COMPARISON OF TEXT SIMPLIFICATION DATASETS

Datasets	Split Size			N-grams overlap in Simplified sentence [%]				Mean # words	
	Train	Validation	Test	unigrams	bigrams	trigrams	4-grams	Original	Simplified
Literal-translation	32,949	893	1,781	64.48	42.00	31.76	25.28	15.06	17.14
Free-translation	30,259	817	1,637	61.97	38.37	28.05	21.85	15.32	15.84

C. Text Generation from Pre-trained Language Model

We translate an original sentence to a simplified sentence using only a transformer decoder similar to Khandelwal[34] and Hoang[35]. Given the N tokens original sentence $X^o = (x_1^o, x_2^o, \dots, x_N^o)$ and the M token simplified sentences $X^s = (x_1^s, x_2^s, \dots, x_M^s)$, a transformer decoder receives the following input sequence, where, $\langle \text{delim} \rangle$ is a special token that means delimiter between an original sentence and a simplified sentence.

$$X = [X^o, \langle \text{delim} \rangle, X^s] \quad (2)$$

We use the same word embedding layer when the original sentence and the simplified sentence are vectorized. The positional embedding obtained from the following equations adds to word embeddings.

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}}) \quad (3)$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}}) \quad (4)$$

where pos indicates a position, i indicates the dimension, and d_{model} indicates embedding dimension. Note that when the delimitation token $\langle \text{delim} \rangle$ is reached, the position counter is reset. We add language modeling loss to translation loss in the fine-tuning step in the same way as the previous section III-B. The translation and language modeling losses are weighted equally.

IV. EXPERIMENTAL SETUP

A. Datasets

We experiment using two text simplification datasets contained *simplification corpus of local government announcement* as supervised data. When preprocessing, we excluded sentence pairs that are over 100 tokens on an original side or a simplified side. Some statistics of these datasets are shown in Table I.

The corpus is constructed by Moku et al[36]. One thousand one hundred official documents that are distributed in public facilities, such as a city office, hospital, and school, is simplified by 40 Japanese language teachers. This parallel corpus has three simplified versions; *literal-translation*, *free-translation*, *summary*. Each simplified level is defined as follow.

- **literal-translation:** The simplified version that rewrites difficult words or phrases into simple expressions;
- **free-translation:** The simplified version that rewrites a difficult sentence into a simplified sentence while preserving the meaning in the best possible manner;

Table III
COMPARISON OF EACH SYSTEM

Model	Literal-translation BLEU	Literal-translation SARI	Free-translation BLEU	Free-translation SARI
Identical translation	34.65	17.87	29.31	15.86
<i>Non Pre-training</i>				
Encoder-Decoder	19.70	38.35	20.11	40.40
TransformerLM	42.86	51.91	35.96	49.78
<i>Pre-training on parallel corpus</i>				
Pre-train Encoder only	18.44	38.17	17.09	39.25
Pre-train Decoder only	10.86	31.10	8.92	31.19
Encoder-Decoder	14.38	33.92	15.04	36.18
TransformerLM + language modeling loss	34.45	46.36	25.54	42.74
	30.03	43.52	24.67	41.99
<i>Pre-training on Wikipedia</i>				
Pre-train Encoder only	25.21	41.63	24.16	42.86
Pre-train Decoder only	7.44	30.88	10.38	33.70
Encoder-Decoder	13.32	34.41	13.67	36.16
TransformerLM + language modeling loss	44.15	52.46	37.37	50.39
	40.69	50.37	34.22	48.55

- **summary:** The simplified version that contains document-level rewritings such as sentence extraction in addition to sentence-level rewritings.

These consist of grammar and vocabulary defined in the Japanese-Language Proficiency Test Level 2 (N2). Each simplified sentence is manually aligned. In this paper, we attempt to translate an original sentence into a *literal-translation* sentence or a *free-translation* sentence, which is word-level or sentence level simplification. The *summary*, which is a document-level simplification, will be addressed in the future.

B. Model Specifications and Training Details

We use a unidirectional transformer language model with six layers and 16 masked self-attention heads. We set the dimension of a word embedding layer to 512, and the dimension of feedforward networks to 2048. In the encoder-decoder model, both the encoder and decoder use the same parameters. We use the Stochastic Gradient Descent (SGD) for optimizing all models. We set the initial learning rate to 0.25, and 0.1 multiplies it when a validation loss has stopped improving during 10 epochs. The training ends if the learning rate becomes less than 1.0×10^{-5} .

C. Evaluation

We evaluated the model's output based on two metrics, BLEU [37] and SARI [10]. BLEU is a traditional evaluation metric for machine translation tasks. It has a positive correlation with fluency and meaning preservation in text simplification task that does not include sentence splitting [38]. SARI is a recently proposed simplification metric that compares the System output Against References and

Table II
EXAMPLES OF OUTPUT

	Examples	English translation of the left column
Input	健康 診査 票 が ない と 健診 を 受ける こ と が でき ま せん (今 回 ご 案内 さ せ て い た だ いた 郵便 物 に 同 封 さ れ て い ま す)。	If you do not have a medical checkup form, you will not be able to receive a medical checkup. (It is enclosed in this mail).
Encoder-Decoder	健 康 診 断 の 結 果 が でき ま せん 。 で す 。	You can not get the result of your health check.
TransformerLM	健 康 診 断 の 紙 が ない と 健 康 診 断 を 受ける こ と が で き ま せん (今 回 案 内 し た 手 紙 に 入 っ て い ま す)。	If you do not have a form for medical checkup, you will not be able to receive a medical checkup. (It is in this mail).
Reference	健 康 診 断 票 が な か つ た ら 健 康 診 断 を 受ける こ と が でき ま せん (今 回 案 内 し た 手 紸 に 入 っ て い ま す)。	If you do not have a medical checkup form, you will not be able to receive a medical checkup. (It is in this mail).
Input	警 报 ・ 避 難 の 指 示 等 の 内 容 の 伝 送 訓 練 及 び 被 災 情 報 ・ 安 否 情 報 に 係 る 情 報 収 集 訓 練	Training to transmit information about warning and evacuation instructions and training to gather information regard to disaster and safety.
Encoder-Decoder	逃 げ る 住 民 を 案 内 の 情 報 を 集 め て 、 整 理 し ま す 。	Gather and organize guides for the people who will run away.
TransformerLM	警 报 ・ 逃 げ る 指 示 等 の 内 容 の 連 絡 練 習 と 灾 害 に つ い て の 情 報 を 集 め て の 練 習	Training to transmit information about warning and instructions to escape and training to gather information about disasters.
Reference	警 报 や 逃 げ る 指 示 等 の 内 容 を 伝 え る 練 習 と 灾 害 に あ っ た 情 報 ・ 無 事 か ど う か の 情 報 に つ い て の 情 報 を 集 め る 練 習	Training to transmit information about warning and instructions to escape, and training to gather information about disaster and safety.

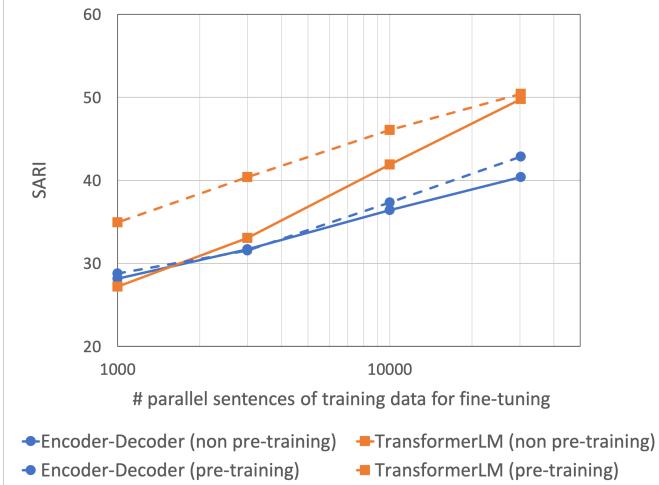


Figure 2. SARI in various data size. Round points (blue line) and square points (orange line) denote Encoder-Decoder and TransformerLM, respectively. The dotted line denotes a model pre-training by Wikipedia, and the solid line denotes a model without pre-training.

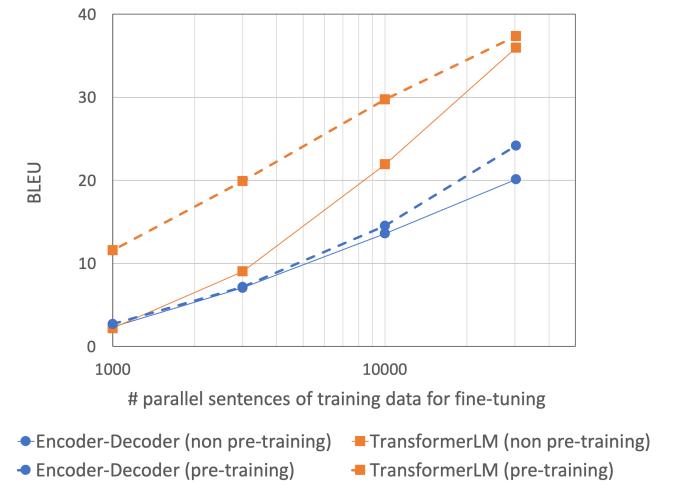


Figure 3. BLEU in various data size. Round points (blue line) and square points (orange line) denote Encoder-Decoder and TransformerLM, respectively. The dotted line denotes a model pre-training by Wikipedia, and the solid line denotes a model without pre-training.

against the Input sentence. This is an arithmetic average of n-gram precision and the recall of three rewrite operations: addition, retention, and deletion. It rewards addition operations, where system output was not in the input but occurred in the references. In addition, it rewards words retained/deleted in both the system output and the references. SARI has a positive correlation with simplicity [38], [39].

V. RESULTS

Comparison of each system is shown in Table III. *Identical translation* denotes a system that outputs an input sentence. Furthermore, *Encoder-Decoder*, *Pre-train Encoder only*, and *Pre-train Decoder* are the models described in section III-B. *TransformerLM* is a model described in section III-C. *+language modeling loss* denotes a model in which language modeling loss adds to translation loss. As shown in Table II, TransformerLM can copy source words more correctly than the encoder-decoder model.

Moreover, it outputs sentences close to reference sentence lengths, unlike the encoder-decoder model outputs. As a result, TransformerLM significantly outperforms Encoder-Decoder in BLEU and SARI.

The results of SARI and BLEU in various supervised data sizes are shown in Figure 2 and Figure 3. We use the encoder-decoder model for which only the encoder is pre-trained and TransformerLM without language modeling loss. As a result, pre-training with large-scale monolingual corpus is more effective on the TransformerLM than on the transformer-based encoder-decoder model. Especially, it is a surprising result that TransformerLM fine-tuned with only 3,000 examples has performance comparable to the encoder-decoder model trained with all the supervised data.

VI. CONCLUSION

We attempt a simple approach which fine-tunes the pre-trained language model for text simplification with a small

parallel corpus. We experiment with the following two models: transformer-based encoder-decoder model and a language model that receives a joint input of original and simplified sentences, called TransformerLM. As a result, pre-training with large-scale monolingual corpus is more effective on the TransformerLM than on the transformer-based encoder-decoder model. We show that the simple TransformerLM outperforms the encoder-decoder model. Furthermore, TransformerLM fine-tuned with only 3,000 supervised examples can achieve performance comparable to a transformer-based encoder-decoder model trained all supervised data.

ACKNOWLEDGMENT

This work was supported in part by JSPS KAKENHI Grants-in-Aid for Challenging Research (Exploratory) Grant ID 17K18481.

REFERENCES

- [1] R. Chandrasekar, C. Doran, and B. Srinivas, “Motivations and Methods of Text Simplification,” *Proceedings of the 16th conference on Computational linguistics-Volume 2*, vol. 0, no. 9, pp. 1041–1044, 1996. [Online]. Available: <https://www.aclweb.org/anthology/C96-2183>
- [2] A. Siddharthan, A. Nenkova, and K. McKeown, “Syntactic simplification for improving content selection in multi-document summarization,” *Proceedings of the 20th International Conference on Computational Linguistics (COLING2004)*, pp. 896–902, 2004. [Online]. Available: <https://www.aclweb.org/anthology/C04-1129>
- [3] W. Xu and R. Grishman, “A parse-and-trim approach with information significance for Chinese sentence compression,” *Proceedings of the 2009 Workshop on Language Generation and Summarisation, ACL-IJCNLP 2009*, no. August, p. 48, 2010. [Online]. Available: <https://www.aclweb.org/anthology/W09-2809>
- [4] D. Vickrey and D. Koller, “Sentence Simplification for Semantic Role Labeling,” *Proceedings of ACL-08: HLT*, no. June, pp. 344–352, 2008. [Online]. Available: <http://www.aclweb.org/anthology/P/P08/P08-1040>
- [5] M. Miwa, R. Sætre, Y. Miyao, and J. Tsujii, “Entity-focused sentence simplification for relation extraction,” *Coling 2010 - 23rd International Conference on Computational Linguistics, Proceedings of the Conference*, vol. 2, no. August, pp. 788–796, 2010.
- [6] H.-b. Chen, H.-H. Huang, H.-H. Chen, and C.-T. Tan, “A Simplification-Translation-Restoration Framework for Cross-Domain SMT Applications,” *Coling-2012*, vol. 2, no. December, pp. 545–560, 2012. [Online]. Available: <https://www.aclweb.org/anthology/C12-1034>
- [7] S. Štajner and M. Popovic, “Can Text Simplification Help Machine Translation?” *In Proceedings of the 19th Annual Conference of the European Association for Machine Translation*, vol. 4, no. 2, pp. 230–242, 2016. [Online]. Available: <https://www.aclweb.org/anthology/W16-3411>
- [8] K. Iwata, “The Preference for English in Linguistic Services: ‘Japanese for Living: Countrywide Survey’ and Hiroshima,” *The Japanese Journal of Language in Society*, vol. 13, pp. 81–94, 2010.
- [9] S. Wubben, A. van den Bosch, and E. Krahmer, “Sentence Simplification by Monolingual Machine Translation,” *The 50th Annual Meeting of the Association for Computational Linguistics*, vol. 1, no. January 2014, pp. 1015–1024, 2012. [Online]. Available: <https://www.aclweb.org/anthology/P12-1107>
- [10] W. Xu, C. Napoles, E. Pavlick, Q. Chen, and C. Callison-Burch, “Optimizing Statistical Machine Translation for Text Simplification,” *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 401–415, 2018. [Online]. Available: <https://www.aclweb.org/anthology/Q16-1029>
- [11] S. Nisioi, S. Štajner, S. P. Ponzetto, and L. P. Dinu, “Exploring Neural Text Simplification Models,” *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pp. 85–91, 2017. [Online]. Available: <https://www.aclweb.org/anthology/P17-2014>
- [12] X. Zhang and M. Lapata, “Sentence Simplification with Deep Reinforcement Learning,” *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 584–594, 2017. [Online]. Available: <https://www.aclweb.org/anthology/D17-1062>
- [13] S. Zhao, R. Meng, D. He, S. Andi, and P. Bambang, “Integrating Transformer and Paraphrase Rules for Sentence Simplification,” *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3164–3173, 2018. [Online]. Available: <https://aclweb.org/anthology/D18-1355>
- [14] H. Guo, R. Pasunuru, and M. Bansal, “Dynamic Multi-Level Multi-Task Learning for Sentence Simplification,” *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 462–476, 2018. [Online]. Available: <https://www.aclweb.org/anthology/C18-1039>
- [15] R. Kriz, J. Sedoc, M. Apidianaki, C. Zheng, G. Kumar, E. Miltsakaki, and C. Callison-Burch, “Complexity-Weighted Loss and Diverse Reranking for Sentence Simplification,” *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019. [Online]. Available: <https://www.aclweb.org/anthology/N19-1317>
- [16] Z. Zhu, D. Bernhard, and I. Gurevych, “A Monolingual Tree-based Translation Model for Sentence Simplification,” *Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 1353–1361, 2010. [Online]. Available: <https://www.aclweb.org/anthology/C10-1152>
- [17] K. Woodsend and M. Lapata, “Learning to Simplify Sentences with Quasi-Synchronous Grammar and Integer Programming,” *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 409–420, 2011. [Online]. Available: <https://www.aclweb.org/anthology/D11-1038>
- [18] W. Coster and D. Kauchak, “Simple English Wikipedia: A New Text Simplification Task,” *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers*, pp. 665–669, 2011. [Online]. Available: <https://www.aclweb.org/anthology/P11-2117>

- [19] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2227–2237, 2018. [Online]. Available: <https://aclweb.org/anthology/N18-1202>
- [20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, 2018. [Online]. Available: <https://www.aclweb.org/anthology/N19-1423>
- [21] J. Howard and S. Ruder, “Universal Language Model Fine-tuning for Text Classification,” *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 328–339, 2018. [Online]. Available: <https://www.aclweb.org/anthology/P18-1031>
- [22] S. Shleifer, “Low Resource Text Classification with ULMFit and Backtranslation,” *CoRR*, pp. 1–9, 2019. [Online]. Available: <http://arxiv.org/abs/1903.09244>
- [23] A. Chronopoulou, C. Baziotsis, and A. Potamianos, “An Embarrassingly Simple Approach for Transfer Learning from Pretrained Language Models,” *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2089–2095, 2019. [Online]. Available: <https://www.aclweb.org/anthology/N19-1213>
- [24] R. Sennrich, B. Haddow, and A. Birch, “Improving Neural Machine Translation Models with Monolingual Data,” *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 86–96, 2016. [Online]. Available: <https://www.aclweb.org/anthology/P16-1009>
- [25] S. Edunov, M. Ott, M. Auli, and D. Grangier, “Understanding Back-Translation at Scale,” *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 489–500, 2018. [Online]. Available: <https://aclweb.org/anthology/D18-1045>
- [26] J. Qiang, “Improving Neural Text Simplification Model with Simplified Corpora,” *CoRR*, 2018. [Online]. Available: <http://arxiv.org/abs/1810.04428>
- [27] D. Kauchak, “Improving Text Simplification Language Modeling Using Unsupervised Text Data,” *Proc. of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1537–1546, 2013. [Online]. Available: <https://www.aclweb.org/anthology/P13-1151>
- [28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention Is All You Need,” *Proceedings of the 31st Conference on Neural Information Processing Systems*, pp. 5998–6008, 2017. [Online]. Available: <https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf>
- [29] A. Radford and T. Salimans, “Improving Language Understanding by Generative Pre-Training (transformer in real world),” *OpenAI*, pp. 1–12, 2018. [Online]. Available: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf
- [30] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to Sequence Learning with Neural Networks,” *Proceedings of twenty-eighth Conference on Neural Information Processing Systems*, pp. 1–9, 2014. [Online]. Available: <https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf>
- [31] D. Bahdanau, K. Cho, and Y. Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate,” *Proceedings of 3rd International Conference on Learning Representations*, pp. 1–15, 2015.
- [32] P. Ramachandran, P. J. Liu, and Q. V. Le, “Unsupervised Pretraining for Sequence to Sequence Learning,” *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 383–391, 2017. [Online]. Available: <https://www.aclweb.org/anthology/D17-1039>
- [33] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio, “An Empirical Investigation of Catastrophic Forgetting in Gradient-Based Neural Networks,” *CoRR*, 2013.
- [34] U. Khandelwal, K. Clark, D. Jurafsky, and L. Kaiser, “Sample Efficient Text Summarization Using a Single Pre-Trained Transformer,” *CoRR*, 2019. [Online]. Available: <http://arxiv.org/abs/1905.08836>
- [35] A. Hoang, A. Bosselut, A. Celikyilmaz, and Y. Choi, “Efficient Adaptation of Pretrained Transformers for Abstractive Summarization,” *CoRR*, 2019. [Online]. Available: <http://arxiv.org/abs/1906.00138>
- [36] M. Moku, K. Yamamoto, and A. Makabi, “Automatic Easy Japanese Translation for information accessibility of foreigners,” *Proceedings of the Workshop on Speech and Language Processing Tools in Education*, pp. 85–90, 2012. [Online]. Available: <https://www.aclweb.org/anthology/W12-5811>
- [37] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: a Method for Automatic Evaluation of Machine Translation,” *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, 2002. [Online]. Available: <https://www.aclweb.org/anthology/P02-1040>
- [38] E. Sulem, O. Abend, and A. Rappoport, “BLEU is Not Suitable for the Evaluation of Text Simplification,” *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 738–744, 2018. [Online]. Available: <https://www.aclweb.org/anthology/D18-1081>
- [39] T. Vu, B. Hu, T. Munkhdalai, and H. Yu, “Sentence Simplification with Memory-Augmented Neural Networks,” *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 79–85, 2018. [Online]. Available: <https://www.aclweb.org/anthology/N18-2013>

Aspect-based Opinion Mining for Code-Mixed Restaurant Reviews in Indonesia

Andi Suciati

Faculty of Computer Science
Universitas Indonesia
Indonesia
andi.suciati@ui.ac.id

Indra Budi

Faculty of Computer Science
Universitas Indonesia
Indonesia
indra@cs.ui.ac.id

Abstract—The goal of opinion mining is to extract the sentiment, emotions, or judgement of reviews and classified it. These reviews are very important because they can affect the decision-making from a person. In this paper, we conducted an aspect-based opinion mining research using customer reviews of restaurants in Indonesia and we focused into analyzing the code-mixed dataset. The evaluation conducted by making four scenarios namely removing stopwords without stemming, without removing stopwords but with stemming, without removing stopwords and stemming, and preprocessing with removing stopwords and stemming. We compared five algorithms which are Random Forest (RF), Multinomial Naïve Bayes (NB), Logistic Regression (LR), Decision Tree (DT), and Extra Tree classifier (ET). The models were evaluated by using 10 folds cross validation, and the results show that all aspects achieved highest scores with different algorithms. LR achieved highest score for *food* (81.76%) and *ambience* (77.29%) aspects while the highest score for *price* (78.71%) and *service* (85.07%) aspects were obtained by DT.

Keywords-opinion mining; restaurant; code-mixed; stemming; stopwords

I. INTRODUCTION

With the rapid development of technology in recent years, the massive amount of data can be found on internet. In past, it is difficult to know what people think about something, for instance a product. However, at this moment, people can express and write their reviews easily. These reviews are very important because they can affect the decision-making from a person. To illustrate, when people are going to buy products, they will check the reviews or the ratings of those products in online websites, such as Amazon¹ and eBay², before buying them. People will choose the product based on the positive sentiments in its reviews. To learn whether the product has many positive, negative, or neutral reviews, the website needs the system that can classified the sentiment of those reviews.

In NLP, there is a task called opinion mining or sentiment analysis. Opinion mining is task for detecting public sentiment towards entities. The goal of opinion mining is to extract the sentiment, emotions, or judgement of reviews and classified it. Usually, a review contains sentiments that can be classified into different polarities. Initial research conducted by Turney [15] in 2002, classified reviews into positive and negative polarities. In SemEval-2015 Task 12, Pontiki et al. [13], divided the

polarities into positive, negative, and neutral, while Ganu et al. [9] divided them into positive, negative, conflict, and neutral. Besides, he also classified the aspect of the restaurant into *food*, *service*, *price*, *ambience*, *anecdotes*, and *miscellaneous*.

There are several works that have been done about sentiment analysis or opinion mining task in various domains. In 2018, Zvarevashe and Olugbara [12] investigated four classification algorithms for hotel domain. Amrania et al. [19] proposed hybrid approach which are Random Forest and Support Vector Machine to identify sentiment of product reviews by Amazon. For movie domain, Yasen and Tedmori [14] used dataset movie reviews from IMBD and compared eight classifiers for classifying the sentiment of the reviews. Next, opinion mining task also important in Indonesia, as a country that has several internet companies that develop in it, according to Mustafa and Budi [1]. There are also several studies about opinion mining that have been conducted in Indonesia. For instances, Alfina et al. [10] conducted an experiment of sentiment analysis in political domain, Putra et al. [18] observed the task in government domain, Jaya et al. [16] analyzed two different domains (president election and online store), and Fiarni et al. [4] proposed rule-based and Naïve Bayes algorithm to classify the sentiment of online transportation service reviews.

In this paper, we conducted an aspect-based opinion mining research using customer reviews of restaurants in Indonesia. The aspect we used are *food*, *price*, *service*, and *ambience* and we divided the polarities into positive, negative and neutral. In Indonesia, there are several websites that can be accessed to see reviews from customers, for instance TripAdvisor³ and Zomato⁴. However, in this research, we used data that obtained from Indonesian culinary review website namely PergiKuliner⁵. In addition, the dataset is using code-mixed language (Indonesian and English).

The rest of this paper is arranged as follows: In section 2, we review the related works with our study. We describe the research steps that applied in this work and the result from that we obtained in section 3. In section 4, we conclude our results.

II. RELATED WORK

Sentiment analysis or opinion mining using code mixing reviews is also gaining attention as a research study. Shalini et. al [11] conducted a study to analyze the

¹ <https://www.amazon.com/>

² <https://www.ebay.com/>

³ <https://www.tripadvisor.com/>

⁴ <https://www.zomato.com>

⁵ <https://pergikuliner.com/>

sentiment of code-mixed Kannada-English, Bengali-English, and Hindi-English. After that, they annotated their data manually into three polarities which are positive, negative, neutral and compared the performances of Doc2Vec with SVM, Fasttext, Bi-LSTM, and CNN. The study shows that CNN gained accuracy 71.50% for Kannada (India)-English dataset while Bi-LSTM achieved 60.20% and 72.20% for Bengali (India)-English and Hindi-English respectively. In 2017, Pravalika et al. [3] did a research about sentiment analysis in movie domain using code-mixed Hindi-English and then implemented two approaches which are lexicon-based and machine learning approaches. The lexicon-based approach gained 86% accuracy while machine learning attained about 72%. However, both of studies were not aspect-based research.

In Indonesia, there are few studies that performed opinion mining task for restaurant domain. Sasmita et al. [7] proposed unsupervised aspect-based sentiment analysis method which was divided by two main tasks (aspect extraction and aspect sentiment orientation classification). The method achieved 88.40 for F1-measure score. Ekawati and Khodra [6] implemented the method from best research in SemEval 2016 for Indonesian restaurant reviews and achieved 0.793, 0.823, and 0.642 F1-measure scores for combination of feature in aspect extraction, aspect categorization, and sentiment classification respectively. To improve the study conducted by Ekawati and Khodra [6], Cahyadi and Khodra [2] performed aspect-based sentiment analysis study for restaurant domain by using Bidirectional Long-Short Term Memory, Conditional Random Field, and Convolutional Neural Network. The scores they obtained are 87.0%, 76.4%, and 78.7% respectively. Those studies for Indonesian reviews were getting promising scores, but all the datasets are in Indonesian. Besides, they divided the polarities into positive and negative only while the sentiment of review can be mildly positive and mildly negative which is classified to neutral according to [13].

III. RESEARCH STEPS

This section will describe the research steps that applied in this study. This work consists 5 important steps as shown in figure 1. The first is collecting the data from the website, the second step is applying few language preprocessing techniques, the third is feature extraction, the fourth is experiment with machine learning models, and the last step was evaluating the models.

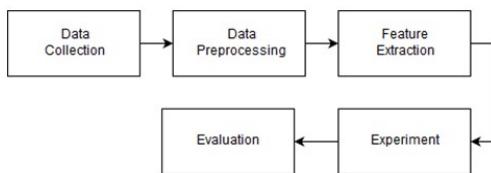


Figure 1. Research steps of opinion mining.

A. Data Collection

In this step, we collected 20000 reviews by scrapping PergiKuliner website. The data contain reviews using Indonesian, English, and mixed (Indonesian & English) languages. The example of the data that retrieved as follow:

- Indonesian review: '*Moen moen terkenal di kota solo dia buka cabang disini, harganya murah dan enak steaknya walaau kadang banyak tepungnya tapi ya pas lah*' (Moen moen that famous in Solo city opened branch in here, price was low and the steak was delicious even though the dough sometimes was thick but it's right)
- English review: '*Great place to drink and do hookah. Especially, all the workers are friendly. They know how to treat customer very well. It is good tho that they open from the afternoon now. Recommended place and worth it to pay!! FYI: Do not forget to order watermelon fruit with sisha is the best recipe in exhale!!*'
- Code-mixed review: '*Best waffle kalo menurut gue hehehe. Terakhir gue pesen waffle gelato yg nuttela banana dan cheese french fries, semuanya enaaaaak! Tempatnya juga asik bgt buat nongki dan untuk harga worth it*' (best waffle in my opinion hehehhe. Last time I ordered waffle gelato Nutella banana and cheese french fries, all were delicious! The place also was cozy for hanging and the price was worth it).

After retrieving our data, we annotated them manually. The aspects that we investigated are *food*, *price*, *service*, and *ambience* following [7]. Besides, we assumed that those aspects always appear in most of reviews. Then we divided the sentiment polarities based on [13] into positive, neutral, and negative. Aspects are classified as positive if the review mentioned positive terms such as '*enak*' (delicious), '*bersih*' (clean), '*murah*' (cheap), and '*sangat bagus*' (excellent). Negative aspects are classified if there were negative terms in review. For example, '*buruk*' (bad), '*mahal*' (expensive), '*kotor*' (dirty), and '*lambat*' (slow). Next, the aspects are classified as neutral aspect if the reviews mentioned term like '*standar*' (standard), '*biasa saja*' (nothing special), '*so so*', and '*not bad but not great*'. In addition, we also classified the aspects that are not mentioned in the review as neutral aspects because we assumed that while people do not mention about the aspect, the polarity will be neither positive nor negative. The following example shows how our data were annotated:

Review: 'i ordered small chicken mushroom with drink and dumplings only 21K. the taste was so so but the *price* was worth.'

{'*food*': '*neutral*', '*price*': '*positive*', '*service*': '*neutral*', '*ambience*': '*neutral*'}

After checking the annotated data, we filtered the reviews and found there are 19201 reviews that can be used in this research.

B. Data Preprocessing

Before entering the feature extraction step, we performed several basic text processing techniques for cleaning our data as follows:

1) *Emoticon Processing*: In our first step for preprocessing, we normalized the emoticons that appeared in the text to string which in Indonesian. For example, we changed ':(' to '*negatif*' (negative) and ':)' to '*positif*' (positive).

2) *Lowercasing*: Second, we lowercased all the words for matching the structures of the words. For instance, ‘GREATTTT!!’ changed to ‘greatttt!!’.

3) *Spelling Correction and Abbreviation (part 1)*: In this step, we corrected the spelling of the words into the formal words and expanding the abbreviation by building our dictionary. This dictionary was made by combining several informal Indonesian and English words. Specifically, ‘tbh i don’t like the taste’ switched to ‘to be honest i do not like the taste’.

4) *Removing URL, Username, Numbers, and Punctuations*: Next, we removed the urls, usernames, and punctuations that occurred in the data. such as, ‘www.instagram.com/Food’, ‘@makanmakan’, and ‘segar banget!!!’ (so fresh!!!) converted to ‘segar banget’ (so fresh).

5) *Spelling Correction and Abbreviation (part 2)*: In this step, we checked the spelling and abbreviation after last step. This was performed to avoid the words that might still not corrected because of the non-alphabet characters still attached to the words. For example, ‘makanannya enk.’, after removing the punctuation ‘makanannya enk’, then changed to ‘makanannya enak’ (the food was delicious) in this step.

6) *Removing Stopwords*: For removing stopwords, we built our own dictionary by combining the stopwords that retrieved from NLTK⁶ and from [8] as our base stopwords dictionary. After that, we removed words such as ‘not’ and ‘tidak’ which are have same meaning. The reason is we wanted to avoid the missing information about the negation of positive words because the words ‘not’ and ‘tidak’ are stopwords. For illustration, ‘the price was not cheap’. If we used the default stopword from library, the sentence will change into ‘price cheap’ instead of ‘price not cheap’. So, by using our own stopwords dictionary, ‘the price was not cheap’ can be changed to ‘price not cheap’.

7) *Removing duplicate character including whitespace*: Next, we checked whether there were strings that contain a character that appear more than once. If we found it, we eliminated the extra character. For example, ‘deliiciousss’ is changed to ‘delicious’. Besides, we also checked whether the review was containing extra space or not.

8) *Stemming*: In this last step of our preprocessing, we applied stemming function from libraries. We used Snowball Stemmer from NLTK library for English and Sastrawi⁷ library for Indonesian. To illustrate, ‘saya memakan kuenya, lumayan’ (*I’m eating its cake, not bad*), changed to ‘saya makan kue, lumayan’ (*I eat cake, not bad*).

C. Feature Extraction

After cleaning our data, we extracted the feature that will be used in the models. The feature we used was bigram term and its vector were extracted by vectorizing

the word representations in the reviews. Besides, we also used the combination of stemming and stopwords steps to see whether they can increase the scores of the models because we used own multi-languages stopwords dictionary (English - Indonesian), and two stemmers for English and Indonesian.

D. Experiment

In our experiments, we used four scenarios and applied five machine learning algorithms. After that, we measured and compared their performances using their F1-scores.

1) *Experiment Scenarios*: We made four scenarios for our experiments in this research. The goal is to see the stopwords and stemming can affect the performances of the models when they applied to our data. In first scenario, we built machine learning models by applying removing stopwords, but we did not use stemming methods. Then in second scenario, we used stemming, but we did not apply the removing stopwords step. After that, we used all preprocessing steps but did not applied the stemming and removing stopwords, and in last scenario, we applied all the preprocessing steps including removing the stopwords and stemming steps.

TABLE I. EXPERIMENT SCENARIOS

Scenarios	Removing Stopwords	Stemming
Scenario 1	✓	✗
Scenario 2	✗	✓
Scenario 3	✗	✗
Scenario 4	✓	✓

2) *Data*: We used all the data (19201 reviews) for all scenarios. From figure 2, we can see that neutral polarity has the highest number in every aspect of the reviews except the *food* aspect. Positive reviews for *food* aspect appeared almost in all reviews and there were only 2965 reviews that were not positive. In contrast, all other aspects had more than 11000 reviews that were neutral. Moreover, *service* aspect has the least reviews with positive polarity and *price* aspect has the highest number of negative reviews.

3) *Classification Algorithms*: For our classification experiments, we used five algorithms. The machine learning algorithms we used are Decision Tree, Random Forest, Logistic Regression, Extra Tree (Extremely Randomized Tree) Classifier, and Multinomial Naïve Bayes. Logistic Regression was selected based on its performance in [17] which is a study about multiclass classification problem and it got best F1-scores in almost all datasets and scenarios. We also chose Multinomial Naïve Bayes based on Naïve Bayes classification results in [5] because it has good performance while classifying the sentiment of reviews from online retail shop in Indonesia with three polarities (positive, neagative, neutral). For Decision Tree, Random Forest, and Extra Tree, we selected them based on algorithms

⁶ <https://www.nltk.org/>

⁷ <https://github.com/har07/PySastrawi>

recommendation from scikit-learn⁸ for multilabel and multiclass problem. After we selected the machine learning algorithms, we compared the F1 scores they obtained, so, we can see which machine learning algorithm works best with our dataset.

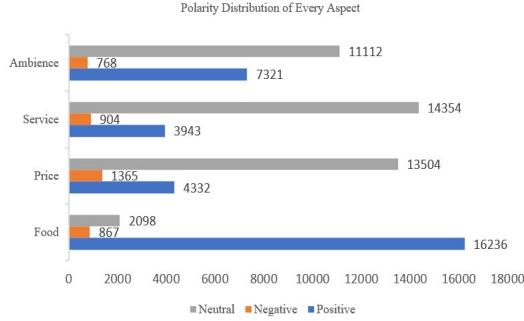


Figure 2. Polarity distribution in every aspect.

E. Evaluation

In these experiments, we used five classifiers: Random Forest (RF), Multinomial Naïve Bayes (NB), Logistic Regression (LR), Decision Tree (DT), and Extra Tree (ET); and cross validation as the validation technique and compared the F1 scores of machine learning models in all scenarios. Number of folds we used for the cross validation are 10 folds. This section shows the model performances in every aspect.

TABLE II. RESULT OF FIRST SCENARIO

Model	Food	Price	Service	Ambience
RF	78.52	75.32	78.15	73.03
NB	74.60	62.85	70.15	72.13
LR	80.57	74.16	80.33	75.42
DT	77.39	78.53	85.00	73.29
ET	78.33	78.35	83.06	74.83

The Table 2 shows the F1 scores of the models for the first scenario which is applying removing stopwords step without using stemming before entering the classification. From the table, we can see that LR has the highest F1 scores for *food* and *ambience* aspects which were 80.57% and 75.42% respectively. For *price* aspect, it was led by DT by obtaining 78.53%. Besides, it also gained highest score for *service* aspect with 85%.

TABLE III. RESULT OF SECOND SCENARIO

Model	Food	Price	Service	Ambience
RF	78.18	73.78	77.04	73.37
NB	77.64	65.04	72.32	74.57
LR	81.76	74.98	80.83	77.29
DT	78.19	76.64	84.78	73.24
ET	78.25	77.43	82.51	76.58

From Table 3, we can see the result of applying stemming technique but the stopwords were not removed. While comparing to Table 2, the scores that RF achieved were lower from that table except for the *ambience* aspect

⁸ <https://scikit-learn.org/stable/modules/multiclass.html>

which was increased around 0.34%. In contrast, the scores of NB and LR in every aspect were increased. However, for DT and ET, the scores they obtained were lower in *price* and *service* aspects compared to Table 2. Despite of that fact, the highest score for *price* aspect was obtained by ET while DT led score for *service* aspect.

TABLE IV. RESULT OF THIRD SCENARIO

Model	Food	Price	Service	Ambience
RF	78.30	73.34	77.33	73.14
NB	77.61	64.71	72.35	74.52
LR	81.72	74.09	80.30	76.69
DT	77.96	76.44	85.07	73.31
ET	78.39	76.62	81.85	76.18

The Table 4 shows the result of all algorithms in third scenario which is preprocessing without applying stemming and removing stopwords steps. In comparison with Table 3, DT has better scores in *service* and *ambience* aspects while ET achieved increasing score only in *food* aspect. RF obtained increasing scores for *food* and *service* aspects while NB scores only increased in *service* aspect. LR got lower scores in all aspects, but while compared to Table 2, the scores for *food* and *ambience* aspects were increased whereas the *price* score 0.07% lower and *service* aspect got 80.30%.

For last scenario, RF achieved highest scores compared to its scores in all previous scores, but *ambience* score lower than second and third scenarios. For NB, the scores were higher only when compared to first scenario. LR got highest score for *price* aspect but lower than its scores in second scenario for other aspects.

TABLE V. RESULT OF FORTH SCENARIO

Model	Food	Price	Service	Ambience
RF	78.54	75.91	79.10	73.09
NB	75.52	63.89	71.10	72.49
LR	80.91	75.69	80.48	76.12
DT	77.39	78.71	85.02	73.59
ET	78.21	78.49	82.98	75.65

DT attained highest scores for *price* and *service* aspects compared to all algorithms, but its *service* score still lower 0.05% than the score it achieved in previous scenario. For ET, it accomplished the second highest for *price* aspect, 0.22% behind DT.

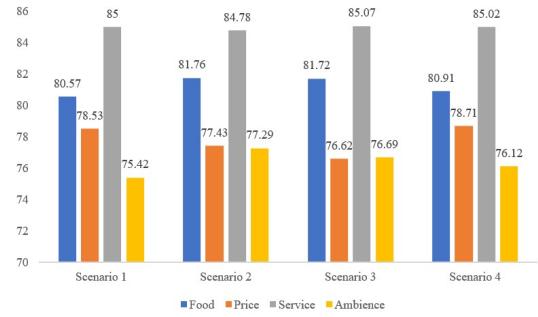


Figure 3. Comparison of highest scores in every aspect of all scenarios.

In summary, by seeing the Figure 3, combination of stemming and stopwords can affect the performance from models. The *food* aspect obtained highest score 81.76% by LR in scenario 2 which is using stemming without removing the stopwords. It seems that stopwords affect the *food* aspect more than stemming because its scores were higher every time the stopwords were not removed. However, its score was better when applied stemming and removing stopwords than only removing stopwords. In contrast, the *price* aspect got highest scores if the removing stopwords technique was used. Even, its best score obtained when using stemming and removing stopwords methods which is 78.71% attained by DT. For *service* aspect, DT achieved highest score for that aspect when removing stopwords and stemming were not applied, but its score also high when using both methods which is only 0.05% lower than without applying both. Besides, its score was better while the stopwords were removed than only using stemming without remove them. Moreover, for *ambience* aspect, same with *food* aspect, its highest score also obtained by LR in second scenario. Similar to *food* aspect, LR classified it better if stopwords were not removed.

Furthermore, the difference of scores that obtained by all classifiers for *food* and *ambience* were not far. This could be caused by the number of reviews from combination of positive and negative in *food* aspect that was higher than its neutral polarity. Consider the neutral polarity was combination of neutral and not mentioned aspect. *Ambience* aspect also has large number of positive and negative reviews compared to *service* and *price* aspects. So, by seeing the number of both positive and negative polarity, the variety of its samples was many. For *ambience*, even though the combination number of the positive and negative reviews was higher than *service* and *price*, but the score of the latter aspects were better than *ambience* for RF, DT, and ET. This might due to reviews rarely mention “*ambience*” or “*suasana*” directly, and people describe it more variative. For instance, “*ruangannya sepi jadi nyaman banget* (the room was dark, so it felt comfortable)” or “*dekorasinya unik jadi bagus banget buat foto tapi lantainya kotor* (the decoration was unique, so, it was cool for taking picture, but the floor was dirty)”. It can make the algorithm classified the review wrongly if in training data there are “Unique decoration”, “cool for taking picture” and “the floor was bit dirty and dusty” in separated reviews. However, *service* aspect has higher scores than *price* and *ambience* in almost all algorithms except for NB. This might highly cause by the reviews that contain *service* aspect usually straight forwarded about it and mention *service* aspect directly. For example, “*pelayannya sangat ramah* (the waitress was really kind)” or “*pelayanannya buruk* (the service was bad)”. Same goes with reviews mentioning *price* aspect which were straight forwarded like *service*, so the variation of the samples was not many. Besides, even though the data were small, models can classify it better.

IV. CONCLUSION AND FUTURE WORK

In this work, we have examined the performances of five machine learning algorithms which are Random Forest, Naïve, Bayes, Logistic, Regression, Decision Tree, and Extra Tree, to classify the opinion of the aspects that mentioned in code-mixed reviews. The aspects are *food*, *price*, *service*, and *ambience*. The evaluation conducted by making four scenarios namely removing stopwords without stemming, without removing stopwords but with stemming, without removing stopwords and stemming, and preprocessing with removing stopwords and stemming. The model performances were measured by using 10 folds cross validation, and the results show Logistic Regression achieved highest score for *food* (81.76%) and *ambience* (77.29%) aspects in second scenario. The highest score for *price* (78.71%) aspect was obtained by Decision Tree in last scenario, and *service* (85.07%) in third scenario. By seeing the results, it can be concluded that removing stopwords and stemming can affect the algorithm performances, specifically for small number of reviews.

In this work, we still using dataset with mixing languages. In the future, we will try to use one language in the same dataset by translating the dataset to either English or Indonesian to see how the language can affect the model performance. In addition, consider we still applied traditional machine learning algorithms in this research, in next study we will use Deep Learning and compare the result with the traditional machine learning models.

ACKNOWLEDGMENT

The authors acknowledge the PIT 9 research grant NKB-0010/UN2.R3.1/HKP.05.00/2019 from Directorate Research and Community Services, Universitas Indonesia.

REFERENCES

- [1] A. A. Mustafa and I. Budi, "Recommendation System Based on Item and User Similarity on Restaurants Directory Online," 2018 6th International Conference on Information and Communication Technology (ICoICT), Bandung, 2018, pp. 70-74, doi: 10.1109/ICoICT.2018.8528775.
- [2] A. Cahyadi and M. L. Khodra, "Aspect-Based Sentiment Analysis Using Convolutional Neural Network and Bidirectional Long Short-Term Memory," 2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA), Krabi, 2018, pp. 124-129, doi: 10.1109/ICAICTA.2018.8541300.
- [3] A. Pravalika, V. Oza, N. P. Meghana and S. S. Kamath, "Domain-specific sentiment analysis approaches for code-mixed social network data," 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Delhi, 2017, pp. 1-6, doi: 10.1109/ICCCNT.2017.8204074.
- [4] C. Fiarni, H. Maharani and E. Irawan, "Implementing Rule-based and Naive Bayes Algorithm on Incremental Sentiment Analysis System for Indonesian Online Transportation Services Review," 2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE), Kuta, 2018, pp. 597-602, doi: 10.1109/ICITEED.2018.8534912.
- [5] C. Fiarni, H. Maharani and R. Pratama, "Sentiment analysis system for Indonesia online retail shop review using hierarchy Naïve Bayes technique," 2016 4th International Conference on Information and Communication Technology (ICoICT), Bandung, 2016, pp. 1-6. doi: 10.1109/ICoICT.2016.7571912

- [6] D. Ekawati and M. L. Khodra, "Aspect-based sentiment analysis for Indonesian restaurant reviews," 2017 International Conference on Advanced Informatics, Concepts, Theory, and Applications (ICAICTA), Denpasar, 2017, pp. 1-6, doi: 10.1109/ICAICTA.2017.8090963.
- [7] D. H. Sasmita, A. F. Wicaksono, S. Louvan and M. Adriani, "Unsupervised aspect-based sentiment analysis on Indonesian restaurant reviews," 2017 International Conference on Asian Language Processing (IALP), Singapore, 2017, pp. 383-386, doi: 10.1109/IALP.2017.8300623.
- [8] F.Z. Tala, "A Study of Stemming Effects on Information Retrieval in Indonesian Indonesia". M.Sc. Thesis, Appendix D, Indonesia, 2003, pp. 39-46.
- [9] G. Ganu, N. Elhadad, and A. Marian, "Beyond the stars: Improving rating predictions using review text content," in WebDB, 2009.
- [10] I. Alfina, D. Sigmawaty, F. Nurhidayati, and A. N. Hidayanto, "Utilizing Hashtags for Sentiment Analysis of Tweets in The Political Domain," Proceedings of the 9th International Conference on Machine Learning and Computing (ICMLC), Singapore ,2017, pp. 43-47, doi: 10.1145/3055635.3056631.
- [11] K. Shalini, H. B. Ganesh, M. A. Kumar and K. P. Soman, "Sentiment Analysis for Code-Mixed Indian Social Media Text With Distributed Representation," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, 2018, pp. 1126-1131. doi: 10.1109/ICACCI.2018.8554835.
- [12] K. Zvarevashe and O. O. Olugbara, "A framework for sentiment analysis with opinion mining of hotel reviews," 2018 Conference on Information Communications Technology and Society (ICTAS), Durban, 2018, pp. 1-4. doi: 10.1109/ICTAS.2018.8368746.
- [13] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, "SemEval-2015 Task 12: Aspect Based Sentiment Analysis," Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval), Colorado, 2015, pp. 486–495. doi: 10.18653/v1/S15-2082.
- [14] M. Yasen and S. Tedmori, "Movies Reviews Sentiment Analysis and Classification," 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Jordan, 2019, pp. 860-865. doi: 10.1109/JEEIT.2019.8717422.
- [15] P. D. Turney, "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews," ACL '02 Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL), Pennsylvania, 2002, pp. 417-424. doi: 10.3115/1073083.1073153.
- [16] O. S. Jaya, H. Murfi, and S. Nurrohmah, "Analysis of Convolution Neural Network for Transfer Learning of Sentiment Analysis in Indonesian Tweets," Proceedings of the 2018 International Conference on Data Science and Information Technology (DSIT), Singapore, 2018, pp. 18-22. doi:10.1145/3239283.3239299.
- [17] S. Chawla and M. Mehrotra, "An Ensemble-Classifier Based Approach for Multiclass Emotion Classification of Short Text," 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2018, pp. 768-774. doi: 10.1109/ICRITO.2018.8748757
- [18] S. J. Putra, I. Khalil, M. N. Gunawan, R. Amin, and T. Sutabri, "A Hybrid Model for Social Media Sentiment Analysis for Indonesian Text," Proceedings of the 20th International Conference on Information Integration and Web-based Applications & Services (iiWAS2018), Indonesia, 2018, pp. 297-301. doi:10.1145/3282373.3282850.
- [19] Y. A. Amrania, M. Lazaar, and K. E. E. Kadiri, "Random Forest and Support Vector Machine based Hybrid Approach to Sentiment Analysis," Procedia Computer Science, vol 127, 2018, pp. 511-520. doi: <https://doi.org/10.1016/j.procs.2018.01.150>.

Using Convolutional Neural Network with BERT for Intent Determination

Changai He,Sibao Chen*

*Key Lab of IC&SP of MOE, School of Computer Science and Technology, Anhui University, Hefei 230601, Anhui, China
PKU Shenzhen Institute, Shenzhen, China. changai.he@imsl.org.cn*

Shilei Huang

*Shenzhen Raisound Technologies, Co., Ltd
Shenzhen, China
shilei.huang@imsl.org.cn*

Jian Zhang

*PKU Shenzhen Institute
Shenzhen, China
jian.zhang@imsl.org.cn*

Xiao Song*

*PKU Shenzhen Institute
Shenzhen, China
xiao.song@imsl.org.cn*

Abstract—We propose an Intent Determination (ID) method by combining the single-layer Convolutional Neural Network (CNN) with the Bidirectional Encoder Representations from Transformers (BERT). The ID task is usually treated as a classification issue and the user’s query statement is usually of short text type. It has been proven that CNN is suitable for conducting short text classification tasks. We utilize BERT as a sentence encoder, which can accurately get the context representation of a sentence. Our method improves the performance of ID with the powerful ability to capture semantic and long-distance dependencies in sentences. Our experimental results demonstrate that our model outperforms the state-of-the-art approach and improves the accuracy of 0.67% on the ATIS dataset. On the ground truth of the Chinese dataset, as the intent granularity increases, our method improves the accuracy by 15.99%, 4.75%, 4.69%, 6.29%, and 4.12% compared to the baseline.

Keywords-CNN; BERT; ID; context representation.

I. INTRODUCTION

Spoken dialogue system, voice assistant, automatic customer service, etc. are the hot spots of current natural language processing research [1]. The success of these applications depends not only on speech recognition but also on text understanding. In human-machine, Spoken Language Understanding (SLU) aims to automatically determinate the intent of the user as expressed in natural language. As an important part of the SLU system, ID is an accurate understanding of the user’s intents.

Generally speaking, the SLU system first needs to transcribe user’s voices into text via automatic speech recognition (ASR), or the input of users is text typed [2], then determine the user’s intents and combine the corresponding constraints, and finally these information can be delivered to the dialogue or task management system to satisfy the special needs of the user.

An example utterance sentence is shown in Table I, which uses the In/Out/Begin (IOB) and intent label representation. The intent label is used for the ID task, the slot label is used for the slot filling (SF) task. ID and SF are two major tasks in SLU. The sample is taken from the airline travel information system (ATIS) corpus [3], which is widely used in the SLU domain.

Table I
ATIS UTTERANCE EXAMPLE WITH INTENT AND SLOT ANNOTATION

Sentence	Slot label	Intent label
show	O	
flights	O	
from	O	
boston	B-dept	flight
to	O	
new	B-arr	
york	I-arr	
today	B-date	

The ID task is usually treated as a classification issue and the SF task as a sequence labeling issue [2]. Applying the information from one task to another can promote each other and achieve joint prediction. We will take more expense and effort to mark intent and slot than to mark intent only when we prepare training data for training a new SLU system. In practice, SF does not bring significant improvement in the ID task. We focus on only utilize the intent labels to improve the ID ability in this work.

In this work, we divide the ID tasks into two main stages, one is the text representation, another one is text classification.

1) *text representation*: In recent years, the unsupervised pre-training models have achieved excellent performance in many natural language tasks, such as Word2Vec, ELMo, BERT [4]. These models can be used to conduct the work of the text representation stage, among them the semantic information learned by BERT is more accurate and complete. BERT consists of multi-layer transformers, each of the transformer is composed of a self-attention sub-layer with multiple attention heads.

2) *text classification*: In the text classification stage, the text classification methods based on deep learning mainly include model based on CNN and RNN or their improved versions. Compared with the classification method based on traditional machine learning, the text classification method based on deep learning does not need to extract the key features of corpus text manually. The user’s query statement is usually a short text type. In short

text classification tasks, CNN is better than RNN [5]. In the ID task, the accuracy of the LSTM model is 1.48% higher than that of the RNN model [6], and GRU is an improvement of the LSTM model.

So we try to combine CNN, LSTM, and GRU with BERT respectively to deal with the ID tasks, where only the intent labels are used. We observe the performance of these models in conducting ID task on the ATIS corpus (English dataset) and the Chinese dataset. The experimental results demonstrate that the method of the CNN combined with BERT is superior to other methods. To our best knowledge, this work is the first time to utilize the CNN combined with BERT to conduct the issue of ID under the condition only the intent labels are utilized, and our model outperforms the state-of-the-art approaches on English dataset and achieves excellent performance compared to the baseline on the Chinese dataset.

II. RELATED WORK

Many researchers have performed research on ID task, the main directions of current research are divided into methods based on rule templates [7] and methods based on classification models [8], [9], [6], [10].

ID based on rule templates generally requires artificial construction of rule templates and category information to classify user intent. Different expressions lead to an increase in the number of rule templates, which requires a lot of manpower and resources. Although the method based on rule templates without a large amount of the training data, which can achieve very high accuracy, it cannot solve the high-cost issue of reconstructing the template when the intent text is changed.

The purpose of ID is to determinate the intent of a sentence, which can be regarded as a standard classification task [11]. The determination of the user's intent based on the classification model is mainly divided into the traditional machine learning methods and deep learning methods. The classification methods based on the traditional machine learning [10] [12] [13] [14], it is necessary to manually extract the key features of the corpus text, such as word features, n-gram, etc., and then implement the intent classification by training the intent classifier. This method is not only costly but also does not accurately understand the deep semantic information of the user.

In recent years, with the continuous development of deep learning, scholars have been explored CNN[15][16], recurrent neural networks (RNN)[6][17], long short-term memory (LSTM)[18][19], etc. to conduct the ID task. Sarikaya et al. used deep belief networks (DBNs) to handle routing classification problems [20]. Tur et al. proposed a sentence simplification method to deal with the issue of increased error rates caused by more complex, longer and more natural sentences [1]. As for joint work on ID and SF, Xu et al. proposed a neural network (NN) version of the triangular CRF (TriCRF) model, which utilized CNN to extract features and shared by two tasks [9]. Guo et al. found that recursive neural networks (RecNNs) have well performance in joint work of ID and SF [8]. Zhang

et al. utilized GRU to learn the representation of each time step to predict the label for each slot and used the max-pooling layer to capture the global features of sentences to perform the ID task [2]. Liu et al. proposed using the RNNs framework for performing joint work of ID, SF, and language modeling (LM) task on ATIS corpus [11]. But these models can not accurately capture the context information of sentences.

In this paper, we propose using CNN built on top of BERT for the ID task. BERT has been proven effective for learning contextualized word representations, and CNN is suitable for conducting short text classification tasks. We verify the effectiveness of our proposed method on the ATIS dataset compared with other methods, and the experimental results on the Chinese dataset show our method can achieve excellence performance.

III. METHODOLOGY

A. Task Description

In the previous work [8], [9], [11], most of them regard the ID task as a classification task, and we follow this idea in this work. Before training the ID model, we need to divide the intent of the sentence corpus, for example, I want to query the weather of Shenzhen tomorrow, we can mark this sentence with a query_weather label. After the marking is completed, the training, development, and test sets are divided according to a certain ratio. Finally, the data is fed to the network framework shown in Figure 1 to train the ID model. In detail, the ID task can be defined as a given marker dataset $S = (X, Y)$, where $x_i \in X$ is the utterance and $y_i \in Y$ is its intent label, and the ID task is an attempt to correctly associate the utterance x_i with the label y_i .

B. Framework Description

The overall framework of the proposed method in Figure 1. x_1, x_2, \dots, x_n represents a sentence of length n . The BERT part consists of 12 layers of the transformer. The transformer is composed of 12 different attention heads, each of which can focus on different types of component combinations. Input representation can unambiguously represent a text sentence in one token, then get the context word embedding through the transformer encoder. We utilize d to denote the dimension of the word embedding. If the sentence length is n , then the dimension of the sentence matrix is $n \times d$, we use the same zero-padding strategy as in [21]. The CNN part consists of three filter region sizes: 2, 3 and 4, with a total of 128 filters. The CNN part first performs convolutions on the sentence matrix generate feature maps, then 1-max pooling layer records the largest number from each feature map, a single variable feature vector is generated from all 128 maps, and the 128 features are connected to form a feature vector of the penultimate layer. The last softmax layer receives this feature vector as input and uses it to determinate the intent [22]. Here we assume $d = 5$, $n = 7$.

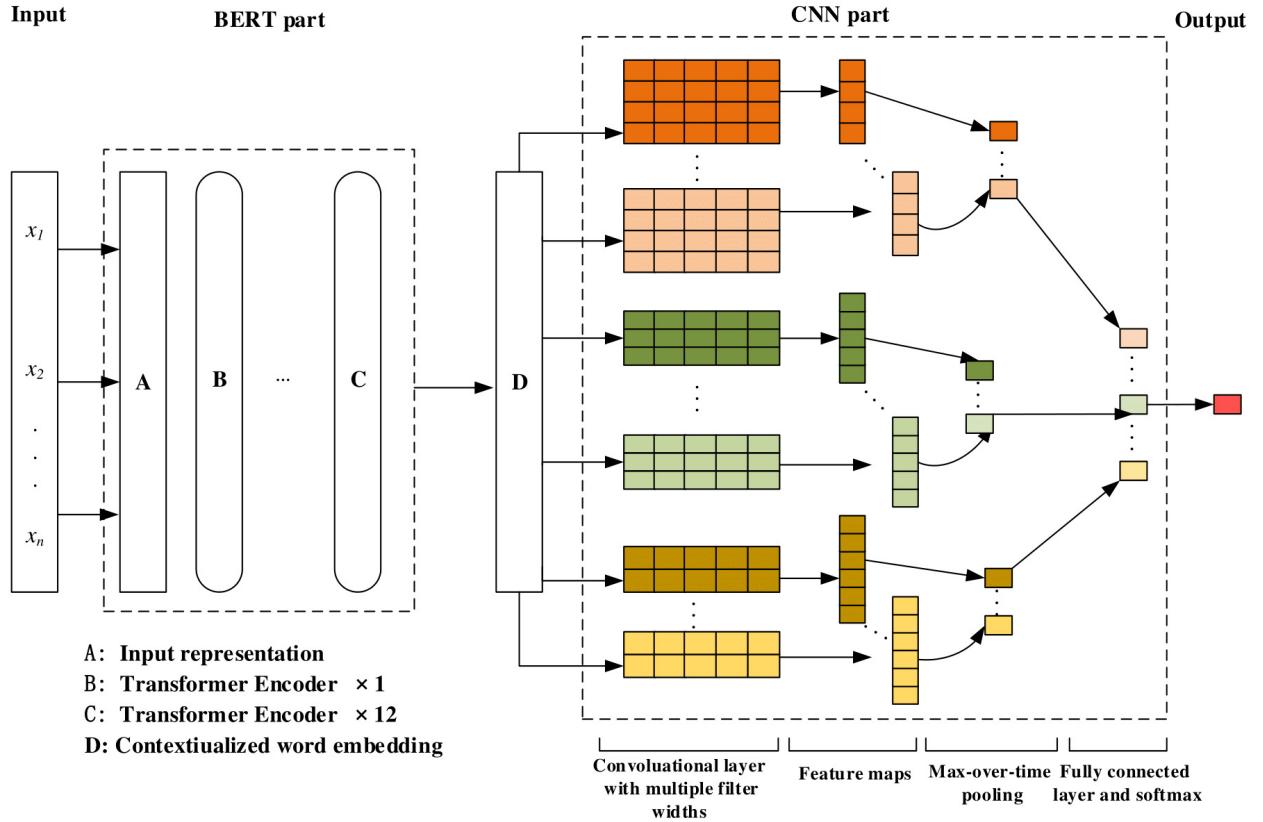


Figure 1. The framework of our method.

C. Our Method

In this section, we will describe our approach in detail to predict the user's intent. In order to conduct this issue, Liu et al. utilized the RNNs framework conduct joint work of ID, SF, and LM, which highlight is real-time parsing. This framework can provide an optimal intent, slot resolution and a prediction of a word for the moment T input to the present.

In fact, although slot filling brings some benefits to the ID task, it does not significantly improve the effect of ID and requires more expense and effort to mark the slot label. In view of the powerful semantic sentence understanding of BERT, which has been proven effective for learning contextualized word representations, and the user's query statement is usually of short text type. It has been proven that CNN is suitable for conducting short text classification tasks. We propose CNN built on top of BERT to conduct the issue of ID. As we all know that CNN congenital convolution operations are not suitable for sequence-level text, but the sentence-level word vector obtained by BERT has just made up for this shortcoming. BERT can get the context representation of a sentence, and the CNN can get the feature maps from the contextualized word embedding, and the prediction labels are given through the softmax layer.

1) *The BERT part:* Recently, researchers from the Google AI language team opened the source code of the

BERT project¹, a library for pre-training language representation. BERT has the ability to capture semantics and long-distance dependencies of sentences, and BERT enhances the generalization ability of the word vector model, so which can accurately capture the context information of sentences compared to previous pre-training models. The BERT part consists of 12-layer transformers and input representation, each of the transformer is composed of a self-attention sub-layer with multiple attention heads. Compared to the deep learning strategy previously ID, we utilize BERT to obtain the context representation of the sentence, which catch the context representation of the sentence easier and more accurately.

2) *The CNN part:* The CNN model has made remarkable achievements in computer vision and speech recognition, and it can also play a very important role in natural language processing. In a way, word is to text what pixel is to image, different from CNN in image recognition, the width of convolution kernel here is fixed. Parameters of each convolution kernel (filter) are Shared, which means that a filter can only identify the same type of features. Convolution has the function of local feature extraction, so CNN can be used to extract the key information similar to n-gram in sentences.

The feature map generated after convolution is no longer a matrix, but a column vector with a width of 1. Then

¹<https://github.com/google-research/bert>

Table II
ID ON THE YTBD DATASET OF DIFFERENT GRANULARITIES

Model	ID (acc)				
	10-intents	20-intents	30-intents	40-intents	50-intents
CNN	82.89%	85.21%	83.94%	81.04%	80.81%
BERT-LSTM	93.88%	89.37%	88.31%	86.51%	83.57%
BERT-GRU	93.16%	89.96%	87.68%	86.10%	83.57%
BERT-CNN(ours)	93.88%	89.96%	88.63%	87.33%	84.93%

1-max pooling is adopted in the pooling layer, that is, each filter gets the maximum value in the column vector. After the pooling layer is a fully connected layer, the operation of the hidden layer connecting the output layer, the softmax function is used to output the distribution probabilities of different intents. At the same time, a dropout operation is added to prevent overfitting. The key of short text classification is to accurately extract the main idea of sentences, and the way to extract the main idea of sentences is to extract the keywords of sentences as features. The convolution and pooling process of CNN is a feature extract process, so CNN is more suitable for conducting short text classification task.

IV. EXPERIMENTS

In our experiments, we used the independent training RNN intent model as a baseline on the ATIS corpus. Liu et al. joint online spoken language understanding and language modeling with RNN achieved the state-of-the-art performance in ID task on the ATIS corpus, at the same time, which provided the result of the independent training RNN intent model [11].

The user's query statement is usually a short text type. In short text classification tasks, CNN is better than RNN [5]. In the ID task, the accuracy of the LSTM model is 1.48% higher than that of the RNN model [6], and GRU is an improvement of the LSTM model. So we try to combine CNN, LSTM, and GRU with BERT respectively to deal with the ID tasks, where only the intent labels are used. At the same time, we do the experiment CNN without the BERT model on ATIS corpus, then we also utilize the above model to do experiments on the Chinese dataset, which consists of questions from Yuetongbao customer service platform², which is a public platform with 70 million users for Electronic Toll Collection (ETC) in China. we refer to the Chinese dataset as Yuetongbao Dataset (YTBD).

A. Dataset

1) ATIS: The ATIS corpus is widely used in SLU research. In this work, we followed the same ATIS corpus setup used in [3]. The ATIS contains 18 different intent labels and 5,781 sentences, there are 4,978 sentences for training, 893 sentences for testing, and then take 893 out of the training set as the development set. The average length of each piece of data is approximately 15 words. We report the results of 10-fold cross-validation.

²<https://www.96533.com/>

2) YTBD: There is no public dataset on the Chinese intent determination task, so we collected the questions on the Yuetongbao customer service platform and manual marked them. Most of the styles of these questions are colloquial. After screening, we manual marked 84,107 training data and 968 test data for intent. The average length of each piece of data is approximately 15 words. There are 50 kinds of intent labels in this dataset. We divide them into five data sets. The criteria for dividing are: first, take all the data of the first ten classes in the training set, and then take all the data of the first twenty classes in the training set, and so on. Do the same on the test set. Finally, we will get 5 training sets and 5 test sets with 10 intents, 20 intents, ..., 50 intents, and We take 15 percent of each training sets as development sets. The purpose of this is to observe if CNN with BERT framework also has an advantage in a fine-grained ID task.

Table III
ID ON THE ATIS DATASET

Model	ID (acc)
CNN	74.79%
online+RNN-LU (independent ID)	97.87%
online+RNN-LU (intent+slot)	98.43%
BERT-LSTM	96.31%
BERT-GRU	96.86%
BERT-CNN (ours)	98.54%

B. Evaluation Metrics

Our experiments were evaluated using accuracy, the metric accuracy for ID can be computed as:

$$acc = \frac{TP + TN}{TP + TN + FN + FP} \quad (1)$$

TP is the number of positive classes predicted as positive.
TN is the number of negative classes predicted as negative.
FN is the number of positive classes predicted as negative.
FP is the number of negative classes predicted as positive. Some sentences of the ATIS dataset contain multiple intents, and our process method is that so long as the identified intents belong to one of the multiple intents, they are counted as correct identification.

C. Results Analysis

In all the experiments shown in Table II and Table III, we set the max length of the input sentence to 32, and input 32 data per batch, we automatically stop

training, when the accuracy of more than 1000 steps is not improved.

Table II shows the accuracy scores of our method and the CNN model without BERT, the BERT-based LSTM model, the BERT-based GRU model on the YTBD dataset of different granularities for conducting the ID task. Our method achieves excellent performance. Comparing with the CNN model without BERT, our method improves the accuracy of ID at each granularity. Comparing with the BERT-LSTM model and the BERT-GRU model, our method achieves the best performance at each granularity, but the ability to determinate intent not improves much. Table III shows the accuracy scores of our method and the CNN model without BERT, online+RNN-LU (independent ID) model: independent training RNN intent model, online+RNN-LU (intent+slot) model: joint model with recurrent intent + slot label context, the BERT-based LSTM model, the BERT-based GRU model on the ATIS dataset for conducting the ID task. Our method is superior to the latest technology in ID. Comparing with the CNN model, the accuracy is improved by more than 23%. Comparing with the online+RNN-LU (independent ID) model, it illustrates the effectiveness of our method in dealing with the ID task, our method only improved by 0.67%, mainly because the baseline score is relatively high. Comparing with the online+RNN-LU (intent+slot) model, it illustrates that our method achieves a comparable result. Comparing with the BERT-LSTM model and the BERT-GRU model, it verifies that the CNN model is more suitable for dealing with short text classification issues.

V. CONCLUSION

In this work, we proposed a method by combining BERT with CNN to conduct the ID task. The result on the ATIS corpus shown that our model outperforms the state-of-the-art approaches. The results on the Chinese dataset shown that our model improves the accuracy by 15.99%, 4.75%, 4.69%, 6.29%, and 4.12% compare to the baseline as the intent granularity increases, and it also shows that even if only the intent label is used, the ID model with superior performance can be trained, which will help us save expense and effort in the process of conduct ID task.

We will apply the ID model trained on the YTBD dataset to an actual project, and plan to improve the robustness of the ID model in the future.

ACKNOWLEDGMENT

This work was supported in part by National Natural Science Foundation of China under Grant 61976004, and in part by Basic Research in Shenzhen (Discipline Layout) under Grant JCYJ20170817155939233.

REFERENCES

- [1] G. Tur, D. Hakkani-Tur, L. Heck, and S. Parthasarathy, “Sentence simplification for spoken language understanding,” in *IEEE International Conference on Acoustics*, 2011.
- [2] X. Zhang and H. Wang, “A joint model of intent determination and slot filling for spoken language understanding,” in *IJCAI*, 2016, pp. 2993–2999.
- [3] G. Tur, D. Hakkani-Tur, and L. Heck, “What is left to be understood in atis?” in *Spoken Language Technology Workshop*, 2011.
- [4] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 2018.
- [5] W. Yin, K. Kann, M. Yu, and H. Schütze, “Comparative study of cnn and rnn for natural language processing,” *arXiv preprint arXiv:1702.01923*, 2017.
- [6] S. V. Ravuri and A. Stolcke, “Recurrent neural network and lstm models for lexical utterance classification.” 2015.
- [7] J. Prager, D. Radev, E. Brown, A. Coden, and V. Samn, “The use of predictive annotation for question answering in trec8,” 1999.
- [8] D. Z. Guo, G. Tur, W. T. Yih, and G. Zweig, “Joint semantic utterance classification and slot filling with recursive neural networks,” in *Spoken Language Technology Workshop*, 2015.
- [9] P. Xu and R. Sarikaya, “Convolutional neural network based triangular crf for joint intent detection and slot filling,” in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2013, pp. 78–83.
- [10] P. Haffner, G. Tur, and J. H. Wright, “Optimizing svms for complex call classification,” in *IEEE International Conference on Acoustics*, 2003.
- [11] L. Bing and I. Lane, “Joint online spoken language understanding and language modeling with recurrent neural networks,” 2016.
- [12] A. SCHAPIRE and Y. Singer, “A boosting-based system for text classification,” *Machine Learning*, vol. 39, no. 1, p. 2, 2000.
- [13] K. M. Schneider, “A comparison of event models for naive bayes anti-spam e-mail filtering,” in *Tenth Conference on European Chapter of the Association for Computational Linguistics*, 2003.
- [14] A. Genkin, D. D. Lewis, and D. Madigan, “Large-scale bayesian logistic regression for text categorization,” *Technometrics*, vol. 49, no. 3, pp. 291–304, 2007.
- [15] H. C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, “Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning,” *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.
- [16] T. N. Sainath, B. Kingsbury, G. Saon, H. Soltau, A. R. Mohamed, G. Dahl, and B. Ramabhadran, “Deep convolutional neural networks for large-scale speech tasks,” *Neural Networks*, vol. 64, pp. 39–48, 2015.
- [17] Y. Gal and Z. Ghahramani, “A theoretically grounded application of dropout in recurrent neural networks,” *Statistics*, pp. 285–290, 2015.
- [18] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and F. F. Li, “Every moment counts: Dense detailed labeling of actions in complex videos,” *International Journal of Computer Vision*, vol. 126, no. 2-4, pp. 375–389, 2018.

- [19] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, “Predicting human eye fixations via an lstm-based saliency attentive model,” *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 5142–5154, 2016.
- [20] R. Sarikaya, G. E. Hinton, and B. Ramabhadran, “Deep belief nets for natural language call-routing,” in *IEEE International Conference on Acoustics*, 2011.
- [21] Y. Kim, “Convolutional neural networks for sentence classification,” *Eprint Arxiv*, 2014.
- [22] Y. Zhang and B. Wallace, “A sensitivity analysis of (and practitioners’ guide to) convolutional neural networks for sentence classification,” *arXiv preprint arXiv:1510.03820*, 2015.

What affects the difficulty of Chinese syntax?

Yueming Du^{*}, Lijiao Yang[§]

Institute of Chinese Information Processing, Beijing Normal University

UltraPower-BNU Joint Laboratory for Artificial Intelligence

Beijing, China

*Email: *dddyym@yeah.net, [§]yanglijiao@bnu.edu.cn*

Abstract—The traditional measurement of sentence difficulty only focuses on lexical features but neglects syntactic features. This paper takes 800 sentences in primary school Chinese textbooks published by People's Education Press as the research object and studies their syntactic features. We use random forest to select the top five important features and then employed SVM to do the classification experiment. The precision rate, recall rate and F-scored for the classification of 5 levels are respectively 50.42%, 50.40% and 50.41%, which indicates that the features we selected has practical value for the related research.

Keywords-Chinese syntax; random forest ; classification ; SVM;

I. INTRODUCTION

Linguistic complexity is a well-studied and multifaceted concept. Several measures are proposed in different frameworks, such as first and second language acquisition, typology and readability assessment. However, there exit two problems in previous studies. The first one is that most researches are based on the chapter level to study the difficulty of linguistic rather than based on a single sentence, which is the basic components of the text. Sentences are the basic unit of a text. Although, at present, there are many studies, which regard the length of sentences (Laughlin,1969), the use of simple sentences or complex sentences as important factors (Dechant,1961), the variables they measured were nothing more than vocabulary difficulty and the shallow sentence characteristics. The current situation, however, is that scholars study the super-outline words and phrases in sentences from the perspective of the overall number of texts, rather than as part of a sentence, let alone a specific sentence as a complete unit. The investigation of individual sentences is helpful to the development of AES and text readability. Second, researchers pay more attention to Chinese lexical features, but rarely to the features of Chinese syntax relate to the complexity. On the contrary, there are many studies on the syntactical difficulty in English. Hunt suggested that T-unit was a useful factor to exam English syntax (1965) . Norries and Ortega (2009) further summarized the multi-dimensional indicators of syntactic complexity into five sub-dimensions, including the use of subordinate structures, the overall complexity of sentences, the phrase expansion of clauses, the use of juxtaposition structures and the diversity, complexity and acquisition order of sentence structures. Lu selected 14 indicators to measure English syntactic complexity on the basis of quantitative analysis and qualitative analysis. However, whether the characteristics applied in English can

be used in Chinese and what characteristics are effective in measuring the complexity of Chinese syntax remains to be studied.

Syntax refers to the rules of conjunction formation in linguistic units. With the development of second language teaching, syntactic complexity has become one of the most important indicators to measure learners' overall language proficiency. However, in the task of language difficulty analysis in the field of Chinese, there are few studies on the measurement of its syntactic complexity.

In our study, we randomly selected 800 sentences from the primary school Chinese textbooks of the People's Education Press as the research object, and used HanLP¹ as well as StanfordCoreNLP² to extract their syntactic features automatically. At the same time, we asked three undergraduates with linguistic background to rate the syntactic complexity of the sentences and proofread the extracted syntactic features manually. On this basis, we further analyzed the contribution of each feature to the complexity of syntactic.

II. OUR METHOD

A. Preprocessing

The syntactic complexity explored in this paper is mainly in the field of modern Chinese, so classical Chinese and ancient poetry are not within the scope of this study. Because of the particularity of modern and contemporary poetry and nursery rhymes, their syntactic complexity is also beyond the scope of this paper.

We use primary school Chinese textbooks published by People's Education Publishing House (hereinafter referred to as "textbooks") as the basic corpus. After eliminating some texts that do not conform to the content of this study (ancient poems, classical Chinese, modern and contemporary poetry, nursery rhymes, etc.), we cut each text into sentences and randomly extract 800 sentences. At the same time, we use HanLP and StanfordCoreNLP to make word segmentation, part-of-speech tagging and dependent grammar on the 800 sentences.

We recruited 3 undergraduates with Chinese linguistic backgrounds and asked them to rate the syntactic complexity of the above data on a 5- point scale where 1 means "very easy" and 5 "very difficult", the distribution of syntactic levels is shown in Figure I. In addition, we asked them to check the results of word segmentation, part-of-speech tagging and dependency grammar analysis above, and to mark the inter-sentence relationship of each sentence.) We computed the Kappa coefficient reliability corresponding to the number of annotators who assigned the same judgment and obtained a reliability of 29%.

¹ <https://github.com/hankcs/HanLP>

² <https://stanfordnlp.github.io/CoreNLP/>

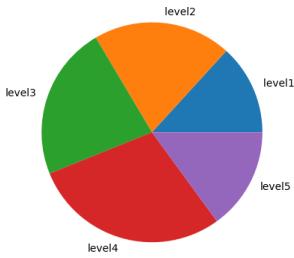


Figure I. Distribution of syntactic levels

B. Raw Sentence Feature

1) *sentence length*, i.e. average number of words per sentence (n_tokens). The basic unit of text is sentence, but the length of sentence is limited. Most researchers believe that sentence length can reflect the complexity of syntax, that is, the longer a sentence is, the more complex components it contains, and the more complex it tends to be. Therefore, we calculated the average sentence length in each level and the statistical results. We find that with the increase of the difficulty of the syntax, the average sentence length basically shows an upward trend.

2) *Punctuation*, the number of punctuation marks in a sentence. The more punctuation the sentence has, the more fragmented it is and the more difficult it is to understand. At the same time, punctuation to a certain extent reflects that the more insertions the sentence has, the more difficult it will be. The result shows the distribution of punctuation symbols.

C. Syntactic Features

1) *The max_depth of the whole parse tree*, a syntactic tree is a graphical representation of a sentence's syntactic structure. It shows us the complex nesting relationships among the syntactic components in the form of a tree graph. Therefore, the depth of the syntactic tree reflects the complexity of the sentence to a certain extent.

The methods for calculating the depth of the syntax tree are as follows: first, the dependency grammar analysis of sentences is performed by Stanford Core NLP; secondly, the syntax tree is transformed into a binary tree to calculate the number of nodes from the root node to the farthest leaf node; finally, the maximum depth of each syntax tree is obtained.

In order to visually reflect the corresponding relationship between the maximum depth of the syntactic tree and the syntactic difficulty, we calculated the maximum depth of the syntactic tree of each corpus in turn. As can be seen, the maximum depth of the parse tree increases with the increase of the syntactic difficulty.

2) *Modifier*: we will calculate the length of modifier and the ratio of length of modifier to sentence length. We label the modifier of 800 sentences manually and count the length of the modifier by rule method. On this basis, we get the values of these two features under different syntactic complexity. Both of the two features are positively

correlated with text difficulty, but the second feature, i.e. distribution of the ratio of length of modifier to sentence length, shows an upward trend of fluctuation with the increase of text difficulty.

3) *Number of inter-sentence relationship types*. The relationship between sentences involved in this study is different from the meaning types of complex sentences defined in Modern Chinese. We believe that nesting is one of the characteristics of Chinese, and this kind of nesting also reflects the complexity of syntax to a certain extent.

Therefore, in order to better carry out the study, we will also include the form of complex sentences and the structural relationship of phrases involved in a single sentence. When it comes to specific classification, we still use the definition of the relationship between complex sentences in modern Chinese. We can see that with the increasing of the syntactic complexity, the number of inter-sentence relationship types also goes on.

4) *Number of the special phrases*: prepositional phrase and pivotal phrase. Complex sentences are usually longer, structurally intense, and impose a higher cognitive burden on the reader, so we developed phrasal features that measure structural complexity, including the proportions of noun phrases, prepositional phrases. Generally speaking, the two features also go on with the increase of syntactic difficulty, but the upward trend is not obvious.

5) *Special parts of speech*: Nouns and verbs are two important notional words. Their density will affect the reader's cognition to a certain extent. We calculate the densities of nouns and verbs in sentences and the ratio of noun verbs to verbs. From the result, we reach the same conclusion, that is to say, both of the two features are positively correlated with text difficulty.

III. EXPERIMENT AND RESULT ANALYSIS

A. Feature Selection

In this paper, we chose random forest to make feature selection. Because of the inherent randomness of stochastic forests, the model may give different importance weights for each feature. However, by training the model many times, that is, by selecting a certain number of features and reserving the intersection of the last feature each time, we can cycle a certain number of times, so that we can finally get a certain number of features that have an important contribution to the impact of classification tasks.

We use random forests to rank the importance of the features mentioned in the previous section, and the results are shown in Figure II . 0-9 in abscissa is average sentence length, average punctuation marks, max_depth of the parse tree, the length of modifier, the ratio of length of modifier to sentence length, inter-sentence relationship types, number of pivotal phrases, number of prepositional phrases, the densities of nouns and verbs.

As can be seen from the figure below, the top five features are average sentence length, the length of modifier, inter-sentence relationship types, the ratio of length of modifier to sentence length, and the density of verbs.

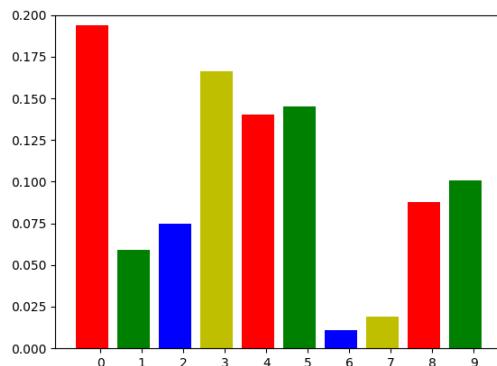


Figure II . The importance of features

B. Process and result of the experiment

800 sentences are divided into training set and testing set according to 7:3 ratios. In order to ensure the accuracy of training and avoid over-fitting, we use 5-fold cross validation within the training set beforehand. Subsequently, we use Support Vector Machine (SVM) as the classifier, adding the top five features in the experiment process successively. The best experiment results are shown in Table I.

Table I: Result of the feature experiment

	P(%)	R(%)	F1(%)
F0	32.19	33.75	32.95
F0+F3	39.58	39.58	39.58
F0+F3+F5	44.16	43.10	43.62
F0+F3+F5+F4	45.25	45.17	45.21
F0+F3+F5+F4+F9	50.42	50.40	50.41

As Table I shown, five features that have been added successively make the precision rate, recall rate and F-score increase gradually. This illustrates that five features have different degrees of influence on classification of Chinese syntactic complexity.

We plotted the confusion matrix of the sixth classification experiment, as shown in Figure III. It can be seen that the classification of level5 level1 and level4 reached the best results.

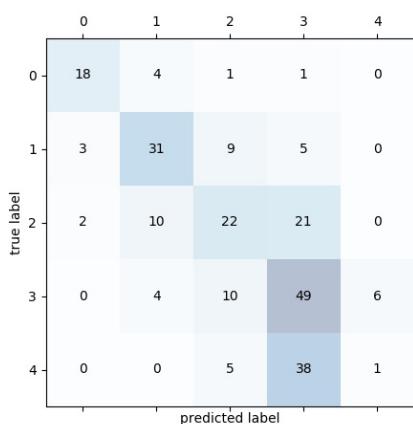


Figure III. the confusion matrix of the experiment

IV. CONCLUSION

In this paper, we take Chinese sentences as the research object and study the Chinese syntactic complexity. Firstly, we chose so many features affecting sentence complexity. Then, we propose to use random forest algorithm to extract features. Finally, SVM classification algorithm is employed to verify the validity of the above features to the syntactic complexity. The precision rate, recall rate and F-scored for the classification of 5 levels can reach over 50 percent, which indicates that the features we selected has practical value for the related research.

In our further study, we will expand the scale of corpus and explore more effective features in order to better explore the factors affecting the syntactic complexity of Chinese.

REFERENCES

- [1] Laughlin G H M. SMOG Grading-A new readability formula[J]. Journal of Reading, 1969, 12(8):639-646.
- [2] Dechant, E.V.& Smith, H. P. Psychology in Teaching Reading[M]New Jersey: Prentice-Hall, InC.,1961:134-150.
- [3] Hunt, K. Grammatical structures written at three grade levels[J]. Elementary Secondary Education. 1965.
- [4] Norris J M, Ortega L. Towards an Organic Approach to Investigating CAF in Instructed SLA: The Case of Complexity[J]. Applied Linguistics,2009, 30(4):555-578.
- [5] Lu X. Automatic analysis of syntactic complexity in second language writing[J]. International Journal of Corpus Linguistics,2010,15(4):474-496.
- [6] Caylor John S, Methodologies for determining reading requirements of military occupational specialties [J]. Adult Literacy,1973:81
- [7] Kincaid J P,Fishburn R P,Chisson B S.Derivation of new readability formulas for navy enlisted personnel[J].Adult Basic Education,1975:49.
- [8] Cortes C,Vapnik V. Support-vector networks[J].Machine Learning,1995,20(3):273-297.
- [9] Petersen S E,Ostendorf M.A machine learning approach to reading level assessment[J].Computer Speech & Language,2009,23(1):89-106.
- [10] Alusio S,et al. Readability assessment for text simplification[C]// NAACL Hlt 2010 15th Workshop on Innovative Use NLP for Building Educational Applications. Association for Computational Linguistics,2010:1-9.
- [11] Schwarm S E, Ostendorf M. Reading level assessment using support vector machines and statistical language model[C]// Meeting on Association for Computational Linguistics. Association for Computational Linguistics,2005:523-530.
- [12] Vogel M, Washburne C. An objective method of deterring grade placement of children's reading material[J]. Elementary School Journal, 1928, 28(5): 373-381.
- [13] Ortega, L. Syntactic Complexity Measures and their Relationship to L2 Proficiency: A Research Synthesis

- of College-level L2 Writing[J]. *Applied Linguistics*, 2003, 24(4):492-518.
- [14] Bachman, L.F. *Fundamental considerations in language testing*. Oxford: Oxford University Press, 1990.
- [15] Wolfe-Quintero,K.,S. Inagaki & H.Kim. *Second Language Development in Writing: Measures of Fluency, Accuracy and Complexity* [M]. Honolulu: University of Hawaii, Second Language Teaching and Curriculum Center, 1998.
- [16] Betts, E. A. *Difficulty: Its Application to the Elementary School* [J]. *Journal of Educational Research*, 1949(42)438—59.
- [17] Hunt, K. *Grammatical structures written at three grade levels*[J]. *Elementary Secondary Education*. 1965.
- [18] Dechant, E.V.& Smith, H. P. *Psychology in Teaching Reading*[M]New Jersey: Prentice-Hall, InC.,1961:134-150.
- [19] Lu X. *Automatic analysis of syntactic complexity in second language writing*[J]. *International Journal of Corpus Linguistics*,2010,15(4):474-496
- [20] Norris J M, Ortega L. *Towards an Organic Approach to Investigating CAF in Instructed SLA: The Case of Complexity*[J]. *Applied Linguistics*,2009, 30(4):555-578.
- [21] Biber D, Gray B, Poonpon K. *Should We Use Characteristics of Conversation to Measure Grammatical Complexity in L2 Writing Development*[J]. *Tesol Quarterly*, 2012, 45(1):5-35.
- [22] Dominique Bruna et al. *Is this Sentence Difficult? Do you Agree?* [C]//*Proceedings of Conference on Empirical Methods in Natural Language Processing*, 2018,2690-2699.
- [23] Jin, H G. *Syntactic maturity in second language writings: A case of Chinese as a foreign language(CFL)*[J]. *Journal of the Chinese Language Teachers Association*, 2007, 42(1):27-54. 2007.
- [24] Jiang W, *Measurements of Development in L2 Written Production: The Case of L2 Chinese*[J]. *Applied Linguistics*, 2012, 34(1):1-24.
- [25] McNamara, D. S. et al. *Automated Evaluation of Text Discourse with Coh-Metrix*[M]. Cambridge: Cambridge University Press,2014
- [26] Sheehan K M, Kostin I, Napolitano D, et al. *The TextEvaluator tool: Helping teachers and test developers select texts for user in instruction and assessment*[J]. *Elementary School Journal*,2014,115(2):184-209.

Classified Description and Application of Chinese Constitutive Role

Mengxiang Wang
 Teachers' college
 Beijing Union University
 Beijing, China
 e-mail:wmx1314-@126.com

Cuiyan Ma
 Research Institute of Information Technology
 Tsinghua University
 Beijing, China
 e-mail:mcymcy402@163.com

Abstract— Constitutive role is one of the 4 qualia roles , which expresses a kind of constitutive relationship between nouns. According to the original definition and description characteristics, this paper divides the constitutive roles into two categories: materials and components. At the same time, combined with the previous methods of extracting the role automatically, this paper optimizes the method of extracting the role automatically. Relying on auxiliary grammatical constructions, we extract noun-noun pairs from large-scale corpus to extract descriptive features of constitutive roles, and then classifies these descriptive knowledge by manual double-blind proofreading. Finally, the author discusses the application of Chinese constitutive roles in word-formational analysis , syntactic analysis and synonym discrimination.

Keywords- *constitutive role; qualia role; classified description*

I. INTRODUCTION

Constitutive role was proposed by the Pustejovsky in describing the qualia role structure of nouns. The qualia role structure is one of the core content of the Generative Lexicon (GL) Theory. (Pustejovsky 1991). The constitutive role is mainly to describe what is X made of / from, which is a relationship between a noun and another noun.

When the GL Theory was introduced into China, because of the difference between Chinese and English grammar system, some Chinese scholars have different understanding of the GT theory, including the number and definition of qualia structures. i.e In the beginning, when the Pustejovsky introduced the qialia structure, there were 4 qualia role: Constitutive, Agentive, Formal and Telic. Later according to the characteristics of Chinese grammar the Chinese scholar Yuan (2014) proposes a descriptive system of Chinese noun's qualia structure, which defines ten qualia roles (Constitutive, Orientation, Evaluation, Material, Formal, Agentive, Action, Telic, Handle). Yuan(2014) believed that besides the internal constitutive property, the constitutive roles can describe the size, shape, dimension, color and orientation of objects, and even include the arguments involved in nouns. For example, the constitutive role of "opinion" includes "presenter, targeted matter",while the constitutive role of "mother" includes "address person, be addressed person". Yuan's description of constitutive roles contain a special purpose, but it is too complex, which is not in line with Pustejovsky's original idea of describing constitutive role. At the same time, it is difficult for extracting the relevant language knowledge, so it is not suitable for language processing.

In view of the characteristics of constitutive roles, this paper will classify the constitutive roles internally, and optimize the method of acquisition of constitutive roles by predecessors. Last but most important, we discuss their applications in word-formational analysis, syntax analysis, synonyms discrimination.

II. THE CLASSIFIED DESCRIPTION OF CONSTITUTIVE ROLE IN CHINESE NOUN

From the original idea, the constitutive role is to describe what is X made of. i.e, If the house is made of wood, the noun "wood" is the constitutive role of "house". But Chinese people's cognitive habits are different from the West, "made of" and "made from" is the same in Chinese. Therefore, we must classify the Chinese constitutive roles.

In fact, According to Pustejovsky's description of the concept of the constitutive role, the relationship of the constitutive roles is similar to the "part-of". Odell(1998) proposed that the "part-of" relationship contains six forms of expression.

As we can see from the table1, the types of "part-of" is still complex. From the point of view of applications, such a subdivision is not necessary. We see that apart from "Material-objects" type, the rest can be classified into one category: "Components-objects". Because the chair legs can be seen as a component of chair. The relation of "car wheel and car", "state and country", "display and computer", "husband and couple" all can be seen as "components-objects". Based on this, we merge the six forms of composition into two types: materials and components. The corresponding expressions of our constitutive roles are also semantic. We can describe the constitutive roles of a noun as The following two types:

(1) 桌子[木头, 石头, 竹…]
 (desk[wood, stone, bamboo...]).

Material-objects type

(2) 电视机[电路板, 显示屏, 喇叭…]
 (television [circuit board, screen, horn...]).

Components-objects type

III. THE EXTRACTION OF CONSTITUTE ROLE OF CHINESE NOUNS

As the extraction and description of Chinese qualia roles are in the initial stage, the extraction of Chinese qualia roles is basically based on artificial extraction. At present, only Wang(2016) has made an attempt to automatically extract the constitutive role.

Table 1: Odell's classification of "part-of"

Representation of "part-of" relationships	Annotation	Example
Component-integral object	An object consisting of a necessary component (Chair legs, chair surface, chair back are all necessary parts of the chair)	The chair is composed of a chair legs, a chair surface, and a chair back. (Chair legs, chair surface, chair back are all necessary parts of the chair)
Material-object	An object consisting of a necessary material	The chair is made of wood.(wood is the necessary by making chair)
Portion-object	An object mostly consisting of portion	A car is made up of wheels and a frame. (The wheels and frame are most of the components of the car, but the car has other devices besides the wheels and frame.)
Place-area	An area is made up of places or location	China was made up of nine states including jizhou, qingzhou, jingzhou.(the jizhou, qingzhou, jingzhou consist of the whole china, in ancient times)
Member-bunch	An object consisting of a collection of members (member changes do not affect the whole unless the collection of members changes).	The computer consists of a host and a display.(the host and the display are a collection, including network cards, graphics cards, and other components
Member-partnership	An object consisting of a collection of members' partner.(the change of the member will affect the whole)	Couples are composed of husband and wife(husband and wife are interdependent and become couple)

On that time, Wang(2016) did not take into account the fact that the pattern characteristics of the constitutive role, only just use "由……构成 (be consist of)" pattern, which can ensure the accuracy of extracting, but many information were missing. Therefore, this paper optimizes on the basis of its method. Based on more specific keywords and syntactic configuration matching, we extract specific noun pairs by the shortest path rule, and then extend the synonyms to obtain the knowledge of the constitutive roles. The specific steps are as follows:

Steps 1 pattern recognition

First, we use special words and syntactic constructions to match the desired pairs of nouns from massive texts. This process is called pattern recognition. The special words, syntactic constructions and patterns are shown in the table2.

Table 2: Key Words and Syntactic Constructions In the constitutive role pattern recognition

Special words or constructions	Example
有 (have/there be)	安徽 有 黄山 Anhui there is Huangshan Mountain “There is a Huangshan Mountain in Anhui”
构成/组成 (Constitute/consists)	这支军队 由 农民 组成 Army peasantsconsists “The army consists of peasants”
分为 (divide)	襄樊 分为 襄城 和 樊城 Xiangfan devide Xiang city and Fan city “The Xiangfan is divided into Xiang city and Fan city.”
用……做 (be madeof/ from)	这酒 是 用 高粱 做的 Wine is Sorghum made “This wine is made of sorghum.”
比 (例) 如 such as	我 喜欢 吃 北方菜, 比如 锅包肉、地三鲜 I like eat northern dishes such as pot-fried meat, Disanxian. “I like to eat northern dishes, such as pot-fried meat, Disanxian.”
偏正结构 modifier-headword NP+NP Combinations	荞麦 饼 Buck wheat cake “a kind of cake that is made of buckwheat”

The single-line nouns are the constitutive roles of double-line nouns. It needs to be noted that some patterns are quite generic and using them to extract information can result in false positives (like the "have" pattern), so we use artificial methods to distinguish internal differences such as "have" and "there be" to avoid the noise.

Steps 2 collect (n_i - n_j) nouns pairs. To comb and clean up the extracted nouns pairs according to their co-occurrence rate.

This is mainly due to the fact that once the matching pattern increases, it will cause more data sparse. For example, the semantic complexity covered by "有 (have)" is much greater than the semantic relationship of the constitutive characters. But "有 (have)" as a keyword can match many compositions, so on the one hand, we have to spread it smoothly, and on the other hand we have to carefully extract. We think that coincidence(CoR) is a relatively simple and effective method. A certain number of (n_i - n_j) co-occurrence may indicate a semantic link between two nouns, for example a qualia link. So we calculate the correlation between the nouns (n_i - n_j) from the Chinese GigaWord Corpus and SogouCA Corpus by following equation 1, and rank the unlabelled element (n_j) of (n_i) .

$$\text{CoR}(n_i, n_j) = \frac{2 * \text{Sum}(n_i, n_j)}{\text{Sum}(n_i) + \text{Sum}(n_j)} \quad (1)$$

$\text{Sum}(n_i, n_j)$ presents co-occurrence number of (n_i , n_j). $\text{Sum}(n_i)$ and $\text{Sum}(n_j)$ means the number of n_i and n_j .

Steps 3 Extract the shortest path pattern of the matched (n_i - n_j)nouns from the corpus and to determine the (n_i - n_j)nouns set with constitutive relationship.

Steps 4 Through Baidu search engine, expanding the collection of constitutive roles of (n_i - n_j)nouns.

Steps 5 To extend the knowledge of the constitutive roles through the synonym set

Steps 6 Binary classification of (n_i - n_j) nouns with constitutive relationship.

Steps 3,4,5 can refer to Wang(2016) in detail. For steps 6, we use artificial double-blind notation to classify all noun terms into 2 types(Components-objects or Material-objects), which contained constitutive role. In the process of labeling, in addition to judging directly from semantics, we also use some syntactic patterns to assist in judgment. For example, pattern "composed of ... (由.....构成)" is generally classified as a Components-objects class.

IV. THE APPLICATIONS OF CONSTITUTIVE ROLE OF CHINESE NOUNS

4.1 Application in word-for mational analysis

Noun-noun compound phrases are sometimes not included in dictionaries. What is more, the grammatical type of these words are complex. For example: “石头桌子 (stone desk)” means “the desk is made of stone”, while “中国桌子 (china desk)” means “Chinese desk”. Song Peijie (2014) pointed out there are 3 types of the grammatical structure patterns of the noun-noun compounds: head word-modifier pattern, modifier-headword pattern and parataxis pattern. We find that if the N_1 and N_2 are constitutive relationship, the grammatical structure of “noun+noun compounds” are only 2: headword-modifier pattern and modifier-headword pattern. For the analysis of these “noun+noun compounds”, Song Peijie (2014) pointed out that “implied predicates are the key to the analysis of the internal semantic relations of noun-noun compound words”. Weixue (2013) point out that each of Noun-Noun Compounds contains a event, and the event is triggered by implicit predicates. We found that if the N_1 is N_2 ’s constitutive role, the implicit predicates are almost “make”, “constitute”, “divide”. We can make the following inductions:

If N_1 is the constitutive role of N_2 , the NN structure can be decomposed as follows:

(1) $N_1+N_2=N_1+v$ (做/组成 (make or constitute))+ N_2

If N_2 is the constitutive role of N_1 , the NN structure can be decomposed as follows:

(2) $N_1+N_2=N_1+\text{的}(\text{de})+N_2$

For example, to “石头桌子 (stone desk)”, “石头 (stone) ” is the constitutive role of “桌子 (desk) ”, so the structure can be decomposed as “石头做的桌子 (stone+做 (make) +桌子 (desk))”, which use the Form (1).

To “人脸 (people face)”, “脸 (face) ” is the constitutive role of “人 (people) ”, so the structure can be decomposed as “人 (people) +的 (de) +脸 (face) ”, which use the Form (2).

We have extracted 4610 disyllabic non-juxtaposed noun-noun compounds from the Modern Chinese

Dictionary (2012 edition) and found that there are 2143 compounds which N_1 and N_2 are constitutive relationship. Among them, 2012 compounds belong to “ N_1 is N_2 ’s constitutive role” type, of which 1822 phrases (90.6%) can be interpreted in the form of (1). But there are only 131 compounds belong to “ N_2 is the constitutive role of N_1 ” type, of which 125 phrases (95.4%) can be interpreted in the form of (2).

4.2 Application in syntactic analysis

The structure “S+V1+O1+V2+O2” is usually analyzed in 2 ways:

- (A) S+V1+(O1+V2+O2);
- (B) (S+V1+O1)+V2+O2;

It is difficult to distinguish it through the existing NLP software.

But if the O2 is O1’s constitutive role, the“(O1+V2+O2)” usually cannot exist (Unless the V2 is relation verb or certain verb). We can use this feature to analysis the sentences more correctly. The following figure can express this process:

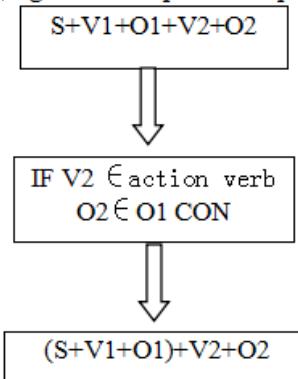


Figure 4:Automatic syntactic analysis of “S+V1+O1+V2+O2”

For example, We can see these 2 sentences:

(1)美国警告俄罗斯将插手委内瑞拉内政

USA warn Russia will intervene in Venezuela's internal affairs

(1a)USA warning: Russia will intervene in Venezuela's internal affairs[S+V1+(O1+V2+O2);]

(1b)USA warns of Russia, the USA will intervene in Venezuela's internal affairs. [(S+V1+O1)+V2+O2]

(2)美国 警告 委内瑞拉 将 颠覆 马杜罗政权

USA warns of Venezuela, the USA will subvert Maduro regime|(S+V1+O1)+V2+O2|

In the sentence (2), S is USA, V1 is warns, O1 is Venezuela, V2 is subvert,O2 is Maduro regime, V2 is action verb, and Maduro regime can be considered as the constitutive role of “Venezuela”. We must output “(S+V1+O1)+V2+O2” automatically.

But if we don’t tell the computer this constitutive knowledge, the result will be another one.

We use the Language Technology Platform(developed by Harbin Institute of technology, China) to analyze the syntactic structure of the Chinese sentence, the result is in Figure 5:

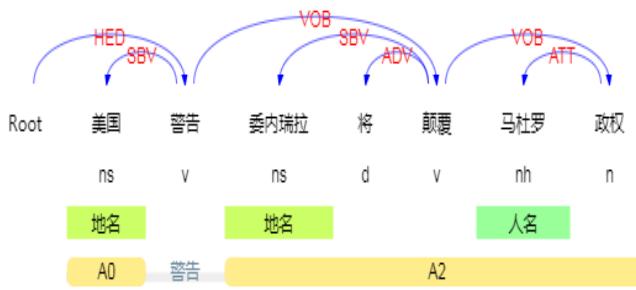


Figure 5:Syntactic analysis results by machine
by Harbin Institute of technology, China

From the analysis chart of this platform, we can see that the actors of “颠覆 (subvert)” are labelled “委内瑞拉 (Venezuela)”, not the “美国 (USA)”. If we add the constitutive knowledge of nouns, this mistake can be avoided. In this case, the constitutive role feature can help us to improve the syntactic analysis result.

4.3 Application in synonyms discrimination

Traditional synonyms discrimination is usually differentiated by sentence interpretation or sememe analysis, which usually focuses on the rational meaning of words. But apart from the rational meaning, it is possible that ordinary people pay more attention to the usage of words, which will involve grammar and pragmatics.

As one of descriptor of the noun ontological meanings, the constitutive role is contained in every noun. We find that if we can construct a complete system of constitutive role's description, We can distinguish some noun synonyms much better.

For example, in our constitutive role's description system, we describe the “奥秘(mystery)” vs “秘密(secret)” like this:

(1)The constitutive role of “奥秘(mystery)”:

The universe (mystery), the stars (mystery), the sun (mystery), natural phenomena (mystery), Luminescence (mystery), physiology (mystery), microorganism (mystery), heredity (mystery), etc.

(2)The constitutive role of “秘密(secret)” :

Economy (Secret), Military (Secret), Diplomacy (Secret), Politics (Secret), Commerce (Secret), Nuclear (Secret), State (Secret), Communication (Secret) etc.

From the constitutive knowledge of the “奥秘(mystery)” and “秘密(secret)”, we can find that “奥秘(mystery)” often describe the object things in nature, while the “秘密(secret)” tends to describe man or man-made results.

In this way, we can analyze the differences of synonyms very concretely, especially to the usage environment.

V. CONCLUSION

This paper just made a classified description of constitutive roles of Chinese nouns, improved an approach for the acquisition of constitutive role for Chinese nouns based on predecessors, and then discuss edits application in word-formational analysis, syntax analysis, synonyms discrimination. In the future, we will

discuss how to expand the specific application of constitutive role in nature language processing.

ACKNOWLEDGMENT

This research was partly supported by National Science Foundation of China(No.61602040)and Supporting Plan for Cultivating High Level Teachers in Colleges and Universities in Beijing(CIT&TCD201904072)

REFERENCES

- [1] Cimiano P and Wenderoth J. 2. Automatic Acquisition of Ranked Qualia Structures from the Web. In Proceedings of the Meeting of the Association for Computational Linguistics, 2007, pages:888-895.
- [2] Claveau V and Sébillot P. Automatic Acquisition of GL Resources, Using an Explanatory, Symbolic Technique. Advances in Generative Lexicon Theory, 2013, pages 431-454.
- [3] E.M. Voorhees. Query expansion using lexical semantic relations. In Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, pages 61–69.1994
- [4] Gonzalez M, Lima V L S D. Redefining Traditional Lexical Semantic Relations with Qualia Information. Revista Palavra, 2004.
- [5] Huang, Chu-Ren and Adam Kilgarriff, et al.. Chinese Sketch Engine and the Extraction of Collocations. In Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing. 2005
- [6] LI Qiang. Semantic Analysis and Construe of Adjective-noun Composition Based on the Qualia Structure. Chinese Language Learning,2014
- [7] Li Qiang and Yuan Yulin. A Qualia role Based Approach to the Analysis of Chinese Synonymous Nouns. Chinese Teaching in the world,2014, 04: pages 519-531.
- [8] Monachini M, Moruzzi V, and Quochi V, et al. Lexical Relations and Domain Knowledge: The BioLexicon Meets the Qualia Structure.2007
- [9] Pustejovsky. The generative lexicon. Computational Linguistics, 1991,17(4): pages 209–441.
- [10] Pustejovsky, P. Anick, and S. Bergler. Lexical semantic techniques for corpus analysis. Computational Linguistics, Special Issue on Using Large Corpora II,1993, 19(2): pages 331–358.]
- [11] Song Peijie Study on the Chinese noun-noun compounds Doctoral dissertation Of Huazhong university of science and technology ,2014
- [12] Voorhees. Query expansion using lexical semantic relations. In Proceedings of the 17th annual international ACMSIGIR conference on Research and development in information retrieval, 1994, pages 61–69.
- [13] Weixue and Yuan Yulin Constructing the Interpretation Templates for Chinese Noun Compounds Based on the Semantic Classes and the Qualia Roles of the Component Nouns, Chinese Teaching in the World,2013
- [14] Wang Mengxiang,Wang Houfeng and Zheng Lvexing, The extraction and application of constitutive role of nouns in Chinese,IALP2016
- [15] Yuan Y. On a descriptive system of qualia structure of Chinese nouns and its application in parsing complex Chinese grammatical phenomena. Contemporary Linguistics,2014
- [16] Zeng, D., Liu, K., and Lai, S. et al. Relation classification via convolutional deep neural network. In Proceedings of COLING, pages: 2335–2344,2014
- [17] Zheng Lvexing, Wang Houfeng, and Lv Xueqiang. Improving Chinese Dependency Parsing with Lexical Semantic Features. Natural Language Processing and Chinese Computing ,2015

Design and Implementation of Burmese Speech Synthesis System Based on HMM-DNN

Mengyuan Liu

School of Information Science and Engineering
Yunnan University
Kunming, China
e-mail: liumeyu@qq.com

Jian Yang

School of Information Science and Engineering
Yunnan University
Kunming, China
e-mail: jianyang@ynu.edu.cn

Abstract—The research and application of speech synthesis in Chinese and English are widely used. However, most non-universal languages have relatively few electronic language resources, and speech synthesis research is lagging behind. Burmese is a type of alphabetic writing, and Burmese belongs to Tibetan-Burmese branch of the Sino-Tibetan language. In order to develop the Burmese speech synthesis application system, this paper studies the Burmese speech waveform synthesis method, designs and implements a HMM-based Burmese speech synthesis baseline system, and based on this, introduces a deep neural network (DNN) to replace the decision tree model of HMM speech synthesis system, thereby improving the acoustic model to improve the quality of speech synthesis. The experimental results show that the baseline system is feasible, and the introduction of DNN speech synthesis system can effectively improve the quality of speech synthesis.

Keywords-speech synthesis; HMM; acoustic model; decision tree; deep neural network

I. INTRODUCTION

With the development of information technology, the research of speech synthesis technology in Chinese and English has been relatively mature. However, due to the lack of linguistic resources in Burmese, the research on speech synthesis is lagging behind. Burmese is the official language of Myanmar and has a history of more than a thousand years, spoken by about 54 million people^[1]. Similar to Chinese, Burmese has four tones.

The initial phase of the Burmese speech synthesis system is mainly focused on waveform concatenation techniques, such as Myanmar text-to-speech system with rule-based tone synthesis proposed by Kyawt in 2011^[2]; Diphone-Concatenation speech synthesis for Myanmar Language proposed by Soe in 2013^[3]. In 2017, Hlaing proposed using phoneme concatenation method Myanmar speech synthesis^[4], which is the speech synthesis through waveform concatenation. This waveform concatenation technology based on large-scale corpus is extremely costly and cannot be effectively put into practice application. In 2015, Ye first proposed a Myanmar statistical parameters speech synthesis method based on HMM, but there are still many pronunciation errors in the synthesized speech^[5]. The HMM-based Burmese speech synthesis still needs to be further studied. In recent years, speech synthesis based on HMM-DNN and end-to-end speech synthesis based on DNN have gradually become the mainstream technology for developing application systems. Due to the end-to-end speech synthesis based on DNN requires high training

sample size and computational ability^[6], this method is not involved in this paper.

In order to develop the Burmese speech synthesis application system, this paper studies the Burmese speech synthesis method, designs and implements a HMM-based Burmese speech synthesis baseline system, completes Grapheme-to-Phoneme transcription, automatic phoneme segmentation, context attributes and question set design, speech synthesis and other work, and based on this, the DNN acoustic model is introduced to replace the decision tree model in the HMM speech synthesis system, which solves the limitations of some traditional acoustic models, thereby improving the quality of speech synthesis.

The content of this paper is organized as follows: The second part introduces the Burmese speech synthesis system based on HMM and DNN; the third part mainly introduces the design and implementation of speech synthesis system; the fourth part is the analysis of experimental results; the last part is a summary.

II. SPEECH SYNTHESIS SYSTEM BASED ON HMM-DNN

A. HMM-based Speech Synthesis System

The typical speech synthesis system has two main parts, front-end text analysis and back-end speech synthesis. The front-end text analysis is mainly to get text, and performing text analysis to obtain the back-end training information. Text analysis mainly including the normalization, word segmentation and Grapheme-to-Phoneme(G2P), etc. The back-end speech synthesis is based on the information obtained by the front-end, through the training of the data model, the parameters are predicted, and then conduct speech synthesis. This paper mainly focuses on the back-end part of the speech synthesis system.

The framework of statistical parameter speech synthesis system based on HMM is shown as Fig. 1^[7], the complete HMM speech synthesis system can be divided into two parts: training and synthesis. The training part mainly extracts acoustic feature parameters from the corpus, and then performs HMM modeling, and the model clustering and training are conducted based on labels information, context attributes and question set. The synthesis part is to perform text analysis of the text that need to be synthesized, predict the parameters according to the model of the training part, and then perform speech synthesis through the speech synthesizer.

In HMM-based systems, HMM modeling is required for each context attribute, but limited data cannot completely cover all context combinations, so clustering

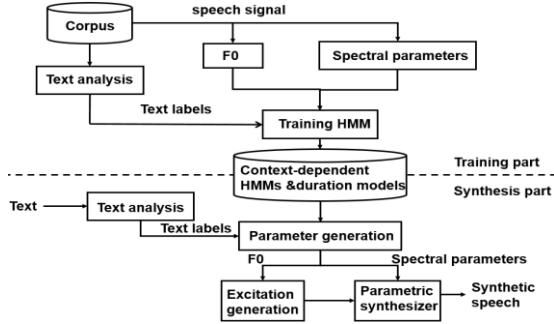


Figure 1. Speech synthesis system framework.

training based on decision tree is used in HMM speech synthesis systems. Although context clustering based on decision trees enables HMM speech synthesis systems to synthesize better quality speech, there are still some limitations^[8]: First of all, decision tree clustering cannot solve complex context feature problems well. Solving this problem can also be represented by a large amount of data, but it violates the design purpose of small data training speech synthesis based on HMM system. Then there is the problem that the decision tree divides the training samples, each leaf node is independent of each other and cannot share parameters, the data of each leaf node will be very rare, and it cannot be used for other related models. This will make some rare linguistic features to be directly ignored, thus affecting the speech quality of speech synthesis system.

An effective way to solve the above problem is to replace the decision tree model with a DNN acoustic model. This method can not only solve complex context feature with high-dimensional data input^[8], but also train all data to get the weight of each feature, which can effectively solve the problem of poor generalization ability of data.

B. Speech Synthesis System Based on HMM-DNN

DNN is an artificial neural network with many hidden layers between the input and output layers. The DNN simulates human speech generation through a hierarchical structure, and transforms language text information into the final speech output^[9]. Due to the complexity of speech data, shallow model structures such as HMM have limited modeling capabilities and cannot capture high-order correlations between data features very well. The powerful modeling ability of DNN is more suitable for modeling complex speech data, and its deep structure can compactly represent large-span, highly complex features. At the same time, DNN considers context dependent speech data features, and can improve the accuracy of modeling by using high-dimensional feature vectors with high discrimination. Therefore, we use the DNN acoustic model to replace the original acoustic model structure, which can better improve the quality of synthesized speech.

The Burmese speech synthesis framework based on the DNN acoustic model is shown in Fig. 2^[8]. First, the input text is converted into a context labeling, and then the system converts into input sequence according to the context labeling. The input sequence is a binary sequence answered by the corresponding question set. The input sequence is trained and mapped by the network forward algorithm to obtain the output sequence. The output

sequence contains various acoustic features and their dynamic parameter^[10], where the weights are trained by the training data. The input and output are frame-to-frame

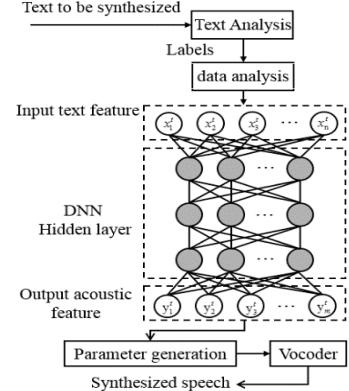


Figure 2. Speech synthesis framework based on DNN acoustic model.

forced alignment by the trained HMM model and then put into the network for training. Through the training of the network, the best matching model of input and output is obtained. Like the HMM, the DNN acoustic model also generates speech parameters. By setting the predicted output feature from the DNN as mean vector and setting the pre-calculated output feature variance from all training data as covariance matrix, the speech parameter generation algorithm can generate the acoustic parameters, and then uses the generated parameters for speech synthesis.

It can be seen that the text analysis, speech parameter generation and waveform synthesis modules of the DNN-based system can be shared with the HMM-based system, and the DNN-based system only needs to replace the part from the context labeling to the decision tree clustering.

III. DESIGN AND IMPLEMENTATION

A. Grapheme-to-Phoneme Conversion

Burmese is a complex language, and its writing is very special. In the process of speech synthesis, in order for the computer to correctly read the Burmese language, Burmese text needs to be translated into Roman text^[11]. The G2P transcription is to translate the Burmese text into a Roman alphabet text according to their pronunciation in Burmese. The transcription of this paper is based on the syllable structure of Burmese. The Burmese syllables can be divided into initials and finals. The initials are composed of consonants and complex consonants, the complex consonants are composed of consonants and medial. The finals are composed of vowels.

There are stacked words and pronunciation change in Burmese, after consulting Burmese experts and reading a large number of documents, we summarized the rules of the stacked words and pronunciation change. According to these rules, the corresponding program is designed to deal with these phenomena to improve the accuracy of the transcription.

The transcription scheme used in this paper is mainly designed with reference to the international phonetic of the Burmese alphabet and some transcription schemes proposed by Burmese scholars. The transcription process is carried out in conjunction with a Burmese dictionary (26600 words) published by the Myanmar Language Association.

The whole transcription process is as follows: The sentence is divided into words. Look for the word in the dictionary, if there is, use the transcription in the dictionary directly. If there is no such word in the dictionary, then the transcription will be performed according to the rules. First, the stacked words are processed according to the rules, then the words are divided into syllables, and the syllables are further parsed into consonant, vowel and medial, and then judged whether they belong to four kinds of pronunciation change. If so, the syllables are changed according to the rules. Then, according to the transcription scheme, the consonant, vowel and medial are transcribed separately, then combined them according to the spelling rules. The generated Roman alphabet text is the G2P transcription text. The transcription correctness rate of the 4000 sentences of Burmese text used in this paper reached 93.6%, which met the conditions for developing the Burmese speech synthesis system. This is a G2P transcription example:

Burmese text: ရာသိတေ (weather)

G2P transcription text: ja3dhi3u1du1

B. Selection of Synthesis Unit

In the process of speech synthesis, it is important to choose the appropriate synthesis unit. Through reading a large number of documents and consulting Burmese experts, Burmese syllable structure is similar to Chinese. Therefore, referring to the structure of the Chinese syllable structure, the Burmese syllables are divided into initials and finals, and the initials and finals are chosen as the synthesis unit of Burmese. Burmese is a tonal language, there are 50 finals and 66 initials in total.

After the synthesis unit is determined, the prosody text can be generated. The prosody text includes all the work of the front-end text analysis, which is very important for the back-end speech synthesis. The prosody text contains prosody information such as word boundaries, syllable boundaries, and phoneme boundaries.

C. Automatic Phoneme Segmentation

The automatic segmentation of the phoneme is a process of dividing the initialized monophone label file into more accurate label files with phoneme duration information. The automatic segmentation of phonemes can be divided into three stages:

The first stage: mainly preparing training data and training models. This paper selects 4000 Burmese sentences, including audio and text. Initialized monophone file is generated combine audio and prosody text. Acoustic feature parameters are extracted from the audio using the HTK tool. Then, according to the initial HMM model of each phoneme, the global mean and variance are obtained by HCompV^[7]. HInit reads in all the initialization training data, and the mean and variance are re-estimated. Finally, HRest re-estimates the parameters of HInit estimation.

The second stage: HHED modifies the HMM model corresponding to each phoneme and the sentence according to the parameters estimated in the first stage, and then the model is re-estimated as training set by HERest. After a total of 5 times training, the second stage of parameter revaluation and model building was completed.

The third stage: using HVite to force alignment to get the segmentation label file. Finally, a monophone labeling

file with accurate duration information can be obtained.

D. Context Attributes and Question Set Design

In continuous speech, each phoneme interacts with each other and there is a common phenomenon of coarticulation. The context attribute set is the coarticulation attribute labeling set of each central phoneme and the front and back phonemes. In the process of speech synthesis, context dependent model training is performed in the training stage to obtain the optimal parameters. In the synthesis stage, the state prediction is carried out through the model and the context attribute label of the input text. Therefore, the context attribute label is important. The label data include the positions of speech units, tones, prosody information, etc. This paper designs a context attributes set based on the language characteristics and pronunciation style of Burmese language, realize the process of automatically converting the Burmese text into the corresponding label file.

Because the number of the phoneme model in the training data is limited, and the number of context dependent models is large, there will be overfitting problems in the training process. In addition, in the synthesis stage, the context attributes of the text that need to be synthesized have not appeared in the training data, the synthesis cannot be completed. In order to solve these problems, the system uses the decision tree clustering algorithm to carry out model clustering, and combines the question set to perform the contextual model clustering training^[12]. Decision tree cluster training based on question in the question set. So, the question sets are designed to combine the pronunciation features of Burmese.

E. Training and Synthesis

1) *HMM speech synthesis system*: In the study of this paper, HMM speech synthesis main modeling parameters are configured as follows: Mel-cepstrum parameters, fundamental frequency parameters as acoustic parameters; initials and finals as the basic modeling unit; 5-state HMM model with no-skip structure from left-to-right is adopted.

The overall design of the Burmese speech synthesis system based on HMM is divided into three stages:

In the first stage, the HTK tool is used to extract the speech acoustic parameters, including the spectral parameters and the fundamental frequency parameters. Then, the monophone list, the monophone time label, the context attribute label and question sets are stored in the specified folder of the system.

In the second stage, the acoustic model training is mainly carried out, and the system mainly calls the HTK toolkit to complete the training. The decision tree model clustering of Burmese speech synthesis system is based on context attributes and question sets, and the acoustic parameters and triphone context dependent HMM model are obtained.

The third stage is to synthesis. First, text analysis is performed on the Burmese sentence that needs to be synthesized, and the text analysis result is converted into a corresponding context dependent labels, then putting them into the synthesis system. Finally, according to the parameters model, the corresponding acoustic parameters are generated by the parameter generation algorithm, and the speech is synthesized by the parameter synthesizer.

2) *HMM-DNN speech synthesis system*: Based on the HMM-DNN Burmese speech synthesis system, the DNN acoustic model is used to replace the traditional decision tree model to train the acoustic model. According to the features of the input and output sequence setting, and the study of the DNN acoustic model, the DNN training network parameters configuration in this paper is shown in Table I.

TABLE I. DNN ACOUSTIC MODEL NETWORK SETTINGS FOR BURMESE SPEECH SYNTHESIS SYSTEM

DNN acoustic model network parameters	Parameter value (function)
Batch size	256
Hidden layer activation function	Sigmoid
Learning rate	0.001
Number of hidden layers	3
Number of hidden layer units	2048
Number of input layer units	483
Number of output layer units	109
Optimizer	Adam

In the DNN acoustic model training, the composition of the input and output data is related to the quality of the entire acoustic model. The input sequence is a 483-bit vector sequence transformed from the Burmese question set and the context attribute set. The DNN training model is a frame-to-frame correspondence, each sequence represents the corresponding text feature of a frame^[13]. Mainly the various feature information is extracted from the context attribute labeling corresponding to the frame, including phonemes and part of speech and other information. The sequence value includes two types. The first type is the binary value of answering questions in question set, for example, is the current phoneme 'k'? The second type is a specific value, such as the position of the phoneme in the current syllable.

Then the input sequence is mapped to the output sequence through the forward algorithm training. The output sequence of the training data is a 109-bit vector sequence, which mainly includes spectral parameters, fundamental frequency features, and corresponding one-order and second-order differences.

In the synthesis stage of the DNN acoustic model, the corresponding context label is generated by the text to be synthesized, and then converted into a binary sequence, the best parameter sequence is predicted by trained DNN model. The predicted output feature sequence is treated as a mean vector, and the variance of the output features constitutes a covariance matrix, thereby generating parameters, and then using the generated parameters for speech synthesis.

IV. EXPERIMENTAL

A. Data Preparation

In this paper, the audio is recorded by a professional female broadcaster whose native language is Burmese. The Burmese pronunciation corpus used in this experiment is a total of 4000 sentences, the duration is about 6.6 hours, the total size of wav audio files is about 2 GB, the Burmese

text has a corpus size of 628 KB, the audio sample rate is 48 KHz.

B. Phoneme Segmentation Experimental Results and Analysis

According to the knowledge of the spectrogram, the vowel has a clear first and second formant. The red line in the below figure is the vowel formant. It can be seen that the vowel position is basically correct, and occasionally there is a deviation in the position of the consonant, and the result of the phoneme segmentation is generally correct. We randomly selected 100 sentences and obtained the correct rate of segmentation is 83.6%, which basically meets the requirements of the speech synthesis system.

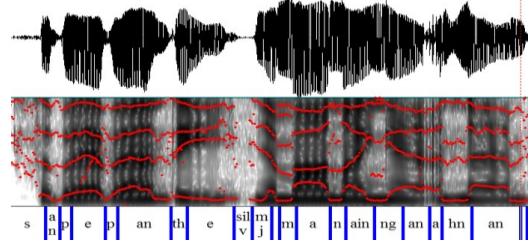


Figure 3. Phoneme segmentation result.

C. Speech Synthesis Experiment Results and Evaluation

1) *Objective evaluation*: The following is a waveform of a sentence (Burmese: ဗုပ်ယုန်းသည် မြန်မာနိုင်ငှာနဲ့ အပြေားတွင် ပေါက်ရောက်သည်). Translation: Osmanthus fragrans is grown and opened in many places in Myanmar), the first is the original speech waveform, the second is the synthetic speech waveform based on HMM, and the third is the synthetic speech waveform based on HMM-DNN. It can be seen that although the shape of the original waveform is basically restored, the waveform of based on HMM-DNN is closer to the original speech waveform.



Figure 4. Speech synthesis comparison of speech waveforms.

The following is the spectrogram, the first is the synthetic speech spectrogram based on HMM, the second is the synthetic speech spectrogram based on HMM-DNN, it can be seen that the phoneme boundary of the synthetic speech based on HMM-DNN is clearer, the transition between phonemes is more natural.

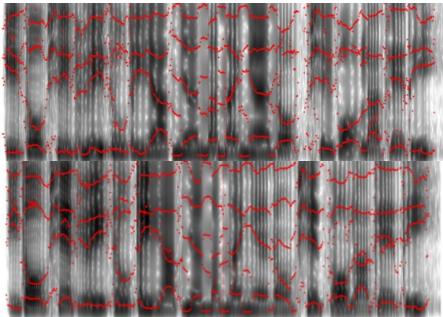


Figure 5. Speech synthesis comparison of spectrograms.

2) *Subjective evaluation:* The speech synthesis system is a key part of human-computer communication. The quality and naturalness of synthetic speech, that is, the subjective feelings that people hear, are critical. Five students studying speech synthesis are invited to make a comparative analysis of the synthesized and original speech (20 sentences), and subjectively describe the naturalness perception of different speeches according to the naturalness level shown in the table below:

TABLE II. NATURALNESS LEVEL

Level	Speech naturalness
5	Very natural
4	More natural
3	Acceptable
2	Less natural
1	Unacceptable

In the naturalness test, the synthesized speech and the original speech are subjectively described, and the subjective description results are statistically analyzed. The results are shown in the Fig 6:

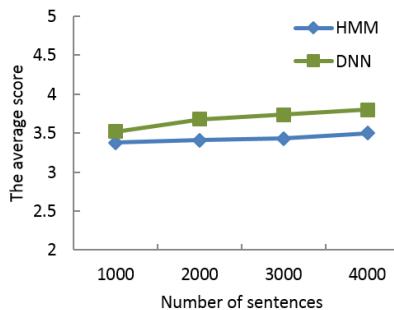


Figure 6. Evaluation results.

From the above subjective evaluation, we can see that the Burmese speech synthesis baseline system based on HMM has certain feasibility. The score of the speech synthesis based on HMM increases with the increase of the data amount, but the amplitude is not large. The score of the speech synthesis based on HMM-DNN increases more with the increase of the data amount. And the overall score is higher than that of HMM-based speech synthesis. It can be said that the introduction of DNN speech synthesis system effectively improves the quality of speech synthesis.

V. CONCLUSION

This paper focuses on the design and implementation of the Burmese speech synthesis baseline system based on

HMM, and completes the Grapheme-to-Phoneme transcription, automatic phoneme segmentation, context attributes and question set design, and the speech synthesis. Based on this, the DNN acoustic model is introduced to replace the decision tree clustering model in HMM speech synthesis, which solves some limitations of traditional decision tree clustering and improves the quality of speech synthesis. The experimental results show that the quality of synthesized speech based on HMM has reached a general level, and the quality of speech synthesis introduced into the DNN acoustic model is improved on the basis of HMM. However, the synthesized speech is still lacking in rhythm. Later, we can consider adding more rhythm information to make the synthesized speech more natural. In addition, it is also possible to consider further improve the acoustic model by using a more advanced network structure to better improve the quality of speech synthesis.

ACKNOWLEDGMENT

This work is supported by the Natural Science Foundation of China (No.61961043).

REFERENCES

- [1] Wang, D, "Burmese Language Tutorial (No.1)," Beijing University Publishing House, 2012. (in Chinese).
- [2] Kyawt Y, and Tomio T, "Myanmar text-to-speech system with rule-based tone synthesis," Acoustical Science and Technology, vol. 32, no. 5, 2011, pp. 174–181.
- [3] Ei.P, Aye T, "Diphone-Concatenation speech synthesis for Myanmar Language," International Journal of Scientific, Engineering and Technology Research, vol. 2, No. 4, 2013, pp. 1078–1087.
- [4] Hlaing C, Thida A, "Myanmar speech synthesis system by using phoneme concatenation method," International Conference on Signal Processing and Communication. 2017.
- [5] Ye K, Win P, Jinfu N, Yoshinori S, Andrew F, Chiori H, Hisashi K, Eiichiro S, "HMM Based Myanmar Text to Speech System," INTERSPEECH 2015.
- [6] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, et al, "Tacotron: Towards end-to-end speech synthesis," 2017.
- [7] Tokuda K, Nankaku Y, Toda T, et al, "Speech Synthesis Based on Hidden Markov Models," Proceedings of the IEEE, 2013, 101(5):1234-1252.
- [8] Zen H, Senior A, "Schuster M. Statistical parametric speech synthesis using deep neural networks," IEEE International Conference on Acoustics. IEEE, 2013.
- [9] Zhen-Hua Ling, Shi-yin Kang, Heiga Zen, et al, "Deep Learning for Acoustic Modeling in Parametric Speech Generation: A systematic review of existing techniques and future trends," J. IEEE Signal Processing Magazine, 2015, 32(3):35-52.
- [10] Qian Y, Fan Y, Hu W, et al, "On the training aspects of Deep Neural Network (DNN) for parametric TTS synthesis," 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014:3829-3833.
- [11] Thu Y, Ye K, Win P, Jinfu N, "Syllable Pronunciation Features for Myanmar Grapheme to Phoneme Conversion," the 13th International Conference on Computer Applications. 2015:161-167.
- [12] Tokuda K, Zen H, Black A W, "An HMM-based speech synthesis system applied to English". IEEE Workshop on Speech Synthesis. 2013.
- [13] Hashimoto K, Oura K, Nankaku Y, et al, "The effect of neural networks in statistical parametric speech synthesis," 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015:4455-4459.

Exploring Letter's Differences between Partial Indonesian Branch Language and English

Nankai Lin, Sihui Fu, Jiawen Huang and Shengyi Jiang[✉]

School of Information Science and Technology

Guangdong University of Foreign Studies

Guangzhou, China

neakail@outlook.com,jiangshengyi@163.com

Abstract—Differences of letter usage are the most basic differences between different languages, which can reflect the most essential diversity. Many linguists study the letter differences between common languages, but seldom research those between non-common languages. This paper selects three representative languages from the Indonesian branch of the Austronesian language family, namely Malay, Indonesian and Filipino. To study the letter differences between these three languages and English, we concentrate on word length distribution, letter frequency distribution, commonly used letter pairs, commonly used letter trigrams, and ranked letter frequency distribution. The results show that great differences do exist between three Indonesian-branch languages and English, and the differences between Malay and Indonesian are the smallest.

Keywords-Differences of letter usage; Austronesian family language; word length distribution; ranked letter frequency distribution

I. INTRODUCTION

Although morphemes, not letters, are usually considered to be the smallest linguistic unit, studying statistics of letter usage has its own merit [1]. For example, information on letter frequency is essential in cryptography for deciphering a substitution code [2]. Moreover, in Morse code, the more commonly used letters, the shorter the coding symbol. There are also some similar methods in data compression techniques, such as Huffman coding, in which a source letter is coded based on the probability of its occurrence. Therefore, it is of great significance to carry out letter usage research.

There are many differences in letter usage between different languages. The study of them promotes the related research in cryptology, natural language processing, linguistics and so on. However, relevant research at this stage mainly focuses on common languages. As to non-common languages such as Malay and Indonesian, the focus is on lexical differences. Lin et al. studied the differences in the word frequency distribution between Indonesian and English using Zipf's Law [3]. They also explored lexical differences between Indonesian and Malay [4].

Indonesian, Malay and Filipino are all members of the Indonesian branch of the Austronesian (Malayo-Polynesian) language family¹ and therefore could somehow reveal the characteristics of this language branch. In this paper, we choose these three languages to conduct a preliminary survey of the letter differences between the Indonesian language branch and English. On

¹ <https://www.britannica.com/topic/Malay-language>

TABLE I. THE DATA DISTRIBUTION OF OUR CRAWLED DATA

Language	Website	Number of words
Filipino	Bandera ²	18521278
Indonesian	Antara News ³	183210621
Malay	Bharian ⁴	31795086
English	Guardian ⁵	255040052

the other hand, all these four languages adopt the Latin alphabet as their writing systems, which makes the comparison much easier. We target the news texts in the above three languages and English as the subjects of the research. Four aspects of letter differences among these four languages are studied, including word length, letter frequency distribution, common letter pairs and common letter trigrams. What's more, we fit the ranked letter frequency distributions with ten letter frequency distribution models.

The remaining part of this paper is organized as follows: Section 2 provides information of our data; Section 3 demonstrates how we analyze the difference of word length; Section 4 introduces the letter frequency distribution analysis; Section 5 discusses distributions of the letter pair frequency and the letter trigrams frequency; Section 6 demonstrates how we fit the ranked letter frequency distributions; Section 7 concludes our work.

II. DATA DESCRIPTION

To obtain news texts, we crawl four influential news websites to represent these four languages' news texts and segment the articles into sentences and then words. The data distribution concerning these websites is shown in Table 1 (note that the punctuations have been excluded).

III. WORD LENGTH FREQUENCY DISTRIBUTION

The literature on word-length frequency distributions is in abundance within quantitative linguistics. Most research nevertheless mainly focus on common languages such as English, German, etc. [5][6][7][8][9]. Word length frequency typically investigates the frequency of words of different lengths in syllables [10]. We count the length frequency to get the frequency distribution of different languages. Table 2 presents the relative frequency distribution of word length for four languages. We could see that compared with the other three languages, Filipino is more likely to use shorter words.

² <http://bandera.inquirer.net/>

³ <http://www.antaranews.com/>

⁴ <http://www.bharian.com.my/>

⁵ <http://www.theguardian.com/uk>

We also calculate the average word length for each language. The average word lengths for English and Filipino are close, 4.69 and 4.77 respectively, while the ones for Malay and Indonesian are relatively longer, which are 6.07 and 6.11. Although Indonesian, Malay and Filipino all belong to the Indonesian language branch, the average word length of Filipino is shorter than Malay and Indonesian's. This is because many words in Indonesian and Malay are formed by adding affixes.

IV. LETTER FREQUENCY DISTRIBUTION ANALYSIS

Studies of letter statistics for many languages have been carried out, such as English, German, Spanish, Esperanto, Russian and Malay [11][12][13][14][15][16][17]. Most studies simply direct towards the distribution of a single language, but rarely compare it with those of other languages. We separately calculate the letter frequency distributions of the four languages in question. Detailed results are shown in Table 3. Note that although C, F, J, Q, V, X and Z are not indigenous Filipino letters, with the introduction of foreign words, these letters gradually appear in Filipino, so we also suppose Filipino has 26 letters, and we could see these letters have lower frequency. The five most commonly used letters in Filipino are A, N, I, G and S. The five most commonly used letters in Indonesian and Malay are the same, which are A, N, E, I and R. As for English, the experimental results we have are consistent with those of other previous studies, which are E, T, A, O and I. In addition, we also count the first, last and middle letter frequency distribution of these four languages. The results are presented in Fig. 2-4. Each chart suggests that the frequency distribution curve of Indonesian is identical to that of Malay. N is the most commonly used first letter of Filipino, with A the middle letter and G the last letter. Of English, they are T, E and E.

TABLE II. WORD LENGTH FREQUENCY DISTRIBUTION

Length	Language			
	Filipino	Indonesian	Malay	English
1	0.0187	0.0040	0.0020	0.0490
2	0.2277	0.0461	0.0427	0.1656
3	0.1426	0.0801	0.0895	0.1918
4	0.1560	0.1708	0.1592	0.1626
5	0.1205	0.1750	0.1752	0.1106
6	0.0977	0.1356	0.1372	0.0881
7	0.0773	0.1182	0.1337	0.0787
8	0.0600	0.0908	0.1024	0.0544
9	0.0380	0.0716	0.0623	0.0406
10	0.0263	0.0495	0.0428	0.0268
11	0.0153	0.0284	0.0284	0.0149
12	0.0085	0.0137	0.0140	0.0081
13	0.0056	0.0084	0.0055	0.0049
14	0.0027	0.0030	0.0023	0.0020
15	0.0015	0.0020	0.0008	0.0010
16	0.0007	0.0008	0.0005	0.0005
17	0.0005	0.0005	0.0003	0.0003
18	0.0002	0.0006	0.0003	0.0001
19	0.0001	0.0002	0.0001	0.0001
20	0.0001	0.0002	0.0001	0.0001

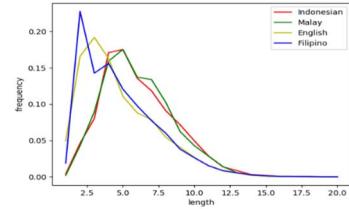


Figure 1. Word length frequency distribution.

TABLE III. LETTER FREQUENCY OF EACH LANGUAGE

Letter	Language			
	Filipino	Indonesian	Malay	English
A	0.1887	0.1908	0.2013	0.0835
B	0.0191	0.0247	0.0282	0.0167
C	0.0141	0.0058	0.0054	0.0310
D	0.0235	0.0393	0.0398	0.0377
E	0.0537	0.0815	0.0840	0.1212
F	0.0072	0.0032	0.0031	0.0217
G	0.0637	0.0361	0.0345	0.0216
H	0.0230	0.0208	0.0240	0.0494
I	0.0805	0.0770	0.0754	0.0741
J	0.0023	0.0108	0.0106	0.0019
K	0.0269	0.0485	0.0505	0.0080
L	0.0402	0.0317	0.0349	0.0423
M	0.0340	0.0430	0.0479	0.0252
N	0.1194	0.1006	0.0981	0.0712
O	0.0545	0.0224	0.0157	0.0745
P	0.0281	0.0340	0.0326	0.0208
Q	0.0008	0.0002	0.0002	0.0009
R	0.0381	0.0562	0.0515	0.0622
S	0.0585	0.0463	0.0420	0.0678
T	0.0526	0.0545	0.0487	0.0900
U	0.0263	0.0480	0.0487	0.0280
V	0.0050	0.0016	0.0013	0.0108
W	0.0112	0.0064	0.0049	0.0190
X	0.0009	0.0003	0.0002	0.0020
Y	0.0262	0.0154	0.0146	0.0179
Z	0.0014	0.0007	0.0017	0.0008

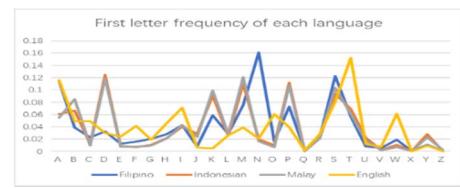


Figure 2. Frequencies of the first letter in each language.

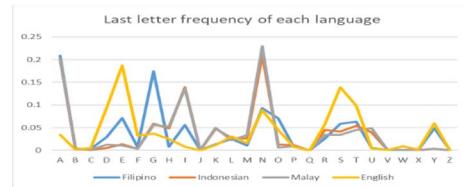


Figure 3. Frequencies of the last letter in each language.

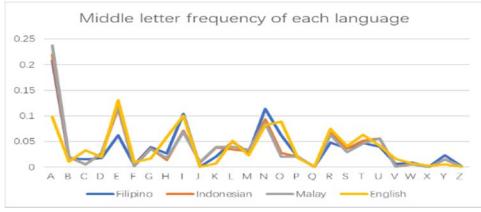


Figure 4. Frequencies of the middle letter in each language.

TABLE IV. TWENTY MOST COMMONLY USED LETTER PAIRS IN EACH LANGUAGE

Filipino		Indonesian		Malay		English	
Pair	Freq	Pair	Freq	Pair	Freq	Pair	Freq
NG	0.0513	AN	0.0603	AN	0.0644	TH	0.0326
AN	0.0500	NG	0.0275	ER	0.0274	HE	0.0281
NA	0.0316	KA	0.0231	NG	0.0267	IN	0.0256
IN	0.0263	ER	0.0229	KA	0.0228	ER	0.0200
SA	0.0217	EN	0.0226	DA	0.0220	AN	0.0194
MA	0.0186	TA	0.0205	EN	0.0211	RE	0.0178
LA	0.0182	AR	0.0203	LA	0.0192	ON	0.0158
KA	0.0160	DA	0.0194	TA	0.0177	EN	0.0136
AL	0.0156	ME	0.0174	ME	0.0169	AT	0.0135
AT	0.0151	LA	0.0161	AT	0.0162	ND	0.0127
PA	0.0147	RA	0.0161	RA	0.0160	OR	0.0123
ON	0.0145	AT	0.0157	AR	0.0150	ES	0.0121
AG	0.0141	DI	0.0150	MA	0.0147	NG	0.0119
GA	0.0125	YA	0.0145	PE	0.0142	IT	0.0117
SI	0.0122	GA	0.0144	DI	0.0140	TO	0.0115
NI	0.0120	AK	0.0139	YA	0.0135	AR	0.0114
AY	0.0116	IN	0.0138	IN	0.0132	ST	0.0113
YA	0.0110	PE	0.0133	GA	0.0130	ED	0.0111
TA	0.0109	SE	0.0126	SE	0.0128	IS	0.0111
ER	0.0107	AS	0.0113	AL	0.0128	TE	0.0110

TABLE V. TEN MOST COMMONLY USED LETTER TRIGRAMS IN EACH LANGUAGE

Filipino		Indonesian		Malay		English	
Pair	Freq	Pair	Freq	Pair	Freq	Pair	Freq
ANG	0.0349	ANG	0.0169	ANG	0.0171	THE	0.0287
ING	0.0097	KAN	0.0138	KAN	0.0139	ING	0.0132
ALA	0.0088	MEN	0.0126	MEN	0.0118	AND	0.0116
INA	0.0070	ARA	0.0087	BER	0.0102	ION	0.0066
IYA	0.0068	ENG	0.0087	ALA	0.0091	ENT	0.0062
LAN	0.0069	NGA	0.0085	ENG	0.0082	FOR	0.0055
ONG	0.0064	NYA	0.0076	ATA	0.0080	HAT	0.0052
AMA	0.0060	ATA	0.0075	ADA	0.0078	THA	0.0052
ILA	0.0058	AKA	0.0072	DAN	0.0078	TIO	0.0050
MAN	0.0057	DAN	0.0072	NGA	0.0077	HER	0.0050
AKA	0.0051	TER	0.007	PER	0.0075	TER	0.0046
THE	0.0050	PER	0.0067	ARA	0.0069	VER	0.0039
YAN	0.0050	YAN	0.0064	TAN	0.0069	ERE	0.0038
ARA	0.0049	GAN	0.0062	NYA	0.0065	ATI	0.0038
AGA	0.0044	BER	0.0061	GAN	0.0064	ALL	0.0036
NAG	0.0044	ALA	0.0056	TER	0.0059	ERS	0.0036
AND	0.0042	ADA	0.0055	ERA	0.0058	HIS	0.0036
HIN	0.0041	ELA	0.0055	RAN	0.0055	ATE	0.0034
MGA	0.0040	TAN	0.0054	YAN	0.0055	ITH	0.0032
PAG	0.0040	NTA	0.0049	AKA	0.0052	WIT	0.0031

TABLE VI. RESULT OF THE JACCARD SIMILARITY COEFFICIENT CALCULATION

Language	Number of common letter pairs		Jaccard similarity coefficient	
	bigram	trigram	bigram	trigram
Filipino, Malay	12	5	0.42	0.14
Filipino, Indonesian	10	5	0.33	0.14
Filipino, English	5	3	0.14	0.08
Indonesian, Malay	18	18	0.82	0.82
Indonesian, English	7	1	0.21	0.03
Malay, English	7	1	0.21	0.03

A and N are the most commonly used middle and last letters in both Malay and Indonesian. M is the most common first letter in Malay and D in Indonesian.

V. LETTER PAIR DIFFERENCE ANALYSIS

Dahlqvist described some of the characteristics of words and single character distribution as well as the distribution of character bigrams and trigrams in the Uppsala Newspaper Corpus [18]. Jones and Mewhort studied case-sensitive letter and character bigrams frequency counts from large-scale English corpora [19]. In this work, we count all the letter pairs' frequency and present the twenty most commonly used letter pairs in each language. The results are shown in Table 4. And then we use the Jaccard similarity coefficient to calculate the similarity between any two languages. The Jaccard similarity coefficient calculation formula is as follows:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

where $|A \cup B|$ is the union of the twenty most commonly used letter pairs in two languages (i.e. the number of letter pairs that occur in any one of the languages), and $|A \cap B|$ is the intersection of the twenty most commonly used letter pairs in two languages (i.e. the number of letter pairs that both languages share). The result is shown in Table 6.

From the perspective of letter pairs, we could see that Malay and Indonesian have the highest similarity, followed by Filipino and Malay, and then by Filipino and Indonesian. It suggests that there is a certain commonality in the use of letter pairs between languages of the Indonesian branch.

We then use the same strategy to study the differences of letter trigrams among languages. Table 5 and Table 6 are the experimental results. For any two languages, except the Indonesian-Malay pair, there is a small overlap between their most frequently used letter trigrams. Most of these commonly used letter trigrams are typical affixes in these languages. For instance, ING is one of the English affixes, and KAN, MEN, BER, NYA etc. are common affixes in Indonesian and Malay. What's more, the frequencies of these letter trigrams in Indonesian approximates to those in Malay.

VI. FITTING RANKED LETTER FREQUENCY DISTRIBUTIONS

The limited range in its abscissa of ranked letter frequency distributions causes multiple functions to fit the observed distribution reasonably well. In order to critically compare various functions, Li applied the statistical model selections to some functions, using the texts of U.S. and Mexican presidential speeches in the last 1-2 centuries [1].

In our work, we use ten different functions to fit the ranked letter frequency distribution. The following is a list of these functions (y denotes the normalized letter frequency, and x the rank of the frequency; x = 1 for most frequent letter and x = 26 for the rarest letter, and n = 26 is the maximum rank value).

Gusein-Zade function is based on the study of Gusein et al. They studied the frequency distribution of letters for

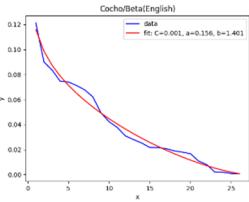


Figure 5. Fitting results for English data.

Russian and fitted this function to the distribution [16][20][21]. Gusein-Zade function is as follows:

$$y = C \log \frac{n+1}{x}$$

Power-Law is a simple but effective function to fit all kinds of distribution. Clauset presented a principled statistical framework for discerning and quantifying power-law behavior in empirical data [22]. Li used this function to fit their data's letter frequency distribution [1]. The function is as below:

$$y = \frac{C}{x^a}$$

Exponential is also a simple but effective function to fit all kinds of distribution. And it is the basis of other functions. Exponential formula is as follows:

$$y = C e^{-ax}$$

Logarithmic is an extension of the Gusein-Zade function by allowing the coefficient of $\log(x)$ term to be independently fitted [23][24]. The function is as below:

$$y = C - a \log(x)$$

Weibull is a statistical distribution function proposed by Weibull. It could be used in many situations [25]. It corresponds to the stretched exponential cumulative and many people apply it to fit letter frequency distribution [26][27]. Weibull distribution is as follows:

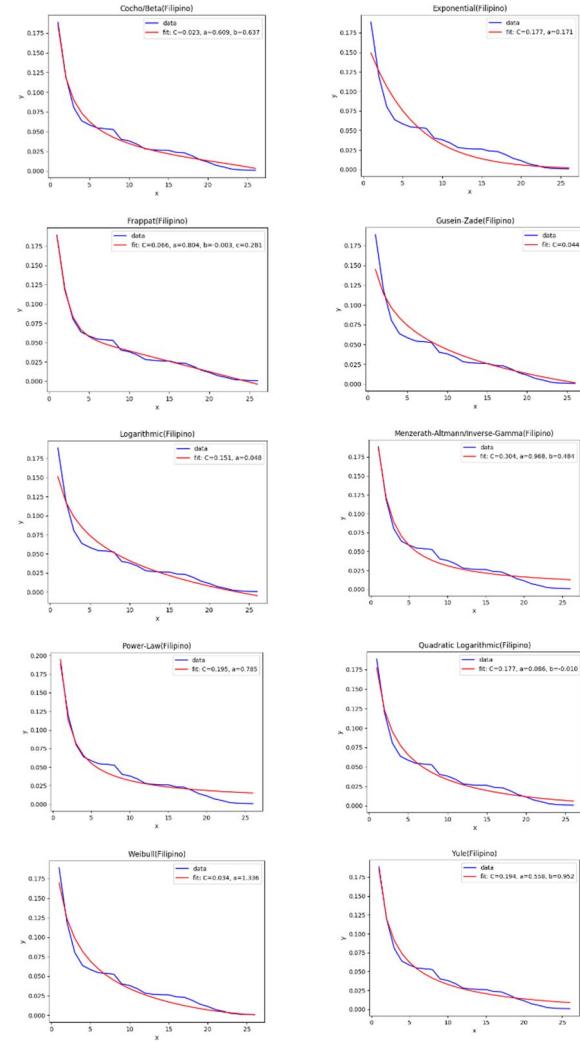


Figure 6. Fitting results for Filipino data.

$$y = C \left(\log \frac{n+1}{x} \right)^a$$

Quadratic Logarithmic is an extension of the logarithmic function by adding one extra term [23][24]. Quadratic Logarithmic formula is as follows:

$$y = C - a \log x - b (\log x)^2$$

Proposed by Frappat, Frappat adds a linear trend over the exponential function [34][35]. The function is as follows:

$$y = C + bx + ce^{-ax}$$

Proposed by Yule et al., Yule uses an exponential function. This function was put forward based on the conclusions of Dr. J. C. Willis, F.R.S. [27]. Li et al. used it to fit the letter frequency distribution and Martindale used it to fit graphemes and phonemes frequency distribution [28][29]. The function is as follows:

$$y = C \frac{b^x}{x^a}$$

Menzerath-Altmann/Inverse-Gamma, concerning the relationship between the length of two linguistic units, uses an exponential function of the inverse of rank [27][30]. Menzerath-Altmann/Inverse-Gamma is as below:

$$y = C \frac{e^{-\frac{b}{x}}}{x^a}$$

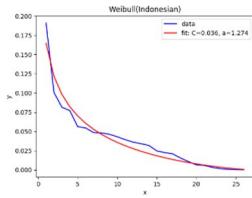
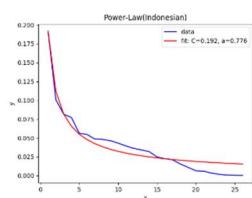
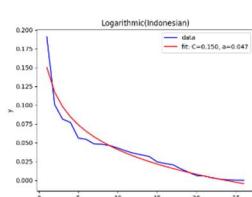
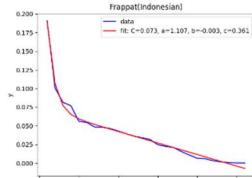
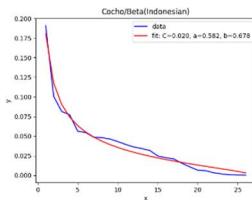


Figure 7. Fitting results for Indonesian data.

Cocho/Beta attempts to fit the two ends of a rank-frequency distribution by power-laws with different exponents [27][31][32][33]. It can be expressed as:

$$y = C \frac{(n+1-x)^b}{x^a}$$

We use these functions to fit data in four languages, and the results are shown in Figures 5-8. How well a function y fits the data can be measured by the sum of squared errors (residuals) SSE:

$$SSE = \sum_{i=1}^n (y_i - y'_i)^2$$

where the loss of each model is measured by above function. Table 8 lists the SSE of each function.

The best function for Filipino, Indonesian and Malay, selected by SSE, is the Frappat function. For English, the best functions are Cocho/Beta and Quadratic Logarithmic. And Cocho/Beta function is also the second best function for Filipino, Indonesian and Malay. The SSE values of four languages' best fitting functions are all less than 0.001. The results show that the fitting of ranked letter frequency distribution of Filipino is the best, with a SSE value of 0.002, using Frappat as the fitting function. We could see

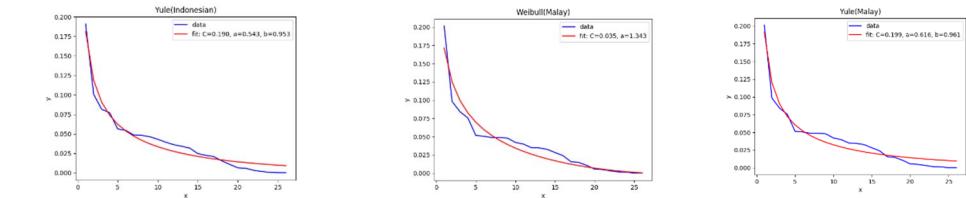
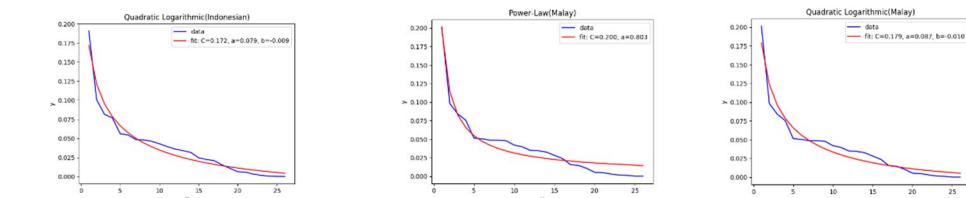
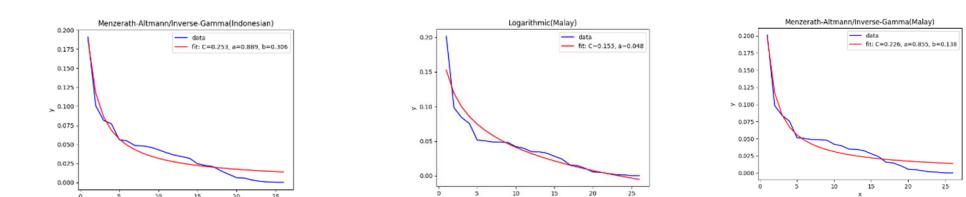
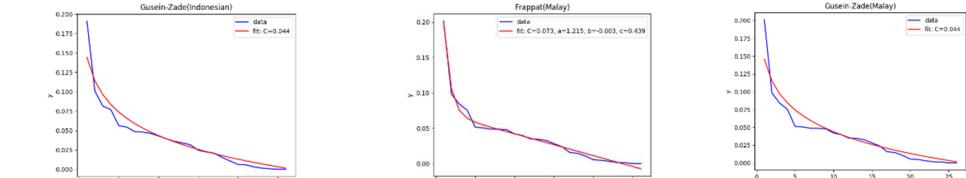
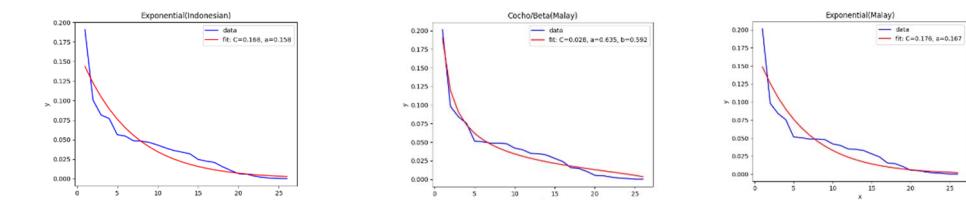


Figure 8. Fitting results for Malay data.

that the statistics for the three Indonesian branch languages are quite similar.

TABLE VII. THE SSE OF EACH FUNCTION

Function	Language			
	Filipino	Indonesian	Malay	English
Gusein-Zade	0.0033	0.0034	0.0048	0.0010
Power-Law	0.0018	0.0025	0.0027	0.0049
Exponential	0.0043	0.0049	0.0064	0.0008
Logarithmic	0.0029	0.0030	0.0043	0.0009
Cocho/Beta	0.0006	0.0011	0.0016	0.0005
Weibull	0.0018	0.0023	0.0032	0.0007
Quadratic Logarithmic	0.0011	0.0018	0.0025	0.0005
Yule	0.0010	0.0017	0.0021	0.0008
Menzerath-Altmann/Inverse-Gamma	0.0015	0.0024	0.0026	0.0026
Frappat	0.0002	0.0004	0.0005	0.0006

VII. CONCLUSION

In this paper, we study the differences of letter usage among four languages in terms of vocabulary length, letter frequency distribution, common letter pairs and common letter trigrams. What's more, we fit the ranked letter frequency distributions with ten letter frequency distribution models. Our results show that there are considerable differences between three Indonesian branch languages and English. In addition, although Filipino, Indonesian and Malay belong to the same language branch, the differences between Filipino and the other two languages could not be ignored. We could also see that Indonesian and Malay are indeed very similar, which further validates that they share the same origin.

In future, we will try to study the differences of letter usage among these languages from other perspectives. And we also consider processing other languages in the Indonesian branch, to verify the results present in this work.

ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China (No. 61572145) and the Major Projects of Guangdong Education Department for Foundation Research and Applied Research (No. 2017KZDXM031). The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

- [1] W. Li and P. Miramontes, "Fitting Ranked English and Spanish Letter Frequency Distribution in U.S. and Mexican Presidential Speeches," *J. Quant. Linguist. - JQL*, vol. 18, 2011.
- [2] W. F. Friedman, *Elements of Cryptanalysis*. Government Printing Office, Washington, 1976.
- [3] N. Lin, S. Fu, S. Jiang, C. Chen, L. Xiao, and G. Zhu, "Learning Indonesian Frequently Used Vocabulary from Large-Scale News," in 2018 International Conference on Asian Language Processing (IALP), 2018, pp. 234–239.
- [4] N. Lin, S. Fu, S. Jiang, G. Zhu, and Y. Hou, "Exploring Lexical Differences Between Indonesian and Malay," in 2018 International Conference on Asian Language Processing (IALP), 2018, pp. 178–183.
- [5] G. A. Miller, E. B. Newman, and E. A. Friedman, "Length-frequency statistics for written English," *Inf. Control*, vol. 1, no. 4, pp. 370–389, 1958.
- [6] F. Golcher, "Wiederholungen in Texten," Humboldt-Universität zu Berlin, Philosophische Fakultät II, 2013.
- [7] R. H. Baayen, *Word Frequency Distributions*. Springer Netherlands, 2012.
- [8] K. Best, "Results and perspectives of the Göttingen project on quantitative linguistics," *J. Quant. Linguist.*, vol. 5, no. 3, pp. 155–162, 1998.
- [9] K.-H. Best, "Häufigkeitsverteilungen in Texten. By Simone Andersen," *Glottometrics*, vol. 2, 2002.
- [10] R. D. Smith, "Distinct word length frequencies: distributions and symbol entropies," *CoRR*, vol. abs/1207.2334, 2012.
- [11] D. R. Ridley and M. Lively, "English letter frequencies and their applications: Part I," *Percept. Mot. Skills*, vol. 96, pp. 545–548, 2003.
- [12] R. L. Solso and J. F. King, "Frequency and versatility of letters in the English language," *Behav. Res. Methods*, vol. 8, pp. 283–286, 1976.
- [13] A. Beutelspacher, *Kryptologie: eine Einführung in die Wissenschaft vom Verschlüsseln, Verbergen und Verheimlichen; ohne alle Geheimniskrämerie, aber nicht ohne hinterlistigen Schalk, dargestellt zum Nutzen und Ergötzen des allgemeinen Publikums*. Vieweg+Teubner Verlag, 2007.
- [14] F. Pratt, *Secret and Urgent: The Story of Codes and Ciphers*. Aegean Park Press, 1939.
- [15] D. G. Simpson, "La Oftecoj de la Esperantaj Literoj," 2007.
- [16] S. M. Gusein-Zade, "Frequency distribution of letters in the Russian language," vol. 24, pp. 338–342, 1989.
- [17] A. Shah, A. Z. Saidin, I. F. Taha, and A. M. Zeki, "Frequencies Determination of Characters for Bahasa Melayu: Results of Preliminary Investigations," *Procedia - Soc. Behav. Sci.*, vol. 27, pp. 233–240, 2011.
- [18] B. Dahlqvist, "The Distribution of Characters, Bi- and Trigrams in the Uppsala 70 Million Words Swedish Newspaper Corpus," 1999.
- [19] M. N. Jones and D. J. K. Mewhort, "Case-sensitive letter and bigram frequency counts from large-scale English corpora," *Behav. Res. Methods, Instruments, Comput.*, vol. 36, no. 3, pp. 388–396, 2004.
- [20] S. M. Gusein-Zade, "On the frequency of meeting of key words and on other ranked series," *Sci. Information, Ser. 2 Inf. Process. Syst.*, vol. 1, pp. 28–32, 1987.
- [21] M. Y. Borodovsky and S. M. Gusein-Zade, "A General Rule for Ranged Series of Codon Frequencies in Different Genomes," *J. Biomol. Struct. Dyn.*, vol. 6, no. 5, pp. 1001–1012, 1989.
- [22] A. Clauset, C. Shalizi, and M. Newman, "Power-Law Distributions in Empirical Data," *SIAM Rev.*, vol. 51, no. 4, pp. 661–703, 2009.
- [23] I. Kanter and D. A. Kessler, "Markov Processes: Linguistics and Zipf's Law," *Phys. Rev. Lett.*, vol. 74, no. 22, pp. 4559–4562, 1995.
- [24] A. Vlad, A. Mitrea, and M. Mitrea, "Two frequency-rank law for letters in printed Romanian," *Proces. del Leng. Nat.*, vol. 26, pp. 153–160, 2000.
- [25] W. Weibull, "A Statistical Distribution Function of Wide Applicability," *ASME Journal of Applied Mechanics*, Vol. 18, pp. 293–297, 1951.
- [26] T. Nabeshima and Y.-P. Gunji, "Zipf's law in phonograms and Weibull distribution in ideograms: Comparison of English with Japanese," *Biosystems*, vol. 73, no. 2, pp. 131–139, 2004.
- [27] W. Li, P. Miramontes, and G. Cocho, "Fitting ranked linguistic data with two-parameter functions," *Entropy*, vol. 12, no. 7, pp. 1743–1764, Jul. 2010.
- [28] F. Y. E. and G. U. Yule, "A Mathematical Theory of Evolution Based on the Conclusions of Dr. J. C. Willis, F.R.S.," *J. R. Stat. Soc.*, 2006.
- [29] C. Martindale, S. M. Gusein-Zade, D. McKenzie, and M. Y. Borodovsky, "Comparison of equations describing the ranked frequency distributions of graphemes and phonemes," *J. Quant. Linguist.*, 1996.
- [30] G. Altmann, "Prolegomena to Menzerath's law," *Glottometrika*, vol. 2, pp. 1–10, 1980.
- [31] R. Mansilla, E. Köppen, G. Cocho, and P. Miramontes, "On the behavior of journal impact factor rank-order distribution," *J. Informetr.*, 2007.
- [32] G. G. Naumis and G. Cocho, "Tail universalities in rank distributions as an algebraic problem: The beta-like function," *Phys. A Stat. Mech. its Appl.*, 2008.
- [33] G. Martínez-Mekler, R. A. Martínez, M. B. del Río, R. Mansilla, P. Miramontes, and G. Cocho, "Universality of rank-ordering distributions in the arts and sciences," *PLoS One*, 2009.
- [34] L. Frappat, C. Minichini, A. Sciarrino, and P. Sorba, "Universality and Shannon entropy of codon usage," *Phys. Rev. E - Stat. Physics, Plasmas, Fluids, Relat. Interdiscip. Top.*, 2003.
- [35] L. Frappat and A. Sciarrino, "Conspiracy in bacterial genomes," *Phys. A Stat. Mech. its Appl.*, 2006.

CIEA: A Corpus for Chinese Implicit Emotion Analysis

Dawei Li, Jin Wang and Xuejie Zhang
*School of Information Science and Engineering
Yunnan University
Kunming, P.R. China
Contact: xjzhang@ynu.edu.cn*

Abstract—The traditional cultural euphemism of the Han nationality has profound ideological roots. China has always advocated Confucianism, which has led to the implicit expression of Chinese people's emotions. There are almost no obvious emotional words in spoken language, which poses a challenge to Chinese sentiment analysis. It is very interesting to exploit a corpus that does not contain emotional words, but instead uses detailed description in text to determine the category of the emotion expressed. In this study, we propose a corpus for Chinese implicit sentiment analysis. To do this, we have crawled millions of microblogs. After data cleaning and processing, we obtained the corpus. Based on this corpus, we introduced conventional models and neural networks for implicit sentiment analysis, and achieve promising results. A comparative experiment with a well-known corpus showed the importance of implicit emotions to emotional classification. This not only shows the usefulness of the proposed corpus for implicit sentiment analysis research, but also provides a baseline for further research on this topic.

Keywords-Chinese corpus; implicit emotion; masking key-words; baseline;

I. INTRODUCTION

Nowadays, individuals or organizations can share and send information on social networks anytime and anywhere. As an example, on the popular Chinese microblogging website Sina Weibo, users can share information, which mainly includes the things in their daily lives and comments on hot news on the site. These messages can be positive, neutral, negative, or one of six basic emotions (e.g., anger, happiness, fear, sadness, disgust and surprise) [1] in these microblog contents. However, Chinese people often express their emotions implicitly and do not directly express happiness or sadness. This leads to no obvious emotional words in the text, and it is necessary to judge the expressed emotions through context. When there are no obvious emotional words in the text, the analysis of emotions will be challenging.

Whether they are used in a conventional model or a neural network, the models built on previous corpora rely on emotional vocabulary (or corresponding representations) and do not focus on the causes of emotions or events. In previous corpus, emotion words appears in the text, which causes the model to pay more attention to emotional words. The model will not predict the correct emotions when there are no obvious emotional words in the text. When emotions are inferred, the type of emotion is often related to the context described in the text [2]. For this purpose, it is necessary to set up an emotional

Table I
THE EXAMPLE OF CORPUS.

ID	Text	Label
1	提交完论文的我[#关键词#]得不是一点点。 (I am so [#keyword#] that I submitted the paper.)	开心 (happy)
2	我好[#关键词#]！爸爸突然买了条边牧，说好的金毛和拉布拉多没有了。 (I am so [#keyword#] that dad bought a border collie. We negotiated to buy a golden lab.)	生气 (angry)
3	有点[#关键词#]，今晚我要开着灯睡觉 (I am so [#keyword#] that i have to sleep with a lights tonight.)	害怕 (fear)
4	刚喝完酒回来寝室就遭贼了，[#关键词#]，你们有什么办法对付这种事吗？ (I am so [#keyword#] that I was stolen after drinking the wine and returning to the bedroom. Do you have any way to deal with this kind of thing?)	伤心 (sad)
5	招商银行这是疯了么，一天发那么多条短信给我，已经有点[#关键词#]的感觉了！ (This is crazy for China Merchants Bank. It is [#keyword#] to send me so many text messages a day.)	厌恶 (disgusting)
6	一张老照片虽然不值钱，但它却保留了你人生中最难忘的一幕。当你若干年后，再看到后说不定会[#关键词#]地说：“咦，这就是我吗？” (An old photo is worthless, but it retains the most memorable scene of your life. When you see it for a few years, you may be [#keyword#] to say: "Hey, is this me?".)	惊讶 (surprised)

classification system without emotional words, and the emotion depends on the context depicted.

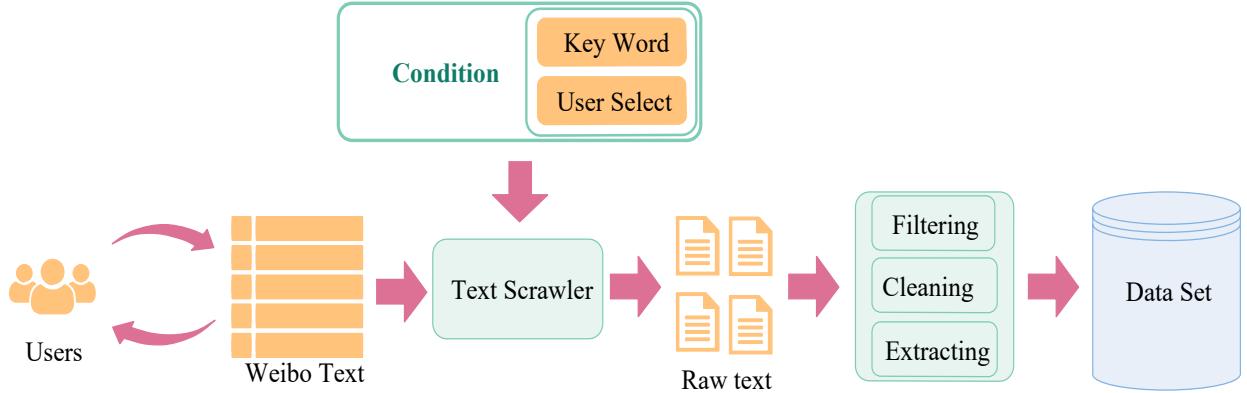


Figure 1. Diagram of the process for creating the corpus.

With the corpus, we aim to solve the problems mentioned above. We chose to build a microblog corpus, because Chinese microblogs are convenient to use, and easy to access. The data on microblogs show diversity and richness. The corpus contains a large amount of self-labeling annotation of the microblog data. We extract the microblogs (including emojis) with a certain number of words and clean the data. Then, we mask the "emotional word" and use context information to predict the masked word. The words masked are as follows: 生气, 厌恶, 害怕, 开心, 伤心, 惊讶 (angry, disgusting, fear, happy, sad, and surprised).

As shown in **Table I**, based on context analysis of Example 1, when an important thing is done, people will be happy. In Example 2, one thing was determined after by consultation, but not done according to the plan. This will make people angry or sad. In the example, the matter of selecting a dog is already discussed, but different dog was bought than that discussed. Therefore, the emotional word of the case should be angry. In example 3, the hidden emotional word should be fear which can be inferred from "i have to sleep with a lights tonight". There are no obvious emotional words in the example, but the masked emotions can be inferred by analyzing context information.

We used several well-known affective analysis corpora in the experiment. Delete the emotional words in the well-known corpus with obvious emotional trends, and build a comparative corpus. Using conventional models and deep learning models to model the corpus. The experimental results show that the performance of model is significantly reduced after deleting the emotional words with obvious emotional trends. This shows that emotional words have an impact on whether the model can predict the classification correctly. At the same time, the corpus we proposed did not appear to drop significantly.

We introduce a Chinese implicit emotional corpus. To our knowledge, it is the first such corpus (masking emotion words) to date. We split the corpus into a standard training set, experimental sets, and test sets to facilitate baseline for related methods. We use a neural network to evaluate the corpus and achieve promising results, which can be used as a baseline for the task. Finally, we compare the

different corpora to verify the impact of emotional words on the prediction results.

The rest of this paper is organized as follows. Section 2 introduces existing corpora in different fields. Section 3 describes the process of building the Chinese implicit emotional corpus and its properties. Section 4 analyzes the quality of the corpus through experiments and compares it with famous corpora. We conclude this work in Section 5.

II. RELATED WORK

There are many works on exploiting corpora to promote the development of NLP. Previous sentiment analysis involved multiple text types, including product [3], movie reviews [4], fairy tales [5], news stories, social commentary and commentary articles [6], blogs [7] [8] and microblogs [9]. Some well-known emotional resources are as follows.

The subjunctive mood in fairy tales is one of the first corpora to annotate the expression of emotions. Annotated a corpus of approximately 185 children stories including Grimm's, H.C. Andersen's and B.Potter's stories. The goal is to classify the emotional affinity of sentences in the narrative domain of children's fairy tales, for subsequent usage in appropriate expressive rendering of text-to-speech synthesis [5]. Then emotions were annotated according to the headlines of news. The aim is to predict emotions from news headlines. [10]. Movie reviews (MR) [11], Large Movie Review Dataset (IMDB) [12] and the Stanford sentiment tree-bank (SST) [13] were used. Use movie reviews to do a polar analysis of emotions, and achieve success. Twitter is one of the most popular sites, and in a variety of topics, it contains various emotions. Therefore, the researchers' enthusiasm for evaluating Twitter is very high. This has been researched by creating a corpus from Twitter posts using emotion-word hashtags [14], tracking the public sentiment of 2012 US presidential election on the polarization of the candidates [15], and modeling the Twitter text to determine whether the author is in favor of, against, or neutral to something [16]. These work on twitter corpus is to collect twitter prediction first. Then manually check the self-annotated text to ensure the correctness of the self-annotated text. These studies play

Table II
THE BASIC PROPERTIES OF THE CORPUS. TOKEN REPRESENTS THE AVERAGE NUMBER OF WORDS FOR EACH MICROBLOG.

Label \ Token	Train	Dev	Test
伤心(sadness)	65.9	64.39	63.79
开心(happiness)	59.03	60.68	59.64
厌恶(disgust)	64.54	63.63	65.41
惊讶(surprise)	75.88	77.5	75.3
生气(anger)	65.97	64.72	64.74
害怕(fear)	66.59	64.94	67.54

an important role in promoting the emotional analysis of online texts.

Wassa-2018 Task uses a tweet with a distinct emotional expression and then hides the emotional words [2]. Then use the tweet text which was masked emotional word to predict the masked emotional word belongs to. Our work is related to this work, using a large number of self-labeled Chinese Microblogs which emotional word was masked to predict the implicit emotion.

III. CHINESE IMPLICIT EMOTION RESOURCE CONSTRUCTION

We chose to use web crawler technology to obtain the corpus of Sina-Microblogs for its diversity and richness. We use six keywords: "生气", "厌恶", "害怕", "开心", "伤心" and "惊讶" as the condition to obtain a large amount self-labeling data. Anyone can create an account (individual, media or organization) on Sina-Microblogs and the content and form of each microblog are different. In order to guarantee the quality of the content, we only crawl personal original microblog data. In this way, the data is closer to daily life and more realistic. The process of the data collection is shown in **Figure 1** and summarized as follows:

(1) Use the key words: "生气", "厌恶", "害怕", "开心", "伤心" and "惊讶" as a condition to crawl personal original microblog text to ensure that the text contains clear emotional words.

(2) Extract the data which the length of sentence is between 15 and 200 as the original data. Delete negative data such as abusive content, advertising and pornography.

(3) Discard sentences which contains synonyms and antonyms related to keywords. Ensure only one type of word with a clear emotional tendency in each sentences. Then delete the repeated sentences.

(4) Ten volunteers were invited to manually repeat the steps 3.

(5) Mask emotional words (label) in the data. Replace the emotional word with "[# 关键词#]".

After data cleaning and processing, the corpus we proposed does not contain emotional words with obvious emotional tendencies. The corpus consists of 285000 pieces of microblogs data, including 156000 training posts, 39000

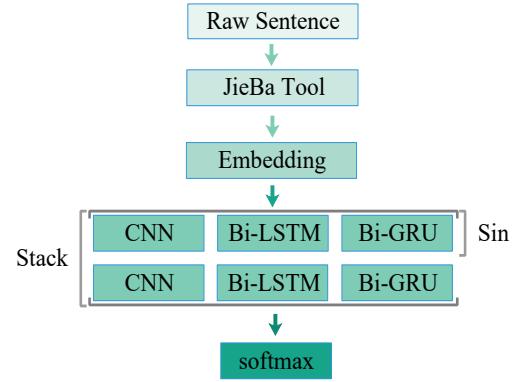


Figure 2. Stacked and non-stacked neural network models. Sin means that the neural network has only one layer, and stack means that there are two layers of neural networks.

Table III
THE RESULT OF BASELINE SYSTEM. SIN MEANS SINGLE LAYER. THE NUMBER REPRESENTS MACAVG- F_1 (%).

Model	MacAvg- F_1 (%)			
	Test		Dev	
	Sin	Stack	Sin	Stack
Logistic	59.12	-	58.5	-
Random Forest	52.51	-	51.4	-
CNN	55.73	59.65	55.39	59.14
Bi-LSTM	72.63	74.47	72.42	74.29
Bi-GRU	73.06	75.06	72.47	74.99

developments posts, and 90000 test posts. The three sub-collection are equally divided into six categories. In this way, the distribution of data is balanced. The basic corpus information is shown in the **Table II**. The table is analyzed to compare the average number of microblogs words in the same category for different subsets. It can be seen that the difference in the average number of microblogs for each class is very small. There are some differences in the average number of microblogs in different categories. This does not affect the training and prediction of the model.

IV. EXPERIMENT

In this section, we provided several baseline systems for evaluating our corpus. We set several baseline systems to test the basic performance of our corpus and provide meaningful comparisons for different systems and corpora.

A. Evaluation Script

In order to facilitate the evaluation of classification, we provide an evaluation script that measures the precision, recall, and F1 score for each emotional classification. We also added the F_1 (%) score of micro-average and macro-average. Because the amount of data in each category in our corpus is almost average, we use the macro-average F_1 (%) scores as the evaluation criteria.

B. Baseline System

In this study, we provide five baseline systems, including two conventional ones and three neural network

Table IV

THE RESULT OF CONTRAST EXPERIMENT. **MAX** REPRESENTS THE MAXIMUM NUMBER OF SENTENCES CONTAINING A SENTIMENT WORD; **AVERAGE** REPRESENTS THE AVERAGE NUMBER OF SENTIMENT WORDS IN EACH SENTENCE; **RATE** REPRESENTS THE PROPORTION OF SENTENCES IN THE CORPUS CONTAINING EMOTIONAL WORDS; **w/ EMOTION** REPRESENTS THE TEST SET ACCURACY RATE CONTAINING EMOTIONAL WORDS, AND **w/o EMOTION** REPRESENTS THE TEST SET ACCURACY RATE WITHOUT EMOTIONAL WORDS.

Corpus	Category	Max	Average	Rate(%)	Acc(%)	
					w/ Emotion	w/o Emotion
SST-2	2	20	1.71	36.68	83.47	73.09
IMDB	2	209	34.3	99.98	89.3	80.69
MR	2	22	2.06	41.25	77.92	67.84
CIEA-b	2	51	1.65	68.8	87.48	84.51

models. In the data processing phase, we only processed the text, and no special processing was used for emotions. We used the Jieba [17] word segmentation tool to segment Chinese sentence and retain the stop words. The goal is to enable a baseline system to accurately reflect the quality of the corpus. For the neural networks, Tencent AI Lab Embedding Corpus [18] is used as input to the word embedding layer, and provide a baseline for stacked and non-stacked models. Architecture shown in Figure 2.

- Conventional model: We used logistic regression [19] and the random forest [20] model and initialize a random forest classifier with 100 trees. Both models used the bag-of-words as features.
- Neural networks: We used the deep learnings models of the convolutional neural network (CNN) [21], and bidirectional long short-term Memory (LSTM) and gated recurrent unit (GRU) base on LSTM [22] and GRU [23]. Stacking and non-stacking were used in the neural network model. Stacked consists of two layers of deep learning model components, while non-stacked consists of one layer of deep learning components. We initialized the batch size to 1024 and the dropout to 0.25. For CNN, we initialized the kernel size to 3 and filter size to 60, and used the Soft-Max function to output predictive probability values. For LSTM or GRU, we initialized hidden units to 120 and the recurrent dropout to 0.25. Finally, the Soft-Max function was used to output Predictive probability value.

We used the conventional models and the neural network to evaluate the corpus. The results are showed in **Table III**.The performance of the neural network model is better than conventional model, while the stacked sequence model has a significant improvement over the non-stacked model. The results of the stacked Bidirectional GRU network is the best. For neural networks, the results of each model of the development set and the test set differ between 0.07% and 0.56%. The predicted results of development set are close to the test set prediction results. There is no significant difference in text characteristics between the Two data sets. It's means that the corpus we propose has good generalization capabilities, and follow the principle of independently identically distribution.

C. Corpus Comparison

In order to verify the impact of emotional words on the classification results, we designed the experiment. In the experiment we used the text of the movie review, which is the IMDB, MR and Stanford sentiment tree-bank-2 (SST-2) corpus. Simultaneously, extract data from our corpus labeled "开心" (happy) and "伤心" (sad). This binary classification corpus (CIEA-b) without strong emotional words is compared with the three corpora above. We use opinion lexicon [24] to process English text and ¹HowNet to process the binary corpus from our corpus. We filter the sentiment dictionary to obtain a secondary sentiment dictionary with obvious or strong emotions. We mask the emotional words which represent strong emotions in the text and generate a corpus that dose not contain emotional words. For the corpus with emotional words and the corpus of hidden emotional words, a stacked bidirectional GRU deep learning model is used to do a comparative experiment. In the case that the corpus is different and the rest of the conditions are identical. The model parameters corresponding to different datasets in the experiment are optimal. The result is shown in Table **IV**.

As indicated in **Table IV**, it is clear that the results of the corpus (SST, IMDB, MR) that doesn't contain emotional words compared to the original corpus have declined a lot. The corpus of the three binary classifications decreased from 8.61% to 10.38%. This result shows that the emotional word with obvious emotional tendency have a great influence on the performance of the model. Comparing the binary datasets extracted from the corpus we propose, the accuracy rate dropped by 2.97% in the case of masking a large number of emotional words. It proves that our corpus contains very few obvious emotional trend words.

V. CONCLUSION

In this study, we propose a new Chinese implicit sentiment corpus for implicit sentiment analysis. The corpus includes a large number of self-labeling training set data, validation set data and test set data. In the baseline experiment, the baseline of the stacked model reached 75.06% (macro-average F_1 scores). In baseline

¹ 知网:<http://www.keenage.com/>

system, we just used the word segmentation tool and no adjustment parameters. This shows that the quality of the corpus we proposed is promising. We used comparative experiments to prove the effectiveness and necessity of our proposed corpus. At the same time, the corpus we proposed is the first Chinese implicit sentiment analysis corpus. It effectively fits the implicit expression of Chinese expression and predicts implicit emotions. The new corpus can be used for research purposes. We hope that the corpus we proposed will help researchers better study sentiment analysis of Chinese texts, and encourage more researchers to perform experiments on this data.

VI. ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No.61966038, No.61702443 and No.61762091.

REFERENCES

- [1] P. Ekman, “An argument for basic emotions,” *Cognition & emotion*, vol. 6, no. 3-4, pp. 169–200, 1992.
- [2] R. Klinger, O. De Clercq, S. M. Mohammad, and A. Balahur, “IEST: WASSA-2018 Implicit Emotions Shared Task,” vol. 1, 2018. [Online]. Available: <http://arxiv.org/abs/1809.01083>
- [3] P. D. Turney, “Thumbs up or thumbs down?” *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, vol. 12, no. 3, p. 417, 2001. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1073083.1073153>
- [4] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? Sentiment Classification using Machine Learning Techniques,” *Antike und Abendland*, vol. 57, no. July, pp. 151–168, 2002. [Online]. Available: <http://arxiv.org/abs/cs/0205070>
- [5] C. O. Alm, D. Roth, and R. Sproat, “Emotions from text: Machine learning for text-based emotion prediction,” *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT '05*, no. October, pp. 579–586, 2005.
- [6] J. Wiebe, T. Wilson, and C. Cardie, “Annotating expressions of opinions and emotions in language,” *Language Resources and Evaluation*, vol. 39, no. 2-3, pp. 165–210, 2005.
- [7] G. A. van Kan, Y. Rolland, M. Houles, S. Gillette-Guyonnet, M. Soto, and B. Vellas, “The assessment of frailty in older adults,” *Clinics in Geriatric Medicine*, vol. 26, no. 2, pp. 275–286, 2010.
- [8] S. Aman and S. Szpakowicz, “Identifying Expressions of Emotion in Text,” *Text, Speech and Dialogue*, pp. 196–205, 2007. [Online]. Available: http://link.springer.com/10.1007/978-3-540-74628-7_27
- [9] P. S. Dodds, K. D. Harris, I. M. Kloumann, C. A. Bliss, and C. M. Danforth, “Temporal patterns of happiness and information in a global social network: Hedonometrics and Twitter,” *PLoS ONE*, vol. 6, no. 12, 2011.
- [10] C. Strapparava and R. Mihalcea, “Semeval-2007 task 14: Affective Text,” in *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, 2007, pp. 70–74.
- [11] B. Pang and L. Lee, “Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales,” *Association for Computational Linguistics*, vol. Proceeding, no. 1, pp. 115—124, 2005. [Online]. Available: <http://aclweb.org/anthology/P05-1015>
- [12] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, “Learning Word Vectors for Sentiment Analysis,” *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT 2011)*, pp. 142–150, 2011. [Online]. Available: <http://www.aclweb.org/anthology/P11-1015%7D>
- [13] H. A. Schwartz, J. C. Eichstaedt, M. L. Kern, L. Dziurzynski, S. M. Ramones, M. Agrawal, A. Shah, M. Kosinski, D. Stillwell, M. E. Seligman, and L. H. Ungar, “Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank ,” *PLoS ONE*, vol. 8, no. 9, 2013.
- [14] S. Mohammad, “#Emotional Tweets,” *{*SEM 2012}: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation ({SemEval 2012})*, pp. 246–255, 2012. [Online]. Available: <http://www.aclweb.org/anthology/S12-1033>
- [15] S. M. Mohammad, X. Zhu, S. Kiritchenko, and J. Martin, “Sentiment, emotion, purpose, and style in electoral tweets,” *Information Processing and Management*, vol. 51, no. 4, pp. 480–499, 2015.
- [16] S. M. Mohammad, P. Sobhani, and S. Kiritchenko, “Stance and Sentiment in Tweets,” vol. 0, no. 0, 2016. [Online]. Available: <http://arxiv.org/abs/1605.01655>
- [17] J. Sun, “‘jieba’ chinese word segmentation tool,” 2012.
- [18] Y. Song, S. Shi, J. Li, and H. Zhang, “Directional skip-gram: Explicitly distinguishing left and right context for word embeddings,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 175–180. [Online]. Available: <https://www.aclweb.org/anthology/N18-2028>
- [19] R. E. Wright, “Logistic regression.” 1995.
- [20] A. Liaw, M. Wiener *et al.*, “Classification and regression by randomforest,” *R news*, vol. 2, no. 3, pp. 18–22, 2002.
- [21] Y. Kim, “Convolutional neural networks for sentence classification,” *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pp. 1746–1751, 2014.
- [22] S. J. Hochreiter Sepp, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

- [23] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation,” *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, 2014. [Online]. Available: <http://arxiv.org/abs/1406.1078>
- [24] M. Hu and B. Liu, “Mining and summarizing customer reviews,” *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '04*, no. November, p. 168, 2004. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1014052.1014073>

Learning Deep Matching-Aware Network for Text Recommendation using Clickthrough Data

Haonan Liu, Nankai Lin, Zitao Chen, Ke Li and Shengyi Jiang[✉]

School of Information Science and Technology
Guangdong University of Foreign Studies
Guangzhou, China

haonan.liu@gmail.com, jiangshengyi@163.com

Abstract—With the trend of information globalization, the volume of text information is exploding, which results in the information overload problem. Text recommendation system has shown to be a valuable tool to help users in such situations of information overload. In general, most researchers define text recommendation as a static problem, ignoring sequential information. In this paper, we propose a text recommendation framework with matching-aware interest extractor and dynamic interest extractor. We apply the Attention-based Long Short-Term Memory Network (LSTM) to model a user's dynamic interest. Besides, we model a user's static interest with the idea of semantic matching. We integrate dynamic interest and static interest of users' and decide whether to recommend a text. We also propose a reasonable method to construct a text recommendation dataset with clickthrough data from CCIR 2018 shared task Personal Recommendation. We test our model and other baseline models on the dataset. The experiment shows our model outperforms all the baseline models and a state-of-the-art model, and the F1-score of our model reaches 0.76.

Keywords-text recommendation; semantic matching; deep learning

I. INTRODUCTION

The Internet has brought convenience for the dissemination of text information. Text information has been widely spread through online platforms such as Weibo, news portals and knowledge communities. People's access to information has gradually changed from traditional paper media reading to electronic media reading. Obtaining text information becomes easier than before. However, excessive information can also cause user information overload issues. In this case, leveraging text recommendation system can alleviate the user's burden of filtering information. Researchers from industry and academia have long widely studied text recommendation task and proposed many methods. In this paper, we propose a reasonable text recommendation dataset construction method using clickthrough data as well as a content-based text recommendation framework.

Unlike traditional recommendation systems, text recommendation is primarily subject to three challenges. First, text information is time-sensitive. While the volume of texts increases, the timeliness of text decreases over time, which makes methods like collaborative filtering have poor performance on this task. Second, the user's interest will change over time. The text recommendation system should involve the sequential information of users' reading behavior. Third, the interest of users in reading texts is diversified. Users will read text information in multiple fields, and the text recommendation system needs to take

the interests of users in various aspects into account to make the best recommendations.

Current text recommendation methods divide into three categories: collaborative filtering methods, content-based methods, and hybrid methods. Collaborative filtering methods [1] depends on the texts read by all users, and learn the features of users relying on the "wisdom of the crowd" to make a recommendation, without using the specific content of the text. Such methods usually suffer from the cold-starting problem. Researchers also propose many content-based methods [2][3][4] due to the rich information in the text content. These methods make a recommendation based on the similarity between texts, but most of them ignore the sequential information of user behaviors. The hybrid method [5] takes the user's information and text information into account. However, most online text information platforms can be accessed anonymously nowadays. In this case, the hybrid method does not perform the text recommendation task well. Most of the researches regard text recommendation as a static problem, ignoring the sequential features of user behavior. Recently, Zhu [6] proposed a content-based news recommendation method Deep Attention Neural Network (DAN). In their work, they involve the user sequential features and achieved state-of-the-art. However, their work does not well preserve the user's static interest features. In the experiments that follow, our model will make a comparison to DAN.

Encountered the challenges in the field of text recommendation and inspired by previous research, we propose a method for constructing a text recommendation dataset that simulates real-world scenarios as well as propose a framework that involves user dynamic interest and user static interest, namely Deep Matching-Aware Network (DMN). We build a dataset by using the clickthrough data of Chinese famous knowledge platform Zhihu from CCIR2018 shared task Personal Recommendation. In our work, DMN uses a convolutional neural network to learn the semantic feature. We implement an Attention-based Long Short-Term Memory Network to learn user dynamic interest features, and the user static interest features are extracted through using methods inspired by semantic matching. With dynamic and static interest features, we predict the probability that a candidate text is recommended.

We use different models to conduct experiments on the dataset. The experimental results show that the proposed DMN model has better results than other models. The remaining part of this paper organizes as follows: Section 2 introduces related studies; Section 3 provides information on our data and describes our data construction method; Section 4 introduces the framework of DMN; Section 5 introduces training of our framework; Section 6 introduces our experiment and analyzes the results of the different

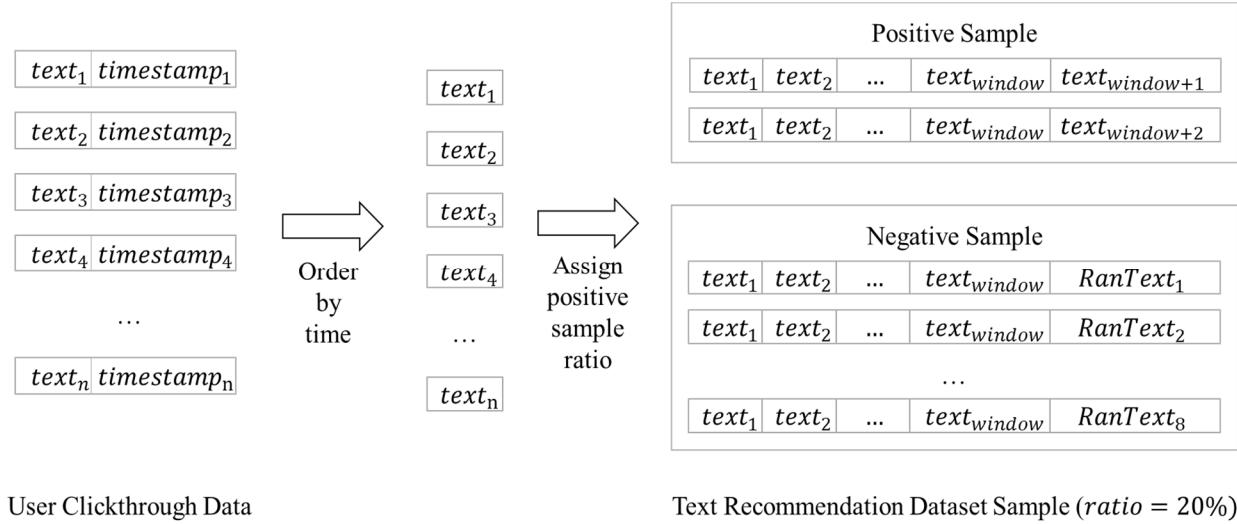


Figure 1. Simple visualization of the data construction method.

models; Section 7 is a case study and explores some variants' influence on our framework; Section 8 concludes our work.

II. RELATED WORK

A. Deep Recommendation System

Deep learning has been successfully applied in many fields, such as computer vision [7], speech recognition [8]. Among these applications, CNN and RNN are the two most popular deep learning methods. Recently, deep neural networks have representative progress in recommendation system. In general, there are two types of deep learning recommendation systems, one for learning the feature of users or items, and the other for simulating the interaction between users and items. For example, the DSSM [9] is used to learn the features of items, and Collaborative Deep Learning [10] learns the interaction features between users and items. Our model DMN learns the interaction features of users and items and also preserve the features of items.

B. Text Recommendation

Text recommendation tasks have long been widely studied, and the tasks are designed to provide users with highly relevant text or to learn user browsing preferences. Examples of text recommendations include recommend blogs [11], social media information [12], news [13], products (based on reviews) [14], and research papers [15]. Due to the rich information in text, many content-based text recommendation methods have been proposed, which are based on the similarity between texts. Zhu proposed a content-based framework DAN [6], using ARNN to capture sequence information and using the attention mechanism to capture the user's static interest feature. Different from DAN, we model the user's static interest based on the semantic similarity method and improve the performance.

C. Semantic Matching

Semantic matching tasks are one of the critical applications of many NLP applications, such as information retrieval, question and answer systems, text recommendations. With the development of deep learning, scholars have proposed several deep learning methods to obtain the semantic information and judge whether the sentence pairs match by comparing the similarity of semantic vectors. DSSM [9], ARC-II [16], DeepMatch [17] are some famous methods. DSSM projects the query and document into the vector space using a nonlinear method. Based on the semantic vector space, the model calculates the similarity between the query and the document. The ARC-II model uses a neural network model to extract information from sentence pairs interactions and to calculate the degree of matching. DeepMatch uses a topic model built with deep architecture to compare similarities between texts. Inspired by the semantic matching task, our work proposes the Matching-Aware Interest Extractor.

III. DATA DESCRIPTION

The data is from CCIR 2018 shared task Personal Recommendation of Zhihu¹. Zhihu is a famous Chinese knowledge platform, users share knowledge and discuss current affairs on the platform. The full dataset includes question information, answer information, user information, and user clickthrough data. We use the user clickthrough data and reconstruct it in our work.

We propose a method for constructing text recommendation dataset using clickthrough data which aims to simulate the user's behavior in a real-world situation. Figure 1 briefly introduces the process of our method. The method origins in Zhu's work [6] and we make some improvement. Given a user's reading history $\{history_1, history_2, \dots, history_j, \dots history_n\}$ ($1 \leq j \leq n$). Each $history_j$ represents the j_{th} record of a user and

¹ <https://biendata.com/competition/CCIR2018/>

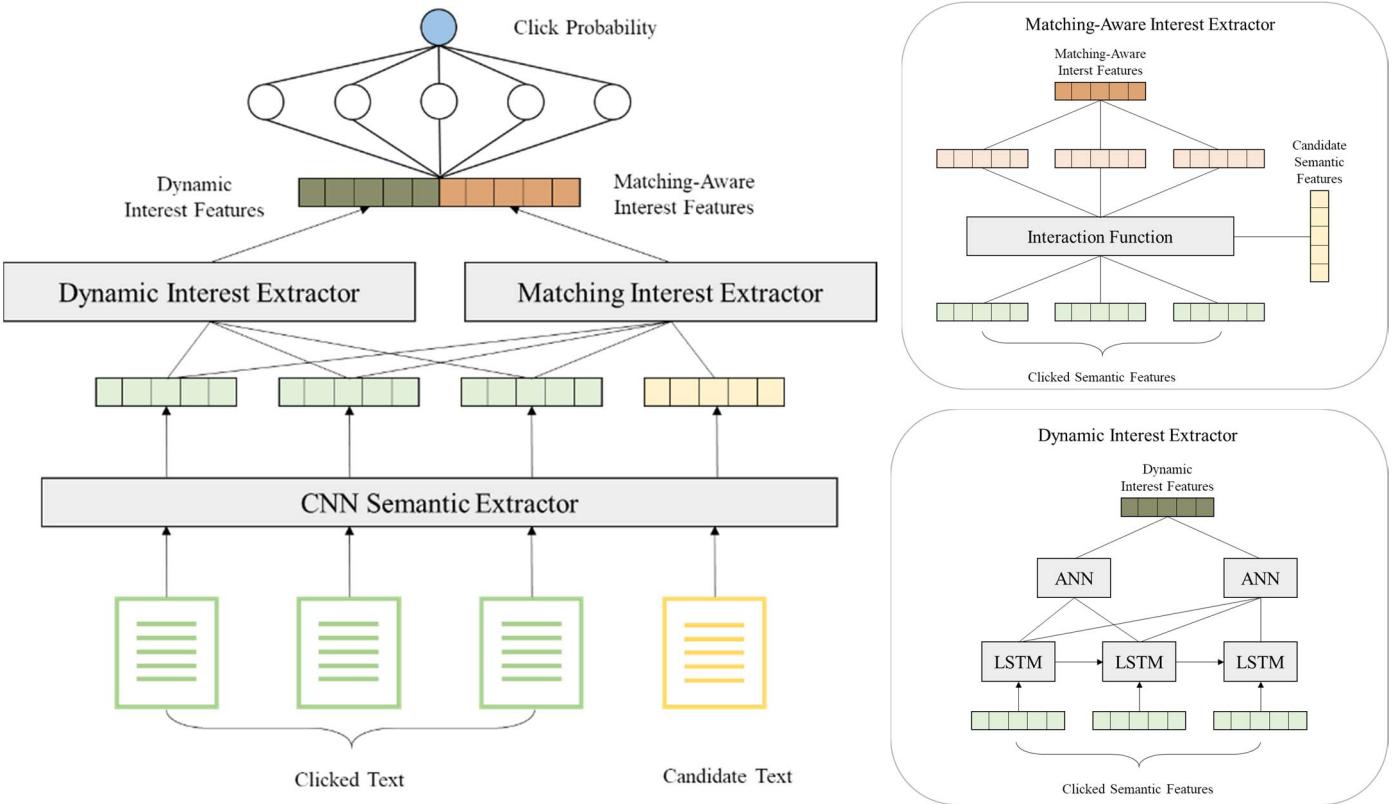


Figure 2. Simple visualization of DMN framework.

comprise of the text content and timestamp. We arrange the user history readings in ascending order of time. The earlier the reading time, the earlier the order and only preserve the text content $text_j$. In other words, the clickthrough data is transform to $\{text_1, text_2, text_3, \dots, text_n\}$. We set a variant $window$ to intercept the user history into small session $\{text_1, text_2, text_3, \dots, text_{window}\}$ ($window \in [1, n]$). Each session can represent a user of a range of time. Inspired by the idea of negative sampling in the Word2Vec [18], we propose a random sampling method to construct our data. We assign ten candidate texts for each session, and the positive sample ratio is randomly chosen from the list $[10\%, 20\%, 30\%]$. The positive samples are from the record after $window_{th}$ record and $label_c = 1$. The negative samples are randomly chosen from the text beyond the user's clickthrough data and $label_c = 0$.

We construct the dataset for text recommendation using the beforehand mention method. The training set, development set and test set has 10k, 1k, 1k samples respectively. Each line has three component: a session that represents a user of a range of time, a candidate text that the user may like, a label that indicates whether the candidate text should be recommended.

IV. DMN FRAMEWORK

The text recommendation problem described in this paper will satisfy the following definition: Give a user's text reading history, $History = \{text_1, text_2, text_3, \dots, text_n\}$, $text_j$ is the j th text read by the user, giving a candidate text $text_c$. Each $text_k$ includes word sequence $text_k = \{w_1, w_2, w_3, \dots, w_n\}$. User's text reading history and the

candidate text feed the model, and it will output the probability of recommending the candidate text.

In order to provide a user with suitable texts information, it is necessary to understand a user's long-term interest as well as short-term interest. Our framework adequately models user's interest with a perspective of static and dynamic. Figure 2 shows the structure of the DMN framework. In our model, for input session and candidate text, we use CNN to extract semantic features. Then we learn the user's static interest feature based on the semantic matching method and learn user's dynamic interest by using ARNN. Finally, we predict the probability of recommending the candidate text using a fully connected network.

A. CNN Semantic Extractor

There are many methods to represent text semantic information. Traditional unsupervised text semantic model like Bag Of Word [19] or TF-IDF [20] can not learn information of word order and suffer from data sparsity. Recently, convolution neural network (CNN) have representative progress in computer vision. Researchers proposed many text semantic model [21] based on convolution neural network. In our work, we use CNN to extract semantic information.

CNN is comprised of convolution layer and pooling layer. We denote the embedding input text matrix as $E = \{e_1, e_2, e_i \dots e_n\}$ ($1 \leq i \leq n$), e_i indicate i_{th} words embedding vector. We apply filter $k \in R^{d1 \times d2}$ with a stride $[s_1, s_2]$ to convolution the embedding matrix and receive feature vectors h , which is given by:

$$h = f(E \odot k + b) \quad (1)$$

The symbol \odot represents the convolution operation and b is the bias term, $f(\cdot)$ represents the activate function *Relu*. In pooling layer, we apply max pooling operation, we define the max-pooling function as $\text{out} = \text{maxpooling}(h, \text{filter}_i)$ and we use multiple filters to get multiple output out_i . We concatenate all the output and get semantic vector S of text embedding vectors E :

$$S = [\text{out}_1; \text{out}_2; \dots; \text{out}_i] \quad (2)$$

Symbol $[\cdot; \cdot]$ represents the concatenate of vectors.

B. Matching-Aware Interest Extractor

Matching-Aware Interest Extractor learns user static interest based on semantic matching methods. Each text reading by user interacts with the candidate text and get the interest features.

We get session semantic vectors list $S = [S_1, S_2, \dots, S_n]$ and candidate text semantic vector S_c from CNN semantic extractor. Semantic vector $S_i \in R^n$, n is the dimension of the semantic vector. Each vector from the session will interact with the candidate text semantic vector:

$$\text{Interaction}(S_i, S_c) \quad (3)$$

We experiment with two interaction functions.

Euclidean distance [22], used to calculate the true distance of two points in n-dimensional space. The Interaction function is defined as:

$$\text{Interaction}(S_i, S_c) = \sqrt{\sum_k^n (S_{ik} - S_{ck})^2} \quad (4)$$

Cosine similarity [23], is a common function to model interactions. The similarity score is viewed as the angle of two vectors. The Interaction function is defined as:

$$\text{Interaction}(S_i, S_c) = \frac{S_i^T S_c}{\|S_i\| \cdot \|S_c\|} \quad (6)$$

Where $\|\cdot\|$ stands for the L2 norm.

We concat all the interaction features together as user static interest features:

$$U_{\text{match}} = [\text{Interaction}(S_1, S_c); \text{Interaction}(S_2, S_c); \dots; \text{Interaction}(S_n, S_c)] \quad (7)$$

C. Dynamic Interest Extractor

In dynamic interest extractor, we use Attention-based LSTM to model user dynamic interest. For each step of the LSTM, we use attention mechanism to dynamically capture user sequential feature. This kind of Attention-based LSTM has applied in many tasks [24][25] and can better capture sequential information. We get session semantic vectors list $S = [S_1, S_2, \dots, S_n]$ from CNN semantic extractor. Semantic vector $S_i \in R^n$, n is the dimension of the semantic vector. We feed the vectors list into LSTM in sequence and get hidden output vector $[h_2, h_3, \dots, h_{n+1}]$:

$$h_i = \text{LSTM}(h_{i-1}, S_i) \quad (8)$$

For each hidden vector, we use attention mechanism to calculate user sequential features $f_{\text{atten}(i)}$ at this step:

$$f_{\text{atten}(i)} = \text{Attention}([h_1, \dots, h_{i-1}], h_i) \quad (9)$$

With hidden vector list $[h_1, \dots, h_{i-1}]$ and hidden vector h_i at i_{th} step, the attention mechanism is as following:

$$V_j = w_j h_i + b_j \quad (j = 1, 2, \dots, i-1) \quad (10)$$

$$V_i = w_i h_i + b_i \quad (11)$$

$$\alpha_{j,i} = \frac{\exp(w_v(V_j + V_i))}{\sum_{j=1}^{i-1} \exp(w_v(V_j + V_i))} \quad (12)$$

$$f_{\text{atten}(i)} = \sum_{j=1}^{i-1} \alpha_{j,i} h_j \quad (13)$$

Note that V_j represents the j_{th} text hidden vector ($j < i$), V_i represents the i_{th} text hidden vector and $f_{\text{atten}(i)}$ represent user sequential interest features at i_{th} step. We concatenate all the user sequential features together and apply CNN to extract the final user dynamic interest features:

$$U_{\text{seq}} = \text{cnn}([f_{\text{atten}(1)}; f_{\text{atten}(2)}; \dots; f_{\text{atten}(n)}]) \quad (14)$$

D. Probability Calculator

Finally, we have user sequential interest feature U_{seq} and user static interest feature U_{match} . We concatenate two features, feed it to the fully connected neural network and get the probability $\text{Prob}(\text{text}_c)$ of recommending the candidate text text_c .

V. TRAINING

We denote a training sample as $X = (\text{session}, \text{text}_c, \text{label}_c)$. session is made up of the click history $\{\text{text}_1, \text{text}_2, \dots, \text{text}_{\text{window}}\}$, text_c is the candidate text and label_c is the label. our model will output the probability of recommending $P_c \in [0, 1]$. We minimize the following likelihood function to train our model. $c \in \Delta^+$ indicates a positive sample set, and $c \in \Delta^-$ indicates a negative sample set:

$$\text{Loss} = -\left\{ \sum_{c \in \Delta^+} y \log(P_c) + \sum_{c \in \Delta^-} (1-y)(1-\log(P_c)) \right\} \quad (15)$$

In order to avoid over-fitting, we apply dropout and L2 regularization to the weight parameters of all the components.

VI. EXPERIMENT

A. Parameter Setting

We implement our model based on Tensorflow[26] and perform multiple experiments with different parameters for each model. We use F1, Precision, and Recall [27][28] as the evaluation metrics. Averaging the results of multiple

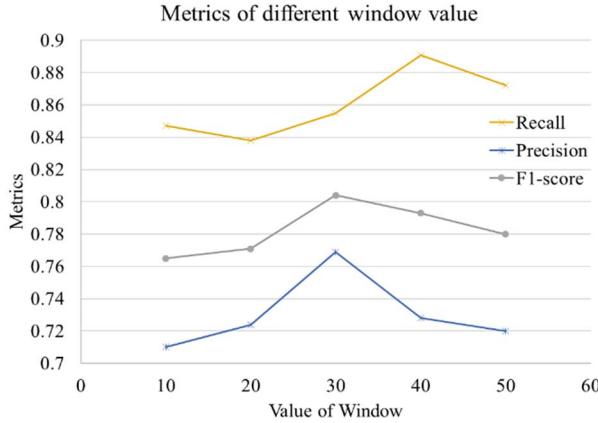


Figure 3. Comparison of different window value.

experiments to compare the performance of different models.

Baseline

We use the following state-of-the-art methods as baselines in our experiments.

- DSSM [9]: is a deep structured semantic model for document ranking. DSSM use a deep neural network to rank a set of documents for a given query. the user's clicked news is treated as the query and the candidate news are treated as the documents.
- DeepFM [29]: An end-to-end deep learning recommendation model that integrates the architectures of FM and deep neural networks. It models low-order feature interaction like FM and model high-order feature interactions like DNN. We concatenate both clicked texts and candidate text to feed into DeepFM.
- DAN [6]: A deep attention neural network for news recommendation. This model combines user sequential information and user interest feature by Attention and RNN.

Among the three models, DSSM is a model based on semantic matching, DeepFM combines traditional recommendation method FM and deep learning method DNN. DAN model is a recently proposed model combining sequential and static information.

B. Result and Analysis

In the experiment, we compare the DMN model with several benchmark models on our dataset. Table 1 shows the performance of different models.

TABLE I. COMPARISON AMONG DIFFERENT MODELS

Model	Evaluation Metrics		
	Precision	Recall	F1-score
DSSM	0.692	0.764	0.720
DeepFM	0.744	0.695	0.714
DAN	0.748	0.725	0.731
DMN	0.722	0.818	0.761

TABLE II. COMPARISON OF DIFFERENT INTERACTION METHOD

Model	Evaluation Metrics		
	F1-score	Precision	Recall
DMN with Eud	0.713	0.601	0.876
DMN with cos	0.761	0.722	0.818

In the experiment, DeepFM performed poorly compared with other models, F1-score only reached 0.714. F1-score of DSSM model reached 0.720, indicating that the semantic matching method performs well in this task. The DAN model performs the best in the benchmark model with the F1-score of 0.731. Our DMN model performs better than other benchmark models, with the F1-score of 0.761, which is about 0.03 higher than the DAN model. In terms of recall rate, the DMN model achieved the highest recall rate of 0.818, while in terms of accuracy, it was similar to the DeepFM and DAN models. In general, the DMN model can achieve better text recommendation performance compared to other models.

Compared with other benchmark models, DMN has the advantages of (1) using semantic matching to capture user's static interest. (2) Using Attention-based LSTM to learn the user's sequential feature, thereby obtaining user dynamic interest feature (3) We use CNN to extract the semantic features of the text, preserving the word order and semantic information. That is why our model can achieve a better result.

VII. CASE STUDY

A. Window Variants

We set the window variable to intercept the user text reading history to represent the user at a certain period. However, we need to consider the effect of the window variable. A small window value may not be able to represent a user. A long window value may not represent a user of a certain period. Thus we choose different values from the list {10,20,30,40,50,60} and conduct experiment. We explore how the window variable affects the recommendation performance. Figure 3 shows that all the metrics increase as the window value grows before window value reaches 30. The highest precision rate and F1-score appear when window value equals 30, while recall rate reaches the highest point with the window value of 40. In conclusion, the model performs the best when the window value equals 30.

B. Different Matching Method

Interaction function can learn the similarity of the user's reading text and candidate text. We experimented with two similarity calculation method and compared the performance. Table 2 shows the comparison of the different interaction method. According to the experimental results, using cosine similarity is higher in F1 value and accuracy than using Euclidean distance. In general, using cosine similarity as an interaction method achieve better recommendations performance.

VIII. CONCLUSION

In this paper, we propose the DMN framework for text recommendation tasks. Unlike other text recommendation models, DMN learns user dynamic interest features and user static interest features from the perspective of semantic matching. We also propose a reasonable data construction method for text recommendation dataset, which simulates the real-world situation, by using clickthrough data. In our experiment, the performance of the DMN model is better than other models, the F1-score reaches 0.76. We also explored how the window length variable affects the result and also experiment our model with different interaction functions in our framework.

In the future, our work will divide into two parts. On the one hand, we will further explore the possibility of applying semantic matching in text recommendation, and try to use a variety of semantic matching methods to learn the user's interest feature. On the other hand, we will consider introducing external knowledge and combine user or knowledge graph information to achieve better performance.

ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China (No. 61572145) and the Major Projects of Guangdong Education Department for Foundation Research and Applied Research (No. 2017KZDXM031). The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

- [1] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” Computer (Long Beach, Calif.), 2009.
- [2] R. J. Mooney and L. Roy, “Content-based book recommending using learning for text categorization,” in Proceedings of the fifth ACM conference on Digital libraries - DL '00, 2000.
- [3] M. Balabanović and Y. Shoham, “Fab: content-based, collaborative recommendation,” Commun. ACM, 1997.
- [4] D. Jannach et al., “Content-based recommendation,” in Recommender Systems, 2013.
- [5] V. Vekariya and G. R. Kulkarni, “Hybrid recommender systems: Content-boosted collaborative filtering for improved recommendations,” in Proceedings - International Conference on Communication Systems and Network Technologies, CSNT 2012. 2012.
- [6] Q. Zhu, X. Zhou, Z. Song , J. Tan, and L. Guo. “Deep Attention Neural Network for News Recommendation.” Aaaai, 2019.
- [7] A. Krizhevsky, I. Sutskever, and H. Geoffrey E., “Imagenet,” Adv. Neural Inf. Process. Syst. 25, 2012.
- [8] A. Graves, A. R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, 2013.
- [9] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, “Learning deep structured semantic models for web search using clickthrough data,” 2013.
- [10] H. Wang, N. Wang, and D.-Y. Yeung, “Collaborative Deep Learning for Recommender Systems,” in Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15, 2015.
- [11] T. Bansal, M. Das, and C. Bhattacharyya, “Content Driven User Profiling for Comment-Worthy Recommendations of News and Blog Articles,” 2015.
- [12] I. Guy, N. Zwerdling, I. Ronen, D. Carmel, and E. Uziel, “Social media recommendation based on people and tags,” 2010.
- [13] O. Phelan, K. McCarthy, and B. Smyth, “Using twitter to recommend real-time topical news,” 2009.
- [14] J. McAuley and J. Leskovec, “Hidden factors and hidden topics: understanding rating dimensions with review text,” Proc. 7th ACM Conf. Recomm. Syst. - RecSys '13, 2013.
- [15] C. Wang and D. M. Blei, “Collaborative topic modeling for recommending scientific articles,” in Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '11, 2011.
- [16] B. Hu, Z. Lu, H. Li, and Q. Chen, “Convolutional Neural Network Architectures for Matching Natural Language Sentences,” in International Conference on Neural Information Processing Systems, 2014.
- [17] Z. Lu and H. Li, “A Deep Architecture for Matching Short Texts,” Adv. Neural Inf. Process. Syst., 2013.
- [18] A. Neelakantan, J. Shankar, A. Passos, and A. McCallum, “Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space,” in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014.
- [19] Z. S. Harris, “Distributional Structure,” WORD, 1954.
- [20] P. K. Gopalan, L. Charlin, and D. Blei, “Content-based recommendations with Poisson factorization,” NIPS Adv. Neural Inf. Process. Syst., 2014.
- [21] Y. Kim, “Convolutional Neural Networks for Sentence Classification,” in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014.
- [22] J. Asquith and B. Kolman, “Elementary Linear Algebra,” Math. Gaz., 2007.
- [23] A. Singhal, “Modern Information Retrieval: A Brief Overview,” Bull. IEEE Comput. Soc. Tech. Comm. Data Eng., 2001.
- [24] Y. Wang, M. Huang, xiaoyan zhu, and L. Zhao, “Attention-based LSTM for Aspect-level Sentiment Classification,” 2016.
- [25] X. Zhou, X. Wan, and J. Xiao, “Attention-based LSTM Network for Cross-Lingual Sentiment Classification,” 2016.
- [26] M. Abadi et al., “TensorFlow: Large-scale machine learning on heterogeneous systems,” Methods Enzymol., 1983.
- [27] Y. Sasaki, “The truth of the F-measure,” Teach Tutor mater, 2007.
- [28] D. M. W. Powers and Ailab, “EVALUATION: FROM PRECISION, RECALL AND F-MEASURE TO ROC, INFORMEDNESS, MARKEDNESS & CORRELATION,” J. Mach. Learn. Technol. ISSN, 2011.
- [29] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, “DeepFM: A factorization-machine based neural network for CTR prediction,” in IJCAI International Joint Conference on Artificial Intelligence, 2017.

“N1 (de) N2” structures

Junyuan Zhao

*Department of Chinese Language and Literature
East China Normal University
Shanghai, China
e-mail: jyzhao9802@gmail.com*

Junru Wu

*Department of Chinese Language and Literature
East China Normal University
Shanghai, China
e-mail: jrwu@zhex.ecnu.edu.cn*

Abstract—Based on corpus materials and on-line semantic judgment surveys, this paper investigates the comprehension differences related to the occurrence and deletion of “de” in the Mandarin “N1 (de) N2” structure. By applying PCA and LME modellings on a set of semantic survey data, this study provides a multi-level database of semantic measurements for a set of Chinese “N1 (de) N2” structures as well as a quantitative analysis regarding the correlation between structure-level and constituent-level semantic features. The research shows that: (1) The “de”-occurring structure is more likely to be interpreted as indefinite than the “de”-deletion structure. (2) Animacy of N1 is positively related to the grammaticality of the “de”-occurring structure, while animacy of N1 is negatively related to the grammaticality of the “de”-deletion structure. The research findings provide evidence for prototype effects in the process of language comprehension. We propose that in natural comprehension, there is a high-animacy bias for N1 regarding the “de”-occurring structure; while a low animacy interpretation for N1 is more prototypical for the “de”-deletion structure. Accordingly, the “de”-occurring structure tends to be interpreted as a possessive, while the “de”-deletion structure is more likely to be interpreted as a modifier-head structure.

Keywords—occurrence and deletion of “de”; Chinese de structure; semantic change; quantitative study

I. INTRODUCTION

The Mandarin structure “N1 (de) N2”, where the presence of the functional word “de” is optional in some conditions (e.g., 皇帝(的)女儿, EMPEROR (DE) DAUGHTER, “daughter of the emperor”), where “de” is optional) has long been under linguistic discussion. Following either a descriptive or formal paradigm, decades of efforts have been made to delineate the mechanism lying behind this phenomenon, because of its theoretical importance to the Mandarin determiner system and the Chinese word boundary puzzle. This problem is also relevant to Natural Language Processing (e.g., for entity recognition) and has also been investigated by recent corpus linguistic studies, such as by Zhang et al [1]. Zhang and his colleagues’ recent study relied on formal features of the constituents (i.e., part-of-speech annotation), while whether two slightly different structures (i.e. “de”-occurring and “de”-deletion structures) stand for two distinct entities remained a question.

Among Chinese linguists, Zhu [2] was the first to systematically probe into this problem. He found that the occurrence and deletion of “de” in this structure is related to

several language-internal factors, such as the fixedness of expressions, use of kinship terms and word-lengths of constituents. He also admitted that the selection of “de” is quite flexible and requires further studies.

Following Zhu, a number of researchers have looked into this problem, among which many have approached the problem from a holistic perspective and focused on the property of the whole structure. As Xu [3] suggested, constituents in the “de”-occurring structure are loosely-combined while constituents in the “de”-deletion structure are tightly-combined, which made the constituents an integrated module in the context. This theory of “module integrity” may have predictive power regarding the occurrence and deletion of “de” in such structures. However, this is an intuitive judgement and requires further empirical verification.

Other pertinent studies focused on internal semantic relations of the constituents. Yuan [4] and Wen [5] are among the first linguists to point out that the most basic semantic distinction involved is the distinction between possessive and modifier-head relations, i.e., possessives tend to select for the “de”-occurring structure while modifier-head relations (which is common cross-linguistically[6] in noun phrases) tend to select for the “de”-deletion structure.

Yuan [4] and Wen’s [5] explanation accounts for a large part of materials in linguistic literatures of Mandarin. An alternative semantic explanation was provided by Si [7], who focused on the subcategories of possessives and proposed that alienable possessives must take a “de” while the case of inalienable possessives is more flexible. However, Si’s explanation is subject to reproductive counter-examples (e.g. the possessive 我帽子-I hat-“my hat”¹ is alienable but is a grammatical ‘de’-deletion structure).

Formal linguists also have been avoiding this phenomenon in their analyses because of its obscure and uncertain nature [8]. Thus far, the description and explanation regarding the occurrence and deletion of “de” in the Mandarin “N1 (de) N2” structure are still insufficient. This is partly due to complex factors involved. For instance, the findings of Feng [9] on prosodic constraints and the findings of Lu [10] on pragmatic constraints remind us that this phenomena is not solely syntactically constrained. A specific example provided by Lu was the pragmatic difference between 我的女朋友 (I DE GIRLFRIEND, “my girlfriend”)

¹ A context can be: 我找不到我帽子了(I FIND NEG ARRIVE I HAT PERF, “I cannot find my hat”)(NEG-negative, PERF-perfect).

and 我女朋友(I GIRLFRIEND, “my girlfriend”), with which Lu suggested that the structure with “de” can denote more than one entity while the one without “de” can only denote one, indicating the “de”-occurring structure is indefinite.

A diachronic explanation can be that the Mandarin functional word “de” as well as the structure involved have undergone complex historical changes [10]–[12], which, along with the property of Mandarin as a language lacking formal markers [13], made the semantic and syntactic interpretation of “de” obscure and fuzzy. Nevertheless, a diachronic explanation, if even exist, cannot directly help synchronic or practical predictions.

The nature of this problem obviously conflicts with the binary feature of the long-adopted descriptive paradigm. Hence, an alternative approach that may improve practical prediction of the occurrence of “de” is to assume a (possessives-to-modifier-head) semantic continuum rather than a binary contrast, similar to what has been pointed out by Downing [14] in a study on English compounds. With a probabilistic view, the problem is thus translated to a probability issue regarding grammaticality and predictive semantic features.

Taking this approach, this study focuses on the probabilistic correlations between structure-level and constituent-level semantic features, as well as the relations between semantic features and grammaticality. Inspired by previous studies, which pointed out that categories of possessives show prototype effects and can be predicted by the concreteness and animacy of constituents [15], [16], this study seeks to better model the phenomenon using these semantic features. Quantitative methods were adopted in this study, which include mixed linear logistic regression (LME) [17], Principal component analysis (PCA, with the princomp function provided in R [18]) and hypothetical tests. Target semantic features based on subjective rating were collected with an online survey.

II. ONLINE SURVEY

A. Sampling of Corpus Material

Materials of the online survey were downloaded from the BCC corpus [19] using the query “../n 的 ..n”. 144560 target structures (i.e., “N1 de N2” structure with different constituents, provided along with an original context of approximately thirty characters) were downloaded under the “news” subcategory as raw materials, of which 53800 entries were downloaded time-wise and 90760 time-reversed. For better sampling results, we clustered the raw materials based on vector representations. First, all original contexts were parsed with the jieba [20] parser for Chinese in python. In order to make sure that all target structures are parsed out, we included all these structures in the parser dictionary. Next, we generated semantic representations for all target structures with a 100-dimension vector, using the word2vec package from the gensim [21] python library. The context required for word2vec was constructed by concatenating all parsed contexts of the original material. Then we performed a K-means clustering using the sklearn package [22] in python. The goodness of clusters was evaluated with silhouette scores [23] (see Table I).

TABLE I. NUMBER OF CLUSTERS ~ SILHOUETTE SCORES

N clusters	Silhouette score
2	0.302
3	0.480
4	0.126

Table I shows that a cluster number of three yielded the best clustering result. Two thousand samples were extracted using stratified sampling, based on the clustering result.

A further filtering of samples was conducted by manual examination. Undergraduates who have received fundamental linguistic training were recruited for this task, following the criteria that a target structure should not overlap with other constituents in the original context (e.g., the structure 民族的精神 (ETHNIC GROUP DE SPIRIT, “spirits of the ethnic group”) extracted from 中华民族的精神 (THE CHINESE ETHNIC GROUP DE SPIRITS, “spirits of the Chinese ethnic group”) was rejected because the constituent 民族 overlapped with the constituent 中华民族). Eventually two hundred “de”-occurring samples were selected and two hundred “de”-deletion samples were generated accordingly by deleting “de”. The two hundred “N1 (de) N2” pairs were used as materials for the online survey.

Note that, most of the materials (“de”-occurring structures) that passed the manual examination carry an animate N1 (e.g. a person), which reveals a natural tendency of the “de”-occurring structure to select for an animate N1.

B. Survey Design

An online survey was designed on the Qualtrics [24] platform with four tasks involved: (a) grammaticality judgement, (b) concreteness rating, (c) animacy rating and (d) definiteness judgement. The grammaticality judgement and definiteness judgement targeted the structures as a whole and both “de”-occurring and “de”-deletion structures were rated (2×1). The concreteness rating and animacy rating targeted both the structures (“de”-occurring, “de”-deletion) and the constituents (N1, N2). Each constituent was rated three times, twice as embedded in (the two) structures and once in isolation, $3(N1/N2/\text{structure}) \times 2(\text{occurring/deletion}) + 2(N1/N2)$ in isolation).

(a) The grammaticality judgement asked the participants to decide whether the target structure is acceptable. (b) The concreteness rating asked the participants to rate the concreteness of the target constituents on a 5-point scale, using the instructions translated from previous English studies on semantic concreteness [25]; target constituents were marked red. (c) The animacy rating required the participants to rate the animacy of the target constituents on a 5-point scale. The definiteness judgement was based on the result of the acceptability judgement of the Chinese existential sentence 有...在这儿 (HAVE ... AT HERE, “there be ... here”), following a previous study [26] on the definiteness of Chinese N-de-N structures.

Twenty-four native Standard Mandarin speakers participated in the on-line survey and were paid. We balanced the two hundred materials across twenty-four participants in a way such that a single N-(de)-N pair is to be rated by three different participants on all target semantic features. To avoid

potential lexical learning effects², we set a time limit of five seconds per structure to tasks that involve grammaticality/acceptability judgements.

III. DATA ANALYSIS

We received twenty-four responses for all twenty-four surveys distributed and data of 199 structures were valid out of the two hundred samples. There were 14328 (199 N1-N2 pairs \times 3 participants \times 24 measurements) data points in total. We carried out the following statistical analysis in R (version 3.5.3).

A. Probabilistic Distribution of Concreteness

First we looked at the distribution of concreteness of constituents:

On the contrary, the concreteness distribution of N1 does not fit the canonical distribution of concreteness as found by previous researches, which suggests that the structures (with and without “de”) have semantic constraints on the N1 constituent, as compared with the distribution in a natural context where target words are separately examined.

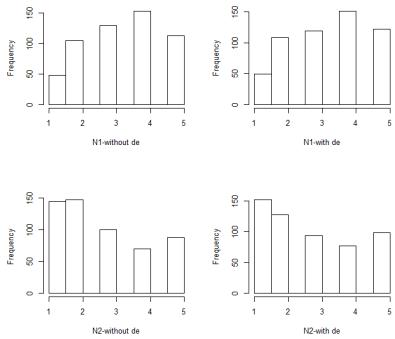


Figure 1. Concreteness distribution of N1 and N2

As shown in Figure 1, the concreteness distributions of N1 and N2 differ in both structures with and without “de”. The concreteness distributions of N1 in both structures are bimodal, while the concreteness distributions of N2 in both structures resembled a normal distribution. The distribution of N2 aligned with the results of previous studies [25], [27] on the concreteness of unrestricted English words based on a large scale survey.

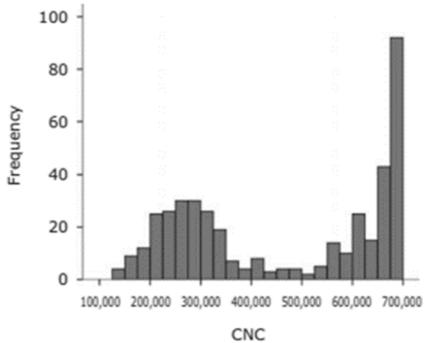


Figure 2. Concreteness distribution of unrestricted English words, from Della Rosa (2010) [24] (CNC-concreteness)

B. Semantic Comprehension of Constituents within Structures versus in Isolation

Next, we further investigated the constraints that “de”-occurring and “de”-deletion structures exert on the constituents. A linear regression was performed, using the individual concreteness of N1 as the independent variable and the concreteness of N1 in the structures as the dependent variable. The regressions for both structures were performed. The results were plotted as is shown in Fig.3, using a black line ($y = x$) for reference.

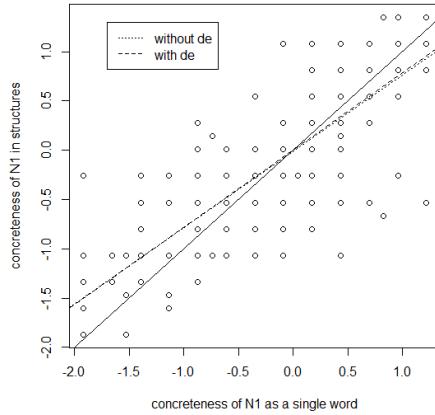


Figure 3. Concreteness of bare N1 v.s. Concreteness of N1 in structures

Figure 3, along with the following t-tests, shows that different structures yield different interpretations of concreteness regarding N1. For the “de”-deletion structure, $t(197) = 26.54$, $p < 0.001$, $R^2 = 0.78$; for the “de”-occurring structure, $t(197) = 24.29$, $p < 0.001$, $R^2=0.75$. Both distributions showed a pattern of “neutralization”, which means that the concreteness of N1 in both structures covers a smaller scale than the original bare noun. This effect applies to both structures, suggesting that “N1 (de) N2” structures alters the semantic interpretations of their constituents. We also found a similar difference regarding constituent animacy in bare noun form versus in structures, see Figure 4.

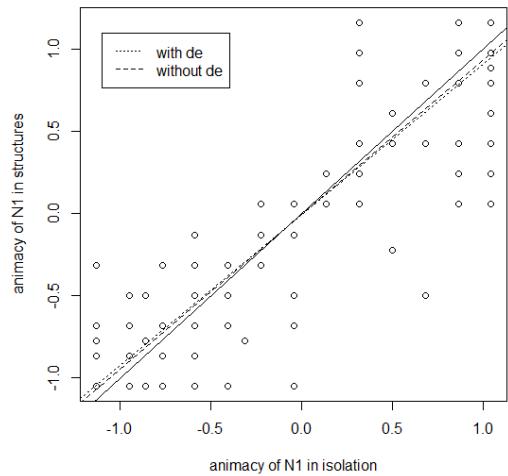


Figure 4. Concreteness of N1 in isolation v.s. Concreteness of N1 in structures

² In a pilot survey, we noticed that without time limits, many apparently-unacceptable constructions were marked as acceptable. This is probably due to the participants’ integration of the novel constructions into their mental lexicon in a relatively short period of time (i.e. lexicalization on an individual level).

We further investigated the relations between semantic features of constituents and the structure as a whole with LME modelling [17]. Taking the concreteness/animacy of the whole structure as the dependent variable and semantic features of the constituents as the independent variables, models for both structures showed that the semantics of the whole structure correlates with the semantic features of N2 ($t = 26.45$, $p < 10^{-5}$), but not with N1. These findings provided quantitative evidence for the headedness of the Chinese modifier-head structure.

C. Definiteness and “de” Occurrence

To investigate the definiteness of different structures, we performed a paired t-test on definiteness scores. Since the raw definiteness data were binary regarding the task involved (i.e., 1-definite, 0-indefinite), by-pair means were calculated beforehand. It turned out that the definiteness of the two structures are significantly different, $t(197) = 2.65$, $p = 0.009 < 0.01$. The “de”-occurring structure is more likely to be interpreted as indefinite and the “de”-deletion structure is more likely to be interpreted as definite. This finding aligned with previous pragmatic studies [10].

D. Principal Component Analysis for Semantic Features

Based on the matrix of correlation, we found apparent multi-collinearity across the semantic measurements (i.e., a high correlation between different semantic features, such as between the concreteness of N1 in both structures). Therefore, we performed a PCA on these predictors before investigating the structure of the semantic predictors that may account for the grammaticality of “N1 (de) N2” structures. As shown in Figure 5, a minimum of four principal components can explain most of the variance within the semantic predictors.

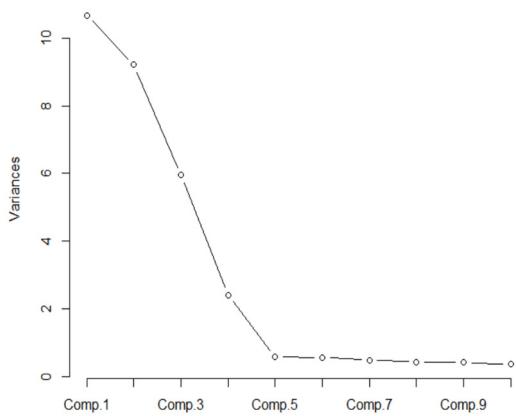


Figure 5. Variances of components of PCA

As is shown in Table II, the four principal components explained relatively separately the four theory-relevant semantic features (i.e., the concreteness and animacy of both N1 and N2), which did not conflict with our further analysis.

E. Linear Mixed Effect Logistic Regression

In order to further delineate the possible different semantic interpretations associated with “de”-occurring versus “de”-deletion structures, we performed a linear mixed effect logistic regression, using the original semantic measurements and the PCA components separately.

TABLE II. LOADINGS OF PRINCIPAL COMPONENTS

	Comp. 1	Comp. 2	Comp. 3	Comp. 4
Concreteness of N1, with “de”	0.230		0.178	0.501
Concreteness of N2, with “de”	0.351	-0.164	0.138	-0.247
Concreteness of structure, with	0.332	-0.154	0.163	-0.116
Concreteness of N1, without “de”	0.205		0.194	0.493
Concreteness of N2, without “de”	0.334	-0.162	0.159	-0.191
Concreteness of structure, without	0.348	-0.152	0.164	-0.129
Concreteness of single N1	0.201		0.188	0.494
Concreteness of single N2	0.345	-0.161	0.111	-0.261
Animacy of N1, without “de”	0.191	0.529		-0.119
Animacy of N2, without “de”	0.190		-0.393	
Animacy of structure, without	0.190		-0.397	
Animacy of N1, with “de”	0.207	0.532		
Animacy of N2, with “de”	0.185		-0.391	
Animacy of structure, with	0.190		-0.397	
Animacy of single N1	0.182	0.536		-0.135
Animacy of single N2	0.158		-0.401	

We modelled the phenomenon using occurrence/deletion of “de” (a binary variable) as the dependent variable. The principal components which explained most parts of the concreteness and animacy of constituents (Comp.1 through Comp.4), were taken as the independent variable; participant and the pair ID were taken as random factors. Based on the fact that this study targeted at the semantics of originally grammatical “de”-occurring structures, we omitted the data points for which “de”-occurring structures were labelled as ungrammatical. Several different models using other original semantic features (e.g., concreteness of N1 in “de”-occurring structures) of constituents were also fit. The models were evaluated with ANOVA (analysis of variance) and the best-fit model was selected and reported in TABLE III ($\chi^2 = 0.153$, $p < 2 \times 10^{-16}$).

TABLE III. MODEL SUMMARY

Formula: validity_0 ~ conc_left_comp + anm_left_comp + (1 subject) + (1 pair_number)				
Fixed effects				
Predictor	Estimate	Std. Error	Z value	p value
Principal comp. of concreteness N1	-0.155	0.100	-1.561	0.118
Principal comp. of animacy N1	-0.170	0.056	-3.018	0.002**
Random effects				
Predictor	Variance	Std. Deviation		
pair number	2.746	1.657		
participant	0.467	0.684		

^a: validity_0: grammaticality of the “de”-deletion structure; conc_left_comp: concreteness of N1 (after PCA); anm_left_comp: animacy of N1 (after PCA). *p<0.05, **p<0.01, ***p<0.001

As is shown by model estimates in TABLE III, there is a significant negative correlation between animacy and grammaticality of the “de”-deletion structure, which indicates that the less animate N1 is, the more acceptable the “de”-deletion structure is. Inferring from the fact that the survey elicited participants’ on-line interpretations, we suggest that this statistical tendency implies an interpretational tendency. The principal component for the concreteness of N1 showed a similar effect, although the effect did not reach significance.

IV. DISCUSSION AND CONCLUSION

This study shows different semantic and pragmatic interpretations of the Mandarin “N1 (de) N2” structure with and without “de” using a quantitative approach. We also provided a multi-level database of constituent semantic measurements.

Regarding the pragmatics of the structure, we found that the “de”-occurring structure tend to be less definite than the “de”-deletion structure. This finding aligns with Lu [10] on the definite-indefinite pragmatic differences of the two structures on a paragraph level. Moreover, the statistical analysis also indicates that the definite/indefinite distinction is not binary nor deterministic.

Regarding the semantics of these structures, the results suggest that the grammaticality of the “de”-deletion structure is negatively related with the animacy of N1, which indicates that the “de”-deletion structure tend to have a low animacy interpretation for N1, since the measurements are participants’ on-line interpretations. Based on the linear nature of the statistic model used, we infer that the “de”-deletion structures that were considered unacceptable (i.e. structures that are only acceptable when “de” is present) tend to have a high animacy interpretation for N1. This quantitative evidence supports that the semantic differences between the two structures associated with structural differences, which is aligned with the very fundamental notion of grammatical constructions[28] and in the meantime rejected a possible theoretical alternative of viewing this problem as a case of ellipsis, as ellipsis hardly alters the meaning of a phrase or a sentence.

Considering that a typical possessor should be animate and concrete, we propose that the “de”-occurring structure is influenced by the prototype effect of possessives in the psychological process of language comprehension, while the “de”-deletion structures tend to be interpreted as normal modifier-head compounds. This finding is in-line with previous cognitive studies which showed that semantic possessives have a tendency to select for a more animate N1 [15]. As mentioned in section II.A, we also noticed that the filtered materials from the BCC corpus were mostly possessives with an animate N1, which provide evidence for a preference for animate N1s in the “de”-occurring structure, in natural language use.

Although this study provided a quantitative insight for this long-discussed problem, the following points require further investigation. First, since we were not able to search for “de”-deletion structures on BCC corpus, only the “de”-occurring candidate structures were natural, and the “de”-deletion candidate structures were generated based on these “de”-occurring structures. This may have introduced some research

bias. Since “de”-deletion structures are quasi-compounds according to some Chinese linguists, a further research based on natural “de”-deletion materials could benefit the investigation of their role in lexicalization. Another sampling bias may come from the fact that all materials were downloaded under the “news” subcategory, which might not reflect the real-world distribution. Also, the survey targeted at only college students whose age/education-specific linguistic background and therefore might influence the research results.

Second, the tendency as is shown by the LME model (in section III.E) have a few alternative explanations. One explanation can be made referring to the prototype effect as proposed above. However, whether this prototype effect stemmed from syntactic influence or merely from a preference regarding the use of a certain surface structure remained to be discussed. An alternative explanation can be made from a psycholinguistic perspective: since “de”-deletion structures are indistinguishable from compounds by its surface form, and it has been shown that constituents in compounds take less cognitive resources than constituents that are morphologically-free in a sentence or a structure [29], it could also be that the comprehension bias regarding “de”-deletion structures are rooted in the prototypical comprehension mechanisms of compounds. Admittedly, this interpretation requires further experimental investigation.

To conclude, this study, with a quantitative approach, revealed pragmatic and comprehension differences of the Mandarin “N1 (de) N2” structures. The results suggested possible prototype effects of the two structures, which serves as an evidence for viewing “N1 (de) N2” structures as grammatical constructions.

REFERENCES

- [1] C. Zhang, Z. Zhang, G. Rao, and E. Xun, “Research on extraction of simple modifier-head chunk based on Corpus,” presented at the Chinese Lexical Semantic Workshop, Beijing, China, 2019.
- [2] D. Zhu, Collected Works of Zhu Dexi, vol. 4. The Commercial Press, 1999.(in Chinese) (朱德熙, 朱德熙文集, vol. 4. 商務印書館, 1999.)
- [3] Y. Xu, “The occurrence and deletion of “de” when using personal pronouns as modifiers revisited”, Studies of the Chinese Language, no. 01, pp. 21-27+95, 2008.(in Chinese) (徐阳春, “也谈人称代词做定语时的‘的’字的隐现,” 中国语文, no. 01, pp. 21-27+95, 2008.)
- [4] Y. Yuan, “Implying predicate and its syntactic influence——on the denoting rules of “de” structures and the grammatical and semantic function of “de”, Studies of the Chinese Language, no. 04, pp. 241–255, 1995.(in Chinese) (袁毓林, “谓词隐含及其句法后果——‘的’字结构的称代规则和‘的’的语法、语义功能,” 中国语文, no. 04, pp. 241–255, 1995).
- [5] Z. Wen, “Identifying the semantic relation between N1 and N2 in the ‘N1 de N2’ modifier-head structure”, Linguistic researches, no. 03, pp. 22–27, 1999.(in Chinese) (文贞惠, “‘N_1(的)N_2’偏正结构中 N_1 与 N_2 之间语义关系的鉴定,” 语文研究, no. 03, pp. 22–27, 1999.)
- [6] G. Fanselow, *The Parametrization of Universal Grammar*. John Benjamins Publishing, 1993.
- [7] F. Si, “‘Zhao Benshan’s grandfather’ and ‘Zhao Benshan’s hat’——on two kinds of possessives in Chinese”, Language teaching and studies, no. 02, pp. 43–51, 2014. (in Chinese) (司富珍, “赵本山的爷爷”和“赵本山的帽子”——漫谈汉语中的两种领属结构,” 语言教学与研究, no. 02, pp. 43–51, 2014.)
- [8] L. L.-S. Cheng and R. Sybesma, “De as an underspecified classifier: first explorations,” *Yuyánxué lùncóng*, vol. 39, pp. 123–156, 2009.
- [9] S. Feng, Prosodic syntax in Chinese. Shanghai Educational Publishing House, 2000. (in Chinese) (冯胜利, 汉语韵律句法学. 上海教育出版社, 2000.)

- [10] B. Lu, “On the basic and derived functions of 'de'”, *Chinese Teaching in the World*, no. 01, pp. 14-29+2, 2003. (in Chinese) (陆丙甫, “‘的’的基本功能和派生功能——从描写性到区别性再到指称性,” *世界汉语教学*, no. 01, pp. 14-29+2, 2003.)
- [11] Y. Shi, “On the uniform grammatical function of 'de'”, *Chinese Teaching in the World*, no. 01, pp. 16–27, 2000. (in Chinese) (石毓智, “论‘的’的语法功能的同一性,” *世界汉语教学*, no. 01, pp. 16–27, 2000.)
- [12] M. Zhu, “Grammatical functions of 地,底 in Zhu Zi Yu Lei”, *Studies of the Chinese Language*, no. 03, 1982. (in Chinese) (祝敏彻, 《朱子语类》中‘地’‘底’的语法作用,” *中国语文* 第 3 期, 1982.)
- [13] D. Zhu, *Lectures on the Chinese Grammar*. Commercial Press, 1982.(in Chinese) (朱德熙, 语法讲义. 商务印书馆, 1982.)
- [14] P. Downing, “On the Creation and Use of English Compound Nouns,” *Language*, vol. 53, no. 4, pp. 810–842, 1977.
- [15] Z. Liu, Y. Ren and D. Zhong, “Constraints of animacy on possessives”, *Journal of Foreign Languages*, vol. 41, no. 04, pp. 30–43, 2018. (in Chinese) (刘正光, 任远, 和钟丹凤, “领属关系的生命度制约,” *外国语(上海外国语大学学报)*, vol. 41, no. 04, pp. 30–43, 2018.)
- [16] N. Ge, “A review on studies of the possessive category”, *Language Education*, vol. 4, no. 04, pp. 62–68, 2016. (in Chinese) 葛娜娜, “领属范畴研究综述,” *语言教育*, vol. 4, no. 04, pp. 62–68, 2016.
- [17] D. Bates, M. Maechler, B. Bolker, and S. Walker, “lme4: Linear mixed-effects models using Eigen and S4,” *R package version*, vol. 1, no. 7, pp. 1–23, 2014.
- [18] R Core Team, *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, 2019.
- [19] E. Xun, G. Rao, X. Xiao and J. Zang, “Developing the BCC corpus with big data”, *Corpus Linguistics*, vol. 3, no. 1, pp. 93–118, 2016.
- [20] J. Sun, ‘*Jieba Chinese word segmentation tool*’. 2012.
- [21] R. Řehůřek and P. Sojka, “Gensim—statistical semantics in python,” *statistical semantics; gensim; Python; LDA; SVD*, 2011.
- [22] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, p. 2825–2830, Oct. 2011.
- [23] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, Nov. 1987.
- [24] Qualtrics. Provo, Utah, USA: Qualtrics. 2005.
- [25] P. A. Della Rosa, E. Catricalà, G. Vigliocco, and S. F. Cappa, “Beyond the abstract—concrete dichotomy: Mode of acquisition, concreteness, imageability, familiarity, age of acquisition, context availability, and abstractness norms for a set of 417 Italian words,” *Behavior Research Methods*, vol. 42, no. 4, pp. 1042–1048, Nov. 2010.
- [26] B. H. Partee, “A note on Mandarin possessives, demonstratives, and definiteness,” in *Studies in Language Companion Series*, vol. 80, B. J. Birner and G. Ward, Eds. Amsterdam: John Benjamins Publishing Company, 2006, pp. 263–280.
- [27] M. Brysbaert, A. B. Wariner, and V. Kuperman, “Concreteness ratings for 40 thousand generally known English word lemmas,” *Behav Res*, vol. 46, no. 3, pp. 904–911, Sep. 2014.
- [28] A. E. GOLDBERG, “The inherent semantics of argument structure: The case of the English ditransitive construction,” *Cognitive Linguistics (includes Cognitive Linguistic Bibliography)*, vol. 3, no. 1, pp. 37–74, 2009.
- [29] D. Koester and N. O. Schiller, “Morphological priming in overt language production: Electrophysiological evidence from Dutch,” *NeuroImage*, vol. 42, no. 4, pp. 1622–1630, Oct. 2008.
- (in Chinese) (荀恩东, 饶高琦, 肖晓悦, and 臧娇娇, “大数据背景下 BCC 语料库的研制,” *语料库语言学*, vol. 3, no. 1, pp. 93–118, 2016.)

Effects of English Capitals On Reading Performance of Chinese Learners:

Evidence from Eye Tracking

Yang Wei / Fu Xinyu

English Department of School of Foreign Studies

Nanjing University of Science and Technology

Nanjing, China

yangwei@njust.edu.cn

Abstract—Native English speakers need more time to recognize capital letters in reading, yet the influence of capitals upon Chinese learners' reading performance is seldom studied. We conducted an eye tracker experiment to explore the cognitive features of Chinese learners in reading texts containing capital letters. Effect of English proficiency on capital letter reading is also studied. The results showed that capitals significantly increase the cognitive load in Chinese learners' reading process, complicate their cognitive processing, and lower their reading efficiency. The perception of capital letters of Chinese learners is found to be an isolated event and may influence the word superiority effect. English majors, who possess relatively stronger English logical thinking capability than non-English majors, face the same difficulty as the non-English majors do if no practice of capital letter reading have been done.

Keywords-Capital letter; Reading; Congition; Eyemovements

I. INTRODUCTION

Capital letter (upper-case letter) is the unique type form in English language. All letters in a word or just the first letter could be capitalized in writing or printing. Capitals in English have both grammatical and rhetoric functions. The pronoun *I*, the first letter of a sentence, the first letter of a proper noun, abbreviations and acronyms must be capitalized. Titles, headlines, headings, logos, public signs, etc. are often capitalized for emphasis.

Conclusions on the influence of capitals upon the reading performance of English natives were quite consistent that all capitals slowed up the reading and increase interest and effort [1, 2, 3, 4]. Tinker and Paterson [4] were among the first ones to observe more fixations with fewer words perceived per fixation in reading all capitals through eye tracking experiments. The difference in shape between upper and lower case letters were considered to contribute to the poorer legibility of upper-case letters [5, 6, 7]. Lower-case letters included ascenders and /or descenders, and form clusters of letters, often a whole word (e.g. cat, dog, pig) while upper-case letters formed the shape of a horizontal rectangle (e.g. CAT, DOG, PIG), and people tended to remember alternating shapes better than square shapes [8].

Chinese characters are in square form but little study has connected Chinese learners' character reading ability with their reading performance of English capitals. Yet Chinese characters are logograms, which has not variation in type form. So English capitals have a higher possibility to be a difficulty in English learning. That is why we would like to explore and compare the cognitive features of Chinese

learners when they are reading English upper and lower case letters.

For Chinese learners, rules about capitals usually are the contents in grammar learning. Although studies on their reading performance of English capitals were relatively few, the results were quite the same that Chinese learners considered capitals were very difficult to read and the reading time on capital words were significantly longer than lower-case words [9, 10]. Chen [11], by conducting tests on recognizing all capital words and phrases, commented that English major students had shown a marked accuracy rate than non-English major students. Usually we take English majors as having higher foreign language proficiency and non-English majors having normal one. Chen's result seemed to prove a positive correlation between English language proficiency and the reading performance of capitals. However, on reading longer sentences with all letters capitalized, both English majors and non-English majors showed very low recognizing accuracy. Also, using tests to measure reading accuracy might be greatly affected by how many words the subjects know and tests cannot describe the cognitive features of Chinese learners in reading capitals. Mistakes and interferences took place in reading may come from heavy cognitive load [12, 13, 14]; but we see little research about cognitive load in capital letters reading in China via eye movement measures like reading time, fixation count, saccade count, etc. experiments. These measures can offer a more direct and objective view of processing difficulty and proficiency in reading both from spatial and temporal dimensions. In China, most eye tracking experiments were conducted to explore the reading performance of Chinese language.

The aim of our experiment was to get six eye movement measures when Chinese learners are reading materials containing English capitals so as to analyze their cognitive features in capital letter reading. Two questions were going to be answered: (1) Does capital letters affect Chinese learners' the cognitive load, cognitive processing, and reading proficiency in reading and how; (2) Does English language proficiency influence the reading performance of capital letters and how.

II. METHODS

This study was 3 ('Text condition': word, phrase and sentence reading) \times 3 ('Text type': the All-letter-capital, All-initial-capital and Normal texts) \times 2 ('Subject group': English and non-English majors) factors mix designing.

A. Participants

A total of 46 college students from Nanjing University of Science and Technology (a major university in China, hereinafter NJUST), none of whom had participated in the pretest or the pilot experiment, were recruited in the eye-movement experiment for payment. They were all sophomores and aged between 18 and 22. Among them, 22 were English majors and were all female participants. The other 24 were non-English majors, 16 males and 8 females. In NJUST, female students in the English major greatly outnumber male students; while in other non-English majors, male students greatly outnumber female students. We cannot find an equal number of participants for both genders. For convenience, for English major, we choose all female participants and for non-English major, we try to include as many as female participants. Having learned English for more than ten years, all of them passed *Syllabus for College English Test-Band Four* (CET-4 for short). They were all native speakers of Mandarin with normal or corrected-to-normal vision, and had no history of neurological or language impairments. All of them provided informed consents. They were not informed of the purpose of the experiment, and had no previous exposure to similar experimental items. A total of 43 participants were successfully eye-tracked including 21 English majors and 22 non-English majors and their data be analyzed.

B. Materials

Experimental materials in this study were classified into 3 text types and 3 text conditions, altogether 9 presenting modes as shown in TABLE I. Text conditions include word, phrase and sentence. Text types include: (1) normal text which has no capital letter in word and phrase, but has the first letter in sentence capitalized; (2) all-capital text in which every letter is capitalized; (3) all-initial-capital text in which the first letter of each word is capitalized. A total 18 words, 12 phrases and 9 sentences were selected after preparation and pretest.

For words and phrases, subjects' familiarity and length were controlled. Firstly, these words came from the 2016 edition of CET-4 and a total of 6,127 words were left after the removing of unsuitable words. Secondly, the command statement '=LEN()' in EXCEL was adopted to calculate how many letters each word contained, and words with six letters had the largest number, totaling 891. Thirdly, Online English Dictionary from Collins was used to decide the frequency of each word. As words with frequency 1 and 2 were very likely to go beyond subjects' vocabulary, words for the pretest were randomly chosen from words with frequency 3, 4 and 5. Finally, 16 words with frequency 3, 12 words with frequency 4 and 12 words with frequency 5 were included in the pretest.

Phrases came from *Nucleus College English: Reading and Writing Course 1 and 2*, the textbooks for the course College English in NJUST. Altogether 227 phrases were selected from these two books. Secondly, the number of characters in each phrase was calculated with the command statement '=LEN()' in EXCEL, and words with 11 characters had the largest number of 27. Thirdly, the number of spaces in these 27 phrases was counted, with one space in 11 phrases and two spaces in 16 phrases.

Finally, ten phrases with one space and ten phrases with two spaces were randomly chosen for the pretest.

For sentences, besides familiarity and length, idiomacity and diversity were also taken into consideration. In consideration of the screen size of the display computer, it was decided that sentences with 28 characters fit this experiment best. Since all sentences were completed sentences containing spaces and punctuation, the 28 characters mentioned here included spaces and punctuation. Firstly, these sentences came from expressions for communication in *Nucleus College English: Listening and Speaking Course 2*. Four topics were adopted, and ten sentences were made for each topic, totaling 40. Secondly, two foreign teachers from School of Foreign Studies in NJUST respectively judged whether these sentences were idiomatic expressions. Eight sentences that were or might be Chinglish were removed. Thirdly, considering the diversity of expressions, we chose 20 sentences from those 32 sentences left.

In the pretest, 80 multiple-choice questions were made for these 40 words, 20 phrases and 20 sentences. The pretest was finished by 29 English majors from Nanjing Agricultural University and 26 non-English majors from NJUST. A total of 49 valid questionnaires were collected. After calculating the accuracy rate for each question, 18 words, 12 phrases and 9 sentences were chosen as experimental materials with the lowest accuracy rate of 95.92%. Hence, it was guaranteed that subjects had no difficulty in recognizing and understanding these materials.

TABLE I. NINE PRESENTING MODES AND SAMPLES

Text Type	Text Condition	Sample
All-capital	word	INTEND
	phrase	ON OCCASION
	sentence	NO, BUT THANKS FOR OFFERING.
All-initial-capital	word	Intend
	phrase	On Occasion
	sentence	No, But Thanks For Offering.
Normal	word	intend
	phrase	on occasion
	sentence	No, but thanks for offering.

C. Apparatus

All experiments were conducted in a sound-attenuated booth in the Language Cognition and Speech Science Lab of School of Foreign Studies in NJUST. The laboratory was equipped with one EyeLink 1000 Plus eye tracker, produced by SR Research Ltd. Eye movements of all subjects were monitored from their right eyes in the Remote Mode supported by the eye tracker. During the experiment, a display computer and a host computer were working at the same time. The display computer was used to present experimental materials, and the host computer was used to run the experimental program. The monitor had a refresh frequency of 75 Hz and a resolution ratio of 1024×768 . A headrest was used to stabilize subjects' heads and to minimize the interference caused by their body movements.

D. Procedure

The entire experiment for one subject included the practice experiment and the formal experiment. Comprised of three words, three phrases and three sentences, the practice experiment was conducted to let the subject acquire a general understanding. All those materials presented in the practice experiment would not appear in the formal experiment. The formal experiment would be conducted if the subject provided stable eye movements. Before every formal experiment started, a nine-point calibration was performed. After that, each trial started with a one-point calibration in the middle of the screen. The subject was instructed to read the materials silently at his/her normal speed of reading. All materials were displayed in single lines on the screen. After reading each material, the subject needed to answer a multiple-choice question with two choices about the Chinese meaning of the material. No feedback would be given to him/her. Then, the subject needed to press the SPACE button on the keyboard to go on to the next material. The entire experiment took approximately 20 minutes for each subject.

The rate of accuracy in answering those multiple-choice questions was over 92%, as the lowest accuracy rates of English and non-English majors were 92.31% and 94.87% respectively. These accuracy rates were well above the chance level, implying that the subjects carefully read and basically understood the experimental materials.

E. Analysis

Data files were exported and sorted out by EXCEL for statistical analyses. The software utilized for processing the data was SPSS (Version 25.0 for Windows). In consideration of the fact that 21 English majors and 22 non-English majors provided valid samples, the data could be assumed to be normally distributed. Prior to analysis, a part of fixations, which were shorter than 80 ms or longer than 1,200 ms, were excluded from analysis according to Rayner and Pollatsek [15]. As a result, about 1.83% of the data were deleted.

These eye-movement data were processed with a multivariate analysis of variance, followed by Bonferroni adjusted post-hoc tests. Five measures were calculated and analyzed, including Fixation Count/FC, Progressive Saccade Count/PSC, Regressive Saccade Count/RSC, Mean Saccade Amplitude/MSC(only for sentences) and Total Time/TT.

a) FC: Fixation count is the number of times when eyes fixate on a specific area. The number of fixations is a convincing indicator of the cognitive load [16]. According to Henderson and Ferreira [17], reading a difficult text causes the reader to generate significantly more fixations than reading an easy text. Besides, with the same text, skilled readers generate significantly fewer fixations than average readers [18].

b) TFD: Total fixation duration is closely related to the cognitive load, and readers generate longer fixations when the cognitive load becomes heavier [19]. If the text becomes more difficult, the reader's fixations will become longer [20]. Also, total fixation duration of skilled readers is shorter than that of average readers when they read the same text [21]. Although TFD and TC can both reflect

cognitvie load, since the two are frequently-used measurements in eyemovement experiments, we finally decide to include both in our analysis.

c) PSC: According to Rayner [22], the reader does not acquire information during saccades, and a new region of the text is presented after every saccade. Though new information is not obtained during saccades, cognitive processing does continue during most saccades [23, 24].

d) RSC: Regressive saccades, also called regressions, serve the purpose of rereading a specific part of the text [25]. The number of regressive saccades reveals the reader's cognitive reprocessing of previous information. It has been shown to be a good indicator of the reading efficiency [26, 27, 28]. With the text becoming more difficult, the reader makes more regressions (Rayner, 1998). Besides, skilled readers tend to generate significantly fewer regressions than average readers [21].

e) MSA: Mean saccade amplitude is the mean value of angular distances that eyes travel during saccades. Mean saccade amplitude concerns the reader's cognitive processing and reading efficiency [16]. Longer saccades show that the reader can acquire more information from fixations before saccades, and that he/she can read the text faster and more efficiently [24, 29]. As the text gets more difficult, the reader's saccades get shorter [20]. Saccades of skilled readers are longer than those of average readers [21]. However, it is not suitable to analyze word or phrase reading with mean saccade amplitude, because words and phrases are not long enough for readers to generate long saccades. Therefore, in this study, mean saccade amplitude is only adopted to analyze sentence reading.

f) TT: Total reading time refers to the sum of all fixations and saccades during reading. In other words, it is the sum of time spent in reading a specific text. It is different from total fixation duration mentioned above. In this study, the measurement of total reading time starts when the subject presses the SPACE button to read the text, and ends when he/she presses the SPACE button again to answer the multiple-choice question. Total reading time is an important measure to reflect the reading efficiency.

Repeated measures ANOVAs were applied to analyze the FC, TFD, PSC, RSC, MSA, and TT with two with-in subjects factors—Text condition (three levels: word, phrase, sentence), Text type (three levels: All-captitals, All-initial-capitals, Normal) and a between-subject factor—Subject group (two levels: English major and Non-English major).

III. RESULTS

The results in TABLE II. show that the main effects of ‘Text condition’ ($F(12, 3278) = 285.542, p < .001$), ‘Text type’ ($F(12, 3278) = 19.481, p < .001$) and ‘Subject group’ ($F(6, 1639) = 39.521, p < .001$) are all significant on the six measures. Nevertheless, there is no significant three-way ‘Text condition × Text type × Subject group’ interaction.

TABLE II. RESULTS OF GENERAL LINEAR MODEL WITH THE EFFECTS OF ‘TEXT CONDITION’, ‘TEXT TYPE’ AND ‘SUBJECT GROUP’. ITEMS IN BOLD INDICATE SIGNIFICANT FINDINGS.

Effect	df1	df2	F	p
(Intercept)	6	1639	3580.533	<.001***
Text condition	12	3278	285.542	<.001***
Text type	12	3278	19.481	<.001***
Subject group	6	1639	39.521	<.001***
Text condition × Text type	24	5719	3.479	<.001***
Text condition × Subject group	12	3278	5.262	<.001***
Text type × Subject group	12	3278	0.892	0.555
Text condition × Text type × Subject group	24	5719	0.897	0.607

A significant two-way ‘Text condition × Text type’ interaction ($F(24, 5719) = 3.479, p < .001$) was witnessed. It proved that for Chinese learners as a whole, capital letters in text increase their cognitive load, complicate their cognitive processing, and lower their reading efficiency. Yet, we cannot say English major students exhibit a marked difference on cognitive load, cognitive processing and reading efficiency because with no significant ‘Text type × Subject group’ interaction was observed.

Bonferroni adjusted post-hoc tests are then performed on the two interactions for further comparisons. The results show: (1) participants generally perform word and phrase reading to a similar degree across the all-initial-capital and normal types. However, word and phrase reading show significant differences on the all-capital type, as participants generate more and longer fixations, more progressive and regressive saccades and longer reading time. This indicates that compared with the other two types of words and phrases, the all-capital ones pose a challenge to Chinese learners by increasing cognitive load, lowering reading efficiency and complicating cognitive processing. (2) Things became complicated on sentence condition as shown in Figure 1. The all-initial-capital sentences showed a marked different from normal type sentences. Students, when reading all-initial-capital sentences, generates more fixation ($p < 0.01$), longer fixation duration ($p < 0.01$), more progressive ($p < 0.01$) and regressive saccades ($p < 0.01$) and longer reading time ($p < 0.01$). Between all-initial-capital type and all-initial type, marked difference was found on fixation duration ($p < 0.001$), but not on fixation counts; and another marked different was found on mean saccade amplitude ($p < 0.01$), but not on progressive and

regressive saccades. Between all-initial type and normal text type, marked differences were observed across all the six measures. Combining result in (1), it indicates that, on sentence condition, with an increased number of capital letters, Chinese learners experienced heavier cognitive load and lower reading efficiency. Cognitive features of all-initial-capital text reading were thus similar to those of all-capital text reading.

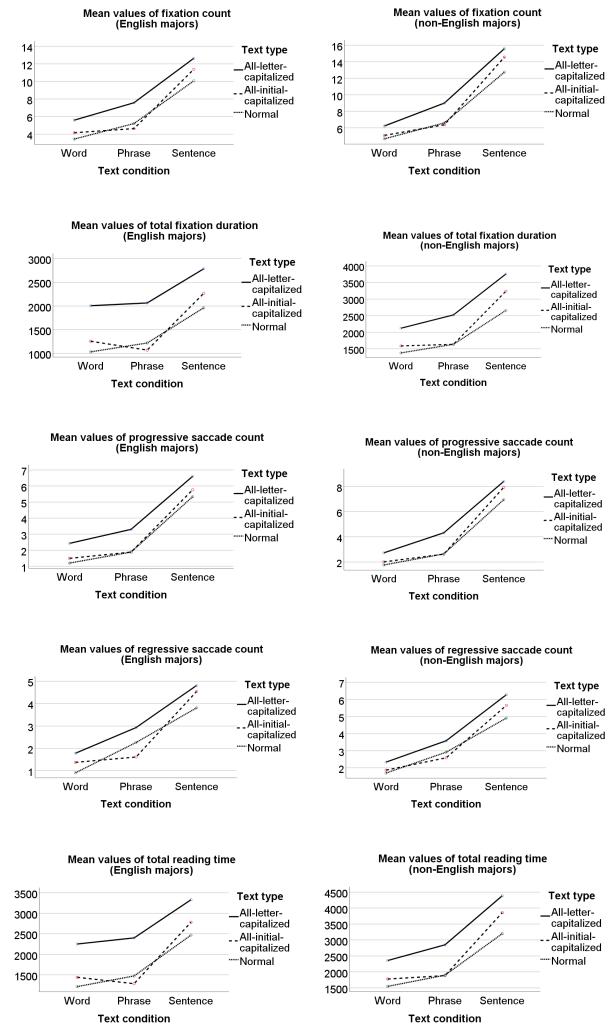


Figure 1. Six measures’ mean values of English majors and non-English majors

There is another significant two-way interaction in ‘Text condition × Subject group’ ($F(12, 3278) = 5.262, p < .001$). Bonferroni adjusted post-hoc tests showed that, as in TABLE III., significant differences were observed in terms of all the six measures across three conditions. In comparison with English majors, non-English majors generate more and longer fixations, more progressive and regressive saccades, shorter saccades and more reading time. This can prove that when reading the same type text, the cognitive load and processing difficulty of English majors were significantly lower than those of non-English majors; while the reading efficiency of English majors were significantly higher than that of non-English majors. It is in line with our expectation, that English majors possess higher language proficiency. However, since no

significant two-way ‘Text type × Subject group’ interaction is found, capitalization was believed to pose a similar degree of challenge to both English and non-English majors. This challenge was great enough to counterbalance the advantages of English majors on ‘Text condition’, so no significant three-way ‘Text condition × Text type × Subject group’ interaction is found.

TABLE III. BONFERRONI ADJUSTED POST-HOC TEST BETWEEN TEXT CONDITION (WORD, PHRASE AND SENTENCE READING) ACROSS SUBJECT GROUPS (ENGLISH AND NON-ENGLISH MAJORS). ITEMS IN BOLD INDICATE SIGNIFICANT FINDINGS.

Measure	Text condition	<i>Mean difference (English vs. non-English)</i>	Std. error	p
FC	word	-0.936	0.252	<.001***
	phrase	-1.526	0.307	<.001***
	sentence	-2.938	0.353	<.001***
TFD (ms)	word	-262.003	74.949	<.001***
	phrase	-478.219	91.200	<.001***
	sentence	-880.291	105.102	<.001***
PSC	word	-0.468	0.138	<.01**
	phrase	-0.845	0.168	<.001***
	sentence	-1.868	0.194	<.001***
RSC	word	-0.613	0.119	<.001***
	phrase	-0.764	0.145	<.001***
	sentence	-1.225	0.167	<.001***
MSA	word	1.069	0.135	<.001***
	phrase	-259.117	85.415	<.01**
	sentence	-490.829	103.934	<.001***
TT	word	-951.997	119.778	<.001***
	phrase	-0.936	0.252	<.001***
	sentence	-1.526	0.307	<.001***

IV. DISCUSSION

A. Influence of capital letters on reading of Chinese learners

Studies on capital letters in English-speaking countries centers around the all-capital type form, which is considered to retard the reading speed of natives. In our study, eye movement measures reflected in the reading process of all-capital texts and normal texts got similar conclusion that all-capital texts also posed challenge for Chinese learners. Influences of capital letters on English reading of Chinese learners include heavier cognitive load, more complicated cognitive processing and lower reading efficiency. Therefore, there is no positive transfer of Chinese learners’ capability of character reading to English all-capital word reading. Similarity in forms won’t reduce the difficulty of recognizing foreign words.

All-initial-capital text, which is also one type of capitalization, has seldom been studied. According to our results, on reading words and phrases, no marked difference has been observed between all-initial-capital text and normal text. Yet on sentence condition, all-initial-capital text reading differentiated itself from normal text reading by presenting heavier cognitive load and lower reading efficiency. In all-initial-capital materials, each word contains one capital letter, each phrase contains 2 to 3 capital letters, and each sentence contains 6 capital letters in average. It could be concluded that if one sentence has more than 6 capitals letters, then the load and complexity in its cognitive processing is very much similar toward those of all-capital letter sentence.

Among the six eye movement measures, FC, TFD, RSC, MSA and TT are used to identify skilled readers [21, 22, 18]. Upon reading same text types, English majors are proved to be more skilled readers who can retrieve and process information in a logic-thinking mode in a quicker way. Nevertheless, when capital letters appeared in the experimental materials, the advantages of English majors on language proficiency and thinking mode were counterbalanced and they presented similar cognitive features as non-English majors.

People tended to remember alternating shapes better than square shapes [8]. It may not suit for Chinese native students. Chinese scripts are pictographic, ideographic and pictophonetic [30]. One character is a combination of form, sound and meaning. Intuition plays a very important role in recognizing and understanding them. Nevertheless, English scripts do not form images but are prescribed abstract signs, so recognition and understanding of them depend largely on logical thinking [30]. The intuitive thinking mode Chinese learners develop when acquiring Chinese language is quite different from the logical thinking mode that is required in perceiving alphabetical languages. We think this can account for the difficulties that Chinese learners experience in English reading.

B. Influence of captital letters on word-superiority effect on Chinese learners

Carroll [31] distinguished three levels of written language processing, i.e., feature, letter and word. Featural and letter information influences higher levels of processing, but the perception of individual letters is facilitated by the presence of a word or a word like context, which is called the word-superiority effect.

All the words in our experimental materials were selected from CET-4 word banks and have a relatively high frequency rate. All participants have learned English for more than 10 years and could provide correct answers for all the comprehension questions in the experiment. That meant the words in our materials were familiar to them. Based on word-superiority effect, the word *thanks* should be perceive as a whole. If we change *thanks* into *Thanks* or *thenks*, it ought to be quickly recognized and won’t hinder the comprehension. However, during the experiment, we noticed that 16 of participants were written something down on the desk by using their finger. They said in the follow-up interview that they were changing the capital letters into lower-case letters. Another 15 participants also said that they needed to

switch capital letters back to lower-case letters in their minds in order to fulfill the reading. That means, for Chinese natives, a change in case of the first letter of a word is very different from that of any in-bedded letter of this word. Even for frequently used words, capital letters should be switched back to lower-cased letters for fulfilling word recognition. This hints a word processing after letter processing for Chinese learners, which contradicts the word-superiority effect. The type form change of a word's first letter restrains the word-superiority effect and controls Chinese student's to follow a bottom-to-up order in perception. If they lack the practice of capital letter reading, especially initial-capital word reading, cognitive load and processing difficulty on letter level will increase and then pass on to sentence level. This can count for the marked difference between reading all-initial-capital sentence and normal sentence.

V. CONCLUSION

This study aims to explore the influence of the capital form on English reading of Chinese learners. We found that capital letters pose challenge on Chinese learners' reading performance by increasing cognitive load, complicate perceptual processing and lowering reading efficiency. Though the rectangular form of all-capital words look like Chinese characters, it won't reduce the level of difficulty in cognitive processing. So the decisive point for correct perception is the link between form and meaning. For initial-capital words, Chinese learners exhibit a letter-to-word processing order, which shows no sign of word-superiority effect. English majors, if they lack capital letter reading practice, they will face the same challenge as non-English majors do on reading capitals despite the fact that they have a relative higher language proficiency and stronger logical thinking ability. Further studies are needed to explore the influence of form change of initial letter on reading of Chinese learners. Also, we need experimental data to study the intuitive thought Chinese student develop in learning Chinese and the logical thought they develop in learning English.

REFERENCES

- [1] M. A. Tinker and D. G. Paterson, "Influence of type form on speed of reading," *Journal of Applied Psychology*, vol. 12, 1928, pp. 359-368, doi: 10.1037/h0073699.
- [2] M. A. Tinker and D. G. Paterson, 1939, "Influence of type form on eye movements," *Journal of Experimental Psychology*, vol. 25, 1939, pp. 528-531, doi: 10.1037/h0063038.
- [3] M. A. Tinker, "The effect of slanted text upon the readability of print," *Journal of Educational Psychology*, vol. 45, 1954, pp. 287-291, doi: 10.1037/h0062825.
- [4] M. A. Tinker, *Bases For Effective Reading*. Minneapolis: University of Minnesota Press, 1965.
- [5] F. Smith, "Familiarity of configuration vs. discriminability of features in the visual identification of words," *Psychonomic Science*, vol. 14, 1969, pp. 261-262, doi: 10.3758/BF03329112.
- [6] D. F. Fisher, "Reading and visual search," *Memory and Cognition*, vol. 3, 1975, pp. 188-196, doi: 10.3758/BF03212897.
- [7] K. Larson, "The legibility of a typeface should not be evaluated on its ability to generate a good word shape," *Eye*, vol. 52, Available at URL: <http://www.eyemagazine.com/opinion/article/the-science-of-word-recognition>.
- [8] N. Pusnik, A. Podlesek and K. Mozina, "Typeface comparison—Does the x-height of lower-case letters increased to the size of upper-case letters speed up recognition?" *International Journal of Industrial Ergonomics*, vol. 54, 2016, pp. 164-169, doi: 10.1016/j.ergon.2016.06.002.
- [9] X. M. Chen and J. Li, X 2016. "The cultivation of reading ability of Chinese college students on all-capital English words (*peiyang daxuesheng zizhu tigao quandaxie yingyu rendu nengli*)," *Jiangsu Foreign Language Teaching and Research*, vol. 1, 2016, pp. 5-8.
- [10] J. H. Dong and L. Pang, "Influence of upper-case and lower-case words on word recognition of Chinese English learners (*yingyu daci daxiaoxie du zhongguo yingyi xuexizhe danxi shibie de yingxiang tantao*)," *Journal of Social Science of Human Medical University*, vol. 5, 2009, pp. 218-219+141.
- [11] X. M. Chen, "Study on the reading abilities of all-capital-letter English of College Students (*gaoxiao daxuesheng quandaxie yingyu rendu yanjiu*)," *Journal of Hunan University of Science and Engineering*, vol. 36, 2015, pp.155-159.
- [12] F. G. Paas, "Training strategies for attaining transfer of problem-solving skill in statistics: A cognitive-load approach," *Journal of Educational Psychology*, vol. 84, 1992, pp. 429-434, doi: 10.1037/0022-0663.84.4.429.
- [13] S. Y. Mousavi, R. Low and J. Sweller, "Reducing cognitive load by mixing auditory and visual presentation modes," *Journal of Educational Psychology*, vol. 87, 1995, pp. 319-334, doi: 10.1037/0022-0663.87.2.319.
- [14] S. Kalyuga, P. Ayres, P. Chandler and J. Sweller, "The expertise reversal effect," *Educational Psychologist*, vol. 38, 2003, pp. 23-31, doi: 10.1207/S15326985EP3801_4.
- [15] K. Rayner and A. Pollatsek, *The Psychology of Reading*. Englewood Cliffs: Prentice Hall, 1989.
- [16] G. L. Yan, et al, "Review of eye-movement measures in reading research (*yuedu zhong de zhuyao yandong zhipiao pingshi*)," *Advances in Psychological Science*, vol. 21, 2013, pp. 589-605, doi: DOI: 10.3724/SP.J.1042.2013.00589.
- [17] J.M. Henderson and F. Ferreira, "Effects of foveal processing difficulty on the perceptual span in reading: Implications for attention and eye movement control," *Journal of Experimental Psychology*, vol. 16, 1990, pp. 417-429, doi: 10.1037/0278-7393.16.3.417.
- [18] K. Rayner, T. J. Slattery and N. N. Bélanger, "Eye movements, the perceptual span, and reading speed," *Psychonomic Bulletin & Review*, vol. 17, 2010, pp. 834-839. doi: 10.3758/PBR.17.6.834.
- [19] M. A. Just and P. A. Carpenter, "A theory of reading: From eye fixations to comprehension," *Psychological Review*, vol. 87, 1980, pp. 329-354, doi: 10.1037/0033-295X.87.4.329.
- [20] K. Rayner, "Eye movements in reading and information processing: 20 years of research," *Psychological Bulletin*, vol. 124, 1998, pp. 372-422, doi: 10.1037/0033-2909.124.3.372.
- [21] J. Ashby, K. Rayner and C. Clifton, "Eye movements of highly skilled and average readers: Differential effects of frequency and predictability," *Quarterly Journal of Experimental Psychology*, vol. 58, 2005, pp. 1065-1086, doi: 10.1080/02724980443000476.
- [22] K. Rayner, "Eye movements and attention in reading, scene perception, and visual search," *Quarterly Journal of Experimental Psychology*, vol. 62, 2009, pp. 1457-1506, doi: 10.1080/17470210902816461.
- [23] D. E. Irwin and L. A. Carlson-Radvansky, "Cognitive suppression during saccadic eye movements," *Psychological Science*, vol. 7, 1996, pp. 83-88, doi: 10.2307/40062915.
- [24] D. E. Irwin, "Lexical processing during saccadic eye movements," *Cognitive Psychology*, vol. 36, 1998, pp. 1-27, doi: 10.1006/cogp.1998.0682.
- [25] K. Rayner, "Eye movements in reading and information processing," *Psychological Bulletin*, vol. 85, 1978, pp. 618-660, doi: 10.1037/0033-2909.85.3.618.
- [26] D. S. Sanders and J. A. Stern, "Effects of text variables on eye movements in reading," *Integrative Psychological and Behavioral Science*, vol. 15, 1980, pp. 30-34, doi: 10.1007/bf03003677.
- [27] S. P. Liversedge and J. M. Findlay, "Saccadic eye movements and cognition," *Trends in Cognitive Sciences*, vol. 4, 2000, pp. 7-14, doi: 10.1016/s1364-6613(99)01418-7.

- [28] E. Meseguer, M. Carreiras and C. Clifton, “Overt reanalysis strategies and eye movements during the reading of mild garden path sentences,” *Memory and Cognition*, vol. 30, 2002, pp. 551-561, doi: 10.3758/bf03194956.
- [29] G. L. Yan and X. J. Bai, “Eye movement research on Chinese reading (*zhongwen yuedu guocheng zhong de yandong yanjiu*),” *Updates in Psychology*, vol. 8, 2000, pp. 19-22.
- [30] S. N. Lian, *Contrast Studies of English and Chinese (yinghan duibi yanjiu)*. Beijing: Higher Education Press, 2010.
- [31] D. W. Carroll, *Psychology of Language*, Fifth Edition. Belmont, CA: Thomson West, 2010.

A Machine Learning Model for the Dating of Ancient Chinese Texts

Xuejin Yu

*University of Science & Technology Beijing
Beijing, China
yuxuejin94@gmail.com*

Wei Huangfu

*University of Science & Technology Beijing
Beijing, China
huangfuwei@ustb.edu.cn*

Abstract—This paper, with the intent of solving the issues on the dating of ancient Chinese texts, takes advantage of the Long-Short Term Memory Network (LSTM) to analyze and process the character sequence in ancient Chinese. In this model, each character is transformed into a high-dimensional vector, and then vectors and the non-linear relationships among them are read and analyzed by LSTM, which finally achieve the dating tags. Experimental results show that the LSTM has a strong ability to date the ancient texts, and the precision reaches about 95% in our experiments. Thus, the proposed model offers an effective method on how to date the ancient Chinese texts. It also inspires us to actively improve the time-consuming analysis tasks in the Chinese NLP field.

Keywords-Ancient Chinese texts; Dating; Machine learning; LSTM

I. INTRODUCTION

Natural Language Processing (NLP) is a cross-disciplinary research field of linguistics, computer science, information engineering, and artificial intelligence. With such technologies, we process and analyze large amounts of natural language data by means of computers. In recent years, Chinese natural language processing, as an important part of NLP, has increasingly attracted attention of academic world, and made gratifying achievements. Ancient Chinese texts, carrying philosophy, culture, knowledge, wisdom and spirit of the Chinese people during thousands of years, are usually referred to as the Pearl of the crown of Chinese language. However, there are many essential issues to be solved, ranging from digitization, labeling, to categorization for such ancient Chinese texts. Thus we need to use advanced Chinese natural language processing technologies to further process and perform data mining. It will be beneficial to the preservation and knowledge extraction of existing ancient Chinese texts, and greatly promotes the construction of digital humanities in China.

The current progress of digitization of ancient Chinese texts mostly remains in the early stage. There are short of studies on part-of-speech tagging, named entity recognition, text structure processing, text classification, etc. Moreover, the accuracy of the existing methods on part-of-speech tagging is not enough since the Chinese ancient texts can be traced back to the Shang dynasty (at least more than 3,000 years ago). In the evolution of these years, the meaning, the grammar and the syntax of the Chinese language are constantly changing.

Take “湯” (tang in pinyin), an ancient Chinese character, as an example. It originally means “hot water”, but now

only refers to the soup obtained after cooking the food. Therefore, even a character in different periods possibly has different meanings.

The Chinese language has a long history, which leads to great difficulties in constructing a suitable model for various periods. Therefore, only by judging the approximate time of the ancient texts, can we conduct subsequent studies on ancient texts and further improve the accuracy and efficiency of the language research.

In this paper, we attempt to explore the field of NLP for the ancient Chinese texts from the perspective of the text dating. We focus on the problem of the dating of ancient Chinese texts with deep learning network model. With our approach, there is no need to manually extract rule features. The research results of this paper will help to study the ancient Chinese word segmentation, part-of-speech tagging, text structure processing, text classification and so forth.

II. RELATED WORK

From a technical point of view, the time judgment of ancient text is a typical text classification task.

Currently, text classification methods can be roughly categorized into two categories: the former is the traditional machine learning methods based on rules or probability, and the latter is the deep learning methods based on Convolutional Neural Networks (CNN) [1], Recurrent Neural Networks (RNN) [2] or self-Attention [3].

The rule-based and the probability-based approaches are relatively simple, easy to implement, and work well in many specific areas. However, in the rule-based and the probability-based methods, many rules or specific conditions need to be considered, so it is necessary to define and manually extract features by experts.

Recently, deep learning algorithms are widely used to solve the language processing problems. Collobert [4] and Tang [5] apply CNN and RNN to process natural language, respectively. The Bidirectional Encoder Representations from Transformers (BERT) model is proposed based on self-attention mechanism in 2018. It performs very well in lots of kinds of typical NLP tasks. However, the BERT model is mainly designed for modern languages, and depends on the massive informational texts of the current Internet era, such as Wiki encyclopedia entries, various news media, and commentary messages. However, it is not feasible in the ancient Chinese research field where the corpus resources are relatively limited.

III. ANCIENT CHINESE TEXT DATING MODEL

We try to model the problem of time determination of ancient Chinese texts as follows:

$$T = M(\{v_1, v_2, \dots, v_n\}) = M[g(\{x_1, x_2, \dots, x_n\})] \quad (1)$$

where $g(\cdot)$ is a mapping function, $\{x_1, x_2, \dots, x_n\}$ is a piece of Chinese texts to be determined, and $\{v_1, v_2, \dots, v_n\}$ denotes its vector sequence [6]. Then the vector sequence is processed by the model M , which calculates the outputs label T .

In this paper, we use Google's word2vec [7] model to obtain word vectors. Assumed that if the contexts of two words are similar, then their semantics are similar in such a model. The word2vec, an unsupervised model, is able to obtain the result in a relatively short period at lower cost. We can learn vector representation of the word in a large number of unlabeled corpora through the continuous bag of words (CBOW) model of word2vec. The input of the CBOW model is the word vectors of words in the context of the central word, and the output is the word vector of the given word. The output vector contains the relationships between words. After getting the output vector with the relationship, we use it as a feature for next steps to improve the generalization ability of the model.

The neural network that is commonly seen is like a spider web that converges from many nodes to a single output. Here we have a single input and a single output. Such a network works well for non-continuous inputs, where the order of the inputs does not affect the output.

In the text processing, the order of the characters is very important. The RNN can accept continuous input, using the activation of the previous node as the parameter of the latter node [8]. However, RNNs are not very good at passing information from very early units. The Long-Short Term Memory Network (LSTM) uses a memory unit to store certain information that occurred before [9]. A memory unit in LSTM includes input gates, output gates, and forgetting gates to control the preservation of information, which makes LSTM better in the language model.

A. Network Structure

The block diagram of the proposed ancient Chinese dating model is shown in Fig. 1.

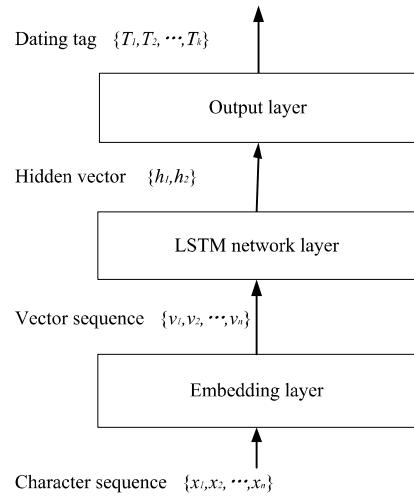


Figure 1. The block diagram of the ancient Chinese dating model.

As shown in Fig. 1, the model accept a piece of text as its input and predict its occurrence time as its output. First, a piece of text to be predicted is obtained, and the text is sent to the embedding layer to obtain a vector representation of the text. Then the character vectors are sent to the LSTM neural network layer to calculate the hidden vector. Finally, a fully connected network calculates the hidden vector and outputs the time stamp labels.

The description of the model in detail is shown in Fig. 2. The model consists of 3 layers, which are the embedding layer, the LSTM forward and backward layer(s), and the output layer. The embedding layer is a CBOW model of word2vec, which represents Chinese characters as a sequence of vectors. Then there are the bidirectional-LSTM [10], [11] layers, which are the main body of the model. Their forward and reverse inputs are the vector sequences in different orders, respectively. Both the forward and reverse parts of the LSTM network output part of the hidden vectors. The output layer concatenates the two hidden vectors and sends them to a fully connected layer. Finally, the different dating tags are determined by the probability with a Softmax operation. It is noting that the hidden vector is changed while the word vector is fed to LSTM since the LSTM networks are for time series.

B. LSTM Memory Unit

LSTM exploits long-term dependencies by using input gates, input gates, and forgetting gates. The typical schematic of LSTM memory unit is shown as Fig. 3

Fig. 3 shows the LSTM neural network expanding in time dimension. The formulae of the LSTM memory unit are as follows:

$$f^{(t)} = \sigma(W_{fv}v^{(t)} + W_{fh}h^{(t-1)} + W_{fc}c^{(t-1)}) \quad (2)$$

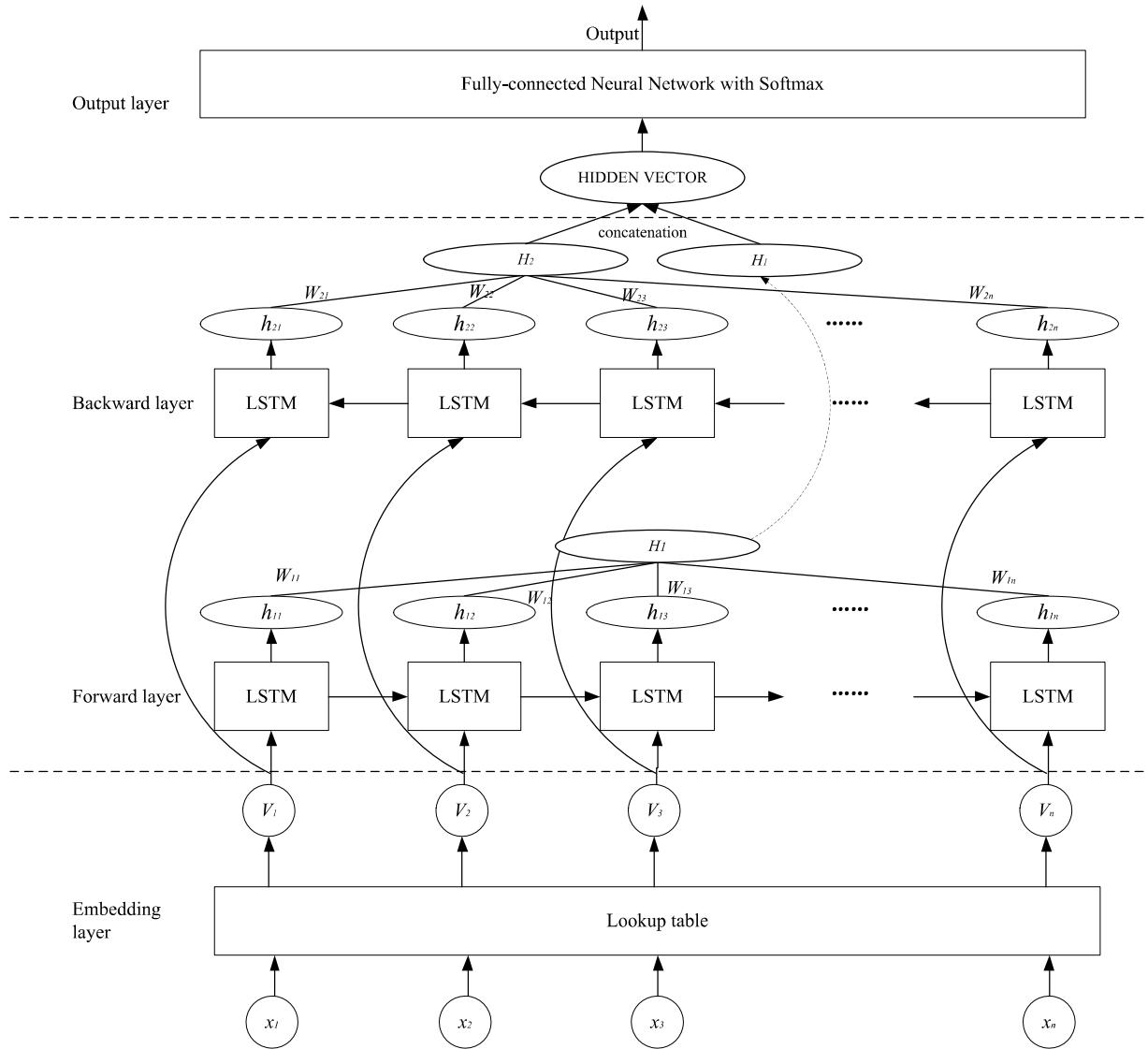


Figure 2. The Bi-LSTM structure diagram.

$$i^{(t)} = \sigma(W_{iv}v^{(t)} + W_{ih}h^{(t-1)} + W_{ic}c^{(t-1)}) \quad (3)$$

$$a^{(t)} = \Phi(W_{cv}v^{(t)} + W_{ch}h^{(t-1)}) \quad (4)$$

$$c^{(t)} = f^{(t)} \cdot c^{(t-1)} + a^{(t)} \cdot i^{(t)} \quad (5)$$

$$o^{(t)} = \sigma(W_{ov}v^{(t)} + W_{oh}h^{(t-1)} + W_{oc}c^{(t)}) \quad (6)$$

$$h^{(t)} = o^{(t)} \cdot \Phi(c^{(t)}) \quad (7)$$

In these formulae, $i^{(t)}$ is the input gate, σ is the sigmoid function, and the function of the sigmoid function is to make each element of the output vector valued in the

interval $[0, 1]$. Also, Φ is a tanh function, which is used to make each element of the output vector between $[1, -1]$; $f^{(t)}$ is the forgetting gate; $c^{(t)}$ is used to store the long-term information, which is obtained by adding the product of the information; $c^{(t-1)}$ is for the last moment; the forgetting gate $f^{(t)}$ is used to the product of the current input state $a^{(t)}$ and the input gate $i^{(t)}$; $o^{(t)}$ is the output gate to control output vectors; $h^{(t)}$ is the output of the current moment, which is obtained by multiplying the output gate $c^{(t)}$ by the current state of the information $\Phi(c^{(t)})$.

C. Output Layer

The output layer is a fully connected network. It accepts a vector h of $1 \times 2n$ dimensions, which is generated by the LSTM layer, while n is the dimensions of hidden vector. The output layer has a weight w vector of $2n \times k$

Table I
DEFINITION OF LABELS

Tag	Year	Dynasty
T1	Before 475 B.C.	the Spring and Autumn Period
T2	475 B.C.-350 B.C.	Early Warring States Period
T3	350 B.C.-221 B.C.	Late Warring States Period

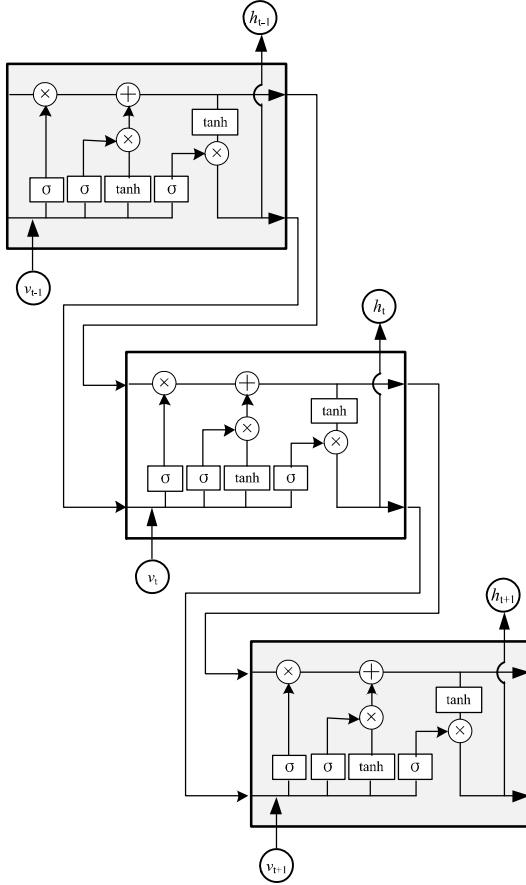


Figure 3. The LSTM memory unit structure.

dimensions and an offset b vector of $1 \times k$ dimensions, where k is the number of the total classification tags. A $1 \times k$ dimensional vector output is obtained by the formula $y = Wx + b$. The value of each of these dimensions can be seen as the likelihood of the kind represented by that dimension.

The Softmax function is used to normalize the vector elements of the output layer output, and the normalized probability is output. Here the Softmax function is given by

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_k e^{z_k}}. \quad (8)$$

IV. EXPERIMENTS AND DISCUSSION

The corpus used in the experiments is some ancient texts of the Spring and Autumn period and the Warring States period. We divided them into three small time periods: the Spring and Autumn Period, Early Warring States Period and Late Warring States Period, as shown in Table I. We use the symbols T_1, T_2, T_3 , to represent the three periods, respectively.

In each time period, we selected some ancient texts as the source of the training data. Totally 8 ancient works

Table II
THE SOURCE OF THE TRAINING DATA.

Tag	Book Name	Number of characters ($\times 1000$)
T1	<i>Shangshu</i>	45
T1	<i>Chunqiu</i>	32
T1	<i>Yili</i>	107
T1	<i>Zhouli</i>	90
T2	<i>Zhouyi</i>	5
T2	<i>Mu Tianzi Biograph</i>	17
T3	<i>Zuozhuan</i>	381
T3	<i>Guoyu</i>	130

are loaded as shown in the following Table II. It covers totally about 800,000 Chinese characters.

Our training procedure is shown in Fig. 4.

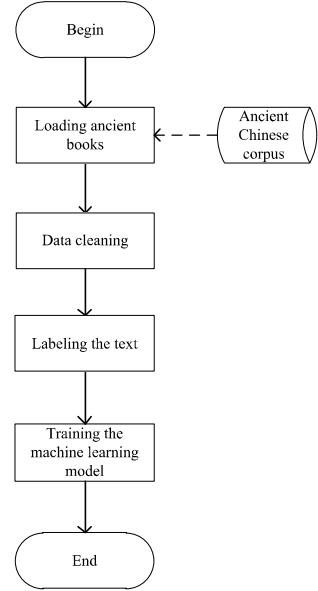


Figure 4. The schematic diagram of the training process.

A. Parameter selection

We conduct experiments to compare different hyper parameters. Different combinations of the parameters for the Hidden Layer Dimension (HLD) and the word Embedding Layer Dimension (ELD) are considered. We compare the accuracy of the training process under different parameter combinations. Fig. 5 shows the classification accuracy curves of the training process. It means that the convergence is the fastest when the HLD is 64 and the ELD is 64.

We also compare the accuracy indexes of the proposed model on the test data under different parameter combi-

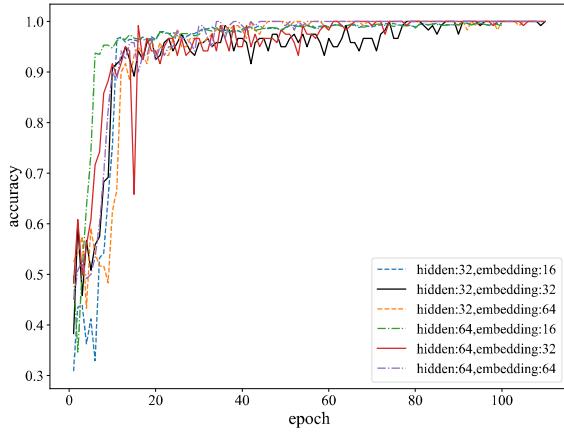


Figure 5. The classification accuracy curve of the training process.

nations. The experimental results are shown in Table III. Note that the following experiments are assigned to a hidden layer dimension of 32 and a word embedding vector dimension of 32.

Table III
THE ACCURACY ON THE TEST DATA.

HLD	ELD	Accuracy
32	16	0.971
32	32	0.987
32	64	0.960
64	16	0.975
64	32	0.962
64	64	0.975

B. Experiments

In the first experiment, we select some chapters which are not fed into the training model for each book. After the training procedure, we feed those non-training sentences to the network to judge their tags. Each line in the table below indicates the number of sentences in a given period that are judged to various tags. Table IV shows the results. In this case, the probability that a sentence is judged to be the correct tags is large, which means the proposed model performs well in the text dating.

Table IV
THE RESULT OF EXPERIMENT ONE.

Input \ Output	T1	T2	T3	Total
T1	1148(76%)	142(10%)	214(14%)	1504(100%)
T2	123(9%)	1092(85%)	75(6%)	1290(100%)
T3	47(4%)	10(1%)	1089(95%)	1146(100%)

We also try to test a book not from the same source. We use the rest of the books labeled with the same tag as the training set to observe the classification effect of the model on the whole texts of the non-training book. Since the proposed model takes text paragraphs as its input, we divide a book into many paragraphs. Then we judge the dating results of these paragraphs.

We conducted the experiment with “Zuo Zhuan” as an example. The judgment result is as Fig. 6. It can be seen that 994 pieces of the texts in “Zuo Zhuan” are labelled T1, 529 are labelled the T2, and 2132 are labelled T3. On the whole, the proposed model judges that the books “Zuo Zhuan” should be labelled with T3. This is consistent with our knowledge in the ancient Chinese community and it also confirms the correctness of our model.

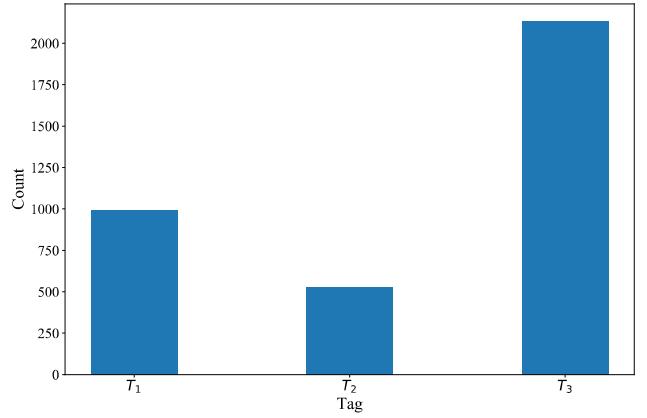


Figure 6. The bar chart of “Zuo Zhuan” dating results.

The experimental results show that if some texts of an ancient book are selected in the training set and the rest are used for the measurement set, the experimental correct rate of the proposed model is about 95%. However, if the ancient books are not involved in the training set, the correct rate of the paragraphs of the ancient books will be reduced. The proposed model also correctly labels the book if all the paragraphs in this book is fed into the model.

V. CONCLUSION AND FUTURE WORK

In this paper, we have utilized the LSTM to date the ancient Chinese texts. LSTM has great ability to analyze and recognize natural language, and we even show LSTM is outstanding in ancient Chinese processing. Our experiment results show that the precision of the LSTM reaches 95% in the dating of ancient Chinese books. Thus, the proposed model offers an effective method on how to date the ancient Chinese texts. It also inspires us to actively improve the time-consuming analysis tasks in the Chinese NLP field.

It is worth noting that the model we proposed is only a small exploration in the field of the NLP technologies related to ancient Chinese studies. There are still many shortcomings that need to be further improved in the future, including but not limited to, increasing the size of the data set, improving the word vector embedding and conducting more extensive experiments.

ACKNOWLEDGE

This research was funded by the Social Science Foundation Project of Beijing (Grant No. 18YYB003). The corresponding author is Dr. Wei Huangfu.

REFERENCES

- [1] J. L. Elman, “Finding structure in time,” *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.
- [2] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [4] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural language processing (almost) from scratch,” *Journal of machine learning research*, vol. 12, no. Aug, pp. 2493–2537, 2011.
- [5] D. Tang, B. Qin, and T. Liu, “Document modeling with gated recurrent neural network for sentiment classification,” in *Proc. EMNLP*, Lisbon, Portugal, Sep. 2015, pp. 1422–1432.
- [6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv preprint arXiv:1301.3781*, 2013.
- [7] O. Levy and Y. Goldberg, “Neural word embedding as implicit matrix factorization,” in *Advances in neural information processing systems*, 2014, pp. 2177–2185.
- [8] T. Mikolov, M. Karafiat, L. Burget, J. Černocký, and S. Khudanpur, “Recurrent neural network based language model,” in *Proc. INTERSPEECH 2010*, Lyon, France, 2010.
- [9] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [10] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional lstm and other neural network architectures,” *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.
- [11] C. Dong, J. Zhang, C. Zong, M. Hattori, and H. Di, “Character-based lstm-crf with radical-level features for chinese named entity recognition,” in *Natural Language Understanding and Intelligent Applications*. Springer, 2016, pp. 239–250.

Correlational Neural Network Based Feature Adaptation in L2 Mispronunciation Detection

Wenwei Dong and Yanlu Xie

Beijing Advanced Innovation Center for Language Resources

Beijing Language and Culture University

Beijing, China

E-mail: dongwenwei_blcu@163.com, xieyanlu@blcu.edu.cn

Abstract—Due to the difficulties of collecting and annotating second language (L2) learner’s speech corpus in Computer-Assisted Pronunciation Training (CAPT), traditional mispronunciation detection framework is similar to ASR, it uses speech corpus of native speaker to train neural networks and then the framework is used to evaluate non-native speaker’s pronunciation. Therefore there is a mismatch between them in channels, reading style, and speakers. In order to reduce this influence, this paper proposes a feature adaptation method using Correlational Neural Network (CorrNet). Before training the acoustic model, we use a few unannotated non-native data to adapt the native acoustic feature. The mispronunciation detection accuracy of CorrNet based method has improved 3.19% over unnormalized Fbank feature and 1.74% over bottleneck feature in Japanese speaking Chinese corpus. The results show the effectiveness of the method.

Keywords-Computer-Assisted Pronunciation Training; Correlational Neural Network; Bottleneck feature

I. INTRODUCTION

With the accelerating of globalization, language learning becomes more and more important. CAPT provides an opportunity for learners to practice their pronunciation and give them feedbacks free from time and space limitations, and its main framework is similar to Automatic Speech Recognition (ASR). Because of the difficulty of annotating non-native data, acoustic model of mispronunciation detection is usually trained with native speaker’s corpus and using it to evaluate L2 learners’ pronunciation, thus there is a mismatch between them. Adding L2 learners’ corpus in the training set can help mitigate the mismatch to some extent [1]. To further reduce the impact of it, the speaker and environment adaptation techniques are explored.

The adaptation techniques can be divided into model adaptation that tune-up the acoustic model parameters to fit the test set and feature adaptation that tune-up the feature before training acoustic model to make it more fitted for the test set. Most of model adaptation techniques are used in GMM acoustic models, such as maximum likelihood linear regression (MLLR) [2] and Maximum A Posterior (MAP), but they cannot be applied to DNN acoustic model which have a number of parameters that cannot be adapted with such small data. Many feature adaptation methods are used in DNN acoustic model.

Feature-space MLLR (fMLLR), Vocal Tract Length Normalization (VTLN) and i-vector are effective methods for reducing the mismatch of native and non-native speech [2, 3], Zhang [4] and Huang [5] used selective maximum likelihood

linear regression (SMLLR) and MLLR to reduce the speaker difference, Luo [6] compared feature-space MLLR (fMLLR) and factorized fMLLR which factorizing the speaker and environment and other acoustic factors, factorized method performs better in CAPT tasks. Besides, some researchers used deep neural networks as a feature extractor to reduce the influence of mismatch, it can effectively model and feature represent. Gao [7] and Nicolao [8] used bottleneck feature that extracting from a neural network as the input of classifier to improve detection result. And a lot of works tried to add a rescore or verification process to get further improvements [9–13]. Most of them are use single view neural network as a feature extractor.

With the great progress of neural network in various fields, researchers work on changing the model structure to get used to different tasks. Chandar [14] proposed CorrNet, it’s a multi-views model based Common Representation Learning (CRL). CorrNet try to embed different descriptions (views) of data in a common subspace and maximize the relation of them. In this paper, we take CorrNet as feature extractor to adapt feature and reduce the mismatch, CorrNet is trained with a few L2 and L1 speech data, the model can learn the relationship between them. The paper is organized as follows: Section 2 presents two way of improving GOP measures and the frameworks of adding CorrNet. Section 3 introduces the experiment corpus and setup. Section 4 shows the experiment results and discussions, and the conclusions are drawn in Section 5.

II. THE FRAMEWORK OF MISPRONUNCIATION DETECTION

In this section, we introduce the CorrNet and the new framework of generating posterior probability, then briefly review the traditional GOP method.

A. CorrNet Framework

Common Representation Learning (CRL) focus on embedding different description (or views) of the data into a common subspace, two popular paradigms are Canonical Correlation Analysis (CCA) based approaches and Autoencoder (AE) based approaches. CorrNet is a kind of AE approach. It’s a multi-view model and aims to learn a common representation from two views of data. We use its loss function as follows:

$$J_Z(\theta) = \sum_{i=1}^N L(z_i, g(h(x_i))) + L(z_i, g(h(y_i))) - \lambda \text{corr}(h(X), h(Y)) \quad (1)$$

where L is reconstruction error,

$$\text{corr}(h(X), h(Y)) = \frac{\sum_{i=1}^N (h(x_i) - \bar{h}(X))(h(y_i) - \bar{h}(Y))}{\sqrt{\sum_{i=1}^N (h(x_i) - \bar{h}(X))^2} \sum_{i=1}^N (h(y_i) - \bar{h}(Y))^2} \quad (2)$$

so we can use CorrNet in our task to benefit GOP in the following aspects:

- Minimize the self-reconstruction error of ASR result.
- Minimize the cross-reconstruction error of native and L2 Fbank.
- Maximize the correlation between the hidden representations of both view's Fbank.

Compared with other multi-task models, CorrNet tries to maximize the correlation of each view and use transfer learning to construct each other. In our task, acoustic model that trained with native speech will perform better in test set of native speech than non-native speech, it is a result of Fbank feature not just containing phone information, but also the speaker and channel et al, and it will affect the accuracy of mispronunciation detection. CorrNet has two input layers, common hidden layer, and two output layers, we can use native speech and L2 speech Fbank as two views to train model. The CorrNet can not only optimize the phone ASR result of each view but also learn the relations of them, the common layer can embed the feature of two kind databases into a common subspace, and it can maximize the phone information and reduce other differences to some extent. The main purpose of this task is detecting mispronunciation, the two output layers can still keep their difference in phone.

In my method, we use CorrNet as feature extractor and combine CorrNet feature and Fbank feature to train acoustic model, then use acoustic model's output of each phone to calculate the score, the framework shows in figure 1,

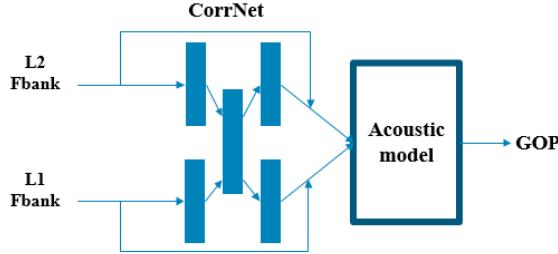


Figure 1. The framework of CorrNet based GOP

B. Bottleneck Framework

Bottleneck feature (BNF) in my experiment use TDNN as feature extractor, we set a bottleneck layer, and the nodes of this layer are less than other hidden layers. Aim to map a high dimensional vector into a low dimensional space with rich information, it also can reduce the mismatch of the acoustic feature, we combine BNF and Fbank to train acoustic model for mispronunciation detection.

We use different feature extractor to compare the ability of reducing native and non-native databases mismatch. The BNF framework is shown in figure 2.

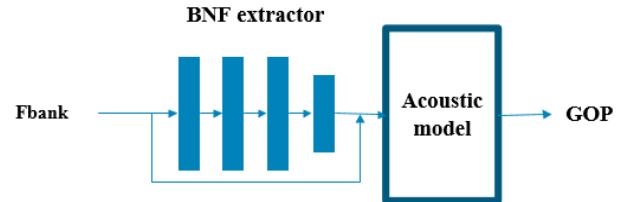


Figure 2. The framework of BNF based GOP

C. GOP Computing Methods

Witt et al proposed GOP in phone level scoring, it's text-dependent, for the acoustic segment X, target phone p,

$$GOP(p) = \frac{1}{d} \log \frac{P(X|p)P(p)}{\sum_{q \in Q} P(X|q)P(q)} \quad (3)$$

where d is the number of frames, $P(X|p)$ is the likelihood of X corresponding to each phone q. Q is the set of phones. In DNN-HMM based system, assuming all phone's prior probability are equal and the sum of all phone posterior probability can be approximated by its maximum [13], in this paper we use the GOP,

$$GOP(p) = \frac{1}{d} \log \frac{P(p|X)}{\max_{q \in Q} P(q|X) + P(p|X)} \quad (4)$$

$P(p|X)$ is the phone p's posterior probability of acoustic model, then we set a threshold to make the final decision,

$$GOP(p) > k \begin{cases} \text{yes, correct pronunciation} \\ \text{no, mispronunciation} \end{cases} \quad (5)$$

those methods can be affected by alignment result of a model. In practice, we can adjust the threshold for language learners with different level. The main idea of GOP is using classifier's confidence score as the score of pronunciation.

III. EXPERIMENTS

A. Speech Corpus

Experiment corpus consists of two parts, the native speech database is provided by the Chinese National Hi-Tech Project 863 for Mandarin continuous speech recognition of large vocabulary system development [15]. It contains 166 speakers about 100 hours. We divide it into the training set about 70 hours and development set about 30 hours, and no speakers overlap. We also use 3600 sentences of native Chinese corpus to test GOP algorithm. The non-native speech corpus is BLCU inter-Chinese speech corpus [16]. It has 19 speakers and each speaker has 301 sentences. We add 12 speakers of it as the training set to reduce the mismatch between native and non-

native dataset, and 7 speakers corpus as the test set. The test set has been annotated at the phone level. The details are shown in Table 1.

TABLE I. JAPANESE L2 INTER-CHINESE CORPUS

Corpus	Description
Text	Conversational Chinese 301
Speaker	7 females
Number of utterances	1899
Number of phones	26431
Average length per utterance	14
Number of annotators	6

B. Evaluation Metrics

There are 4 evaluation indicators:

- False Acceptance Rate (FAR): the percentage of mispronunciation phones that are accepted as correct.
- False Rejection Rate (FRR): the percentage of correctly pronounced phones that are rejected as mispronunciation.
- Diagnostic Accuracy (DA): the percentage of correctly detected.
- The Detection Cost Function (DCF):

$$DCF(\tau) = C_{MISS} FRR(\tau) P_{Target} + C_{FA} FAR(\tau) (1 - P_{Target}) \quad (6)$$

where τ is the threshold of GOP, C_{MISS} is the cost of false rejection, C_{FA} is the cost of false acceptance. P_{target} is a prior probability and in practical application, FAR is more important than FRR, because if too many correct pronunciations are rejected as mispronunciation, it will give users a bad experience.

C. Experiment Setup

Tensorflow toolkit was used to design CorrNet feature extractor, the input feature is Fbank applying CMVN with 10 frames context, a total of 11 frames as an input feature. Common and output layer's nodes are 50, input layer and one hidden layer nodes are 500, 300. The BNF extractor trained with Kaldi, and its input feature same with CorrNet. BNF extractor has 6 layers, each layer has 625 nodes. The last hidden is used as bottleneck layer, and it has 27 nodes.

Kaldi toolkit was used to train Gaussian Mixture Modeling (GMM), Hidden Markov Modeling (HMM) and TDNN, TDNN have 6 hidden layers and each layer have 850 nodes. The alignments generated by GMM-HMM model. Input feature is Fbank, CorrNet, and BNF.

IV. RESULTS AND DISCUSSIONS

A. The DA of Native and Non-native Corpus

We use native and some non-native corpus as training set to test the performance on both native and non-native set. The X-axis of following pictures means different thresholds of GOP.

The Y-axis means the accuracy rate of mispronunciation detection. From figure 3, the result shows adding non-native data to training set still have a certain mismatch. In order to further reduce the mismatch, we conducted a comparative experiment with different features.

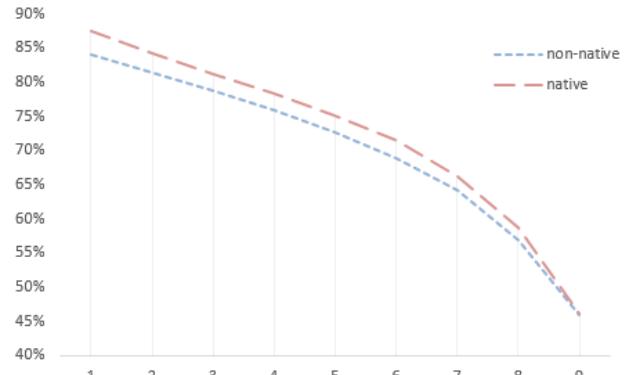


Figure 3. The DA of different dataset

B. Different Feature for TDNN Acoustic Model

We train four kinds of TDNN-HMM acoustic model with Fbank (baseline system), bottleneck (BNF), ivector and CorrNet feature, and use the way to calculate GOP as (4), the DA shown in Figure 4.

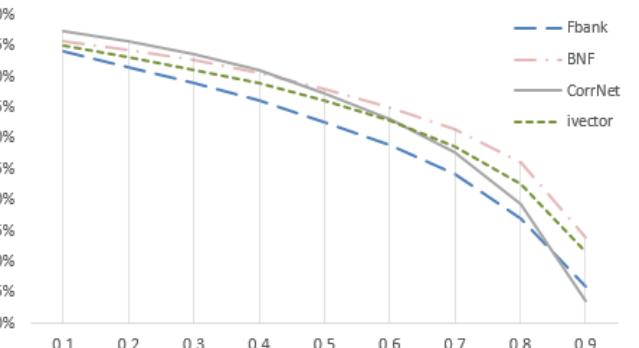


Figure 4. The DA of different feature

BNF is commonly used in language recognition and mispronunciation detection, this method does not consider the relations between two kind databases. In general, BNF and CorrNet features are better than baseline Fbank in all threshold. CorrNet is better than BNF for the threshold of less than 0.45. When the threshold is 0.1, CorrNet DA is 87.31%, BNF DA is 85.57% and Fbank DA is 84.12%. the ivector feature is mostly used in the speaker recognition task, it in mispronunciation detection task can be used to reduce speaker mismatch. When we want to evaluate the pronunciation of non-native speaker, we can use the most similar native speaker as the evaluation criteria. The DA of ivector is 85.04%. The result shows both CorrNet and BNF can reduce the mismatch of two kinds of corpus. The deep neural network as feature extractor can further reduce it than ivector. For the beginning learners of L2

Chinese, setting the threshold a little smaller is good to their language learning, so CorrNet method is more suit for them.

The ROC curves is shown in figure 5.

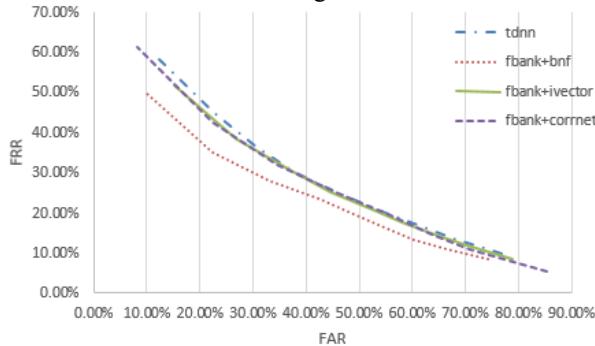


Figure 5. The ROC of different feature

The BNF feature extractor is trained with all native and about 2 hours non-native corpus. And due to the limitations of the model structure, the CorrNet only trained with 2 hours native and 2 hours non-native corpus. The performance of BNF is better than CorrNet.

C. DCF of Different Methods

The total score is 1, we set $\tau = 0.5$ for CorrNet and 0.6 for other methods. And $p_{\text{target}} = 0.8$. FRR is the percentage of mistake making by model. If too many correct pronunciations are judged by the system as errors will affect the user's belief in the system. Thus, In practice, we more care about FRR. Table 2 shows the DCF of different methods,

TABLE II. THE DCF OF DIFFERENT METHODS

Methods	DCF (%)
Fbank	31.83
BNF	27.14
CorrNet	26.71
Ivector	29.17

The DCF of BNF and CorrNet are close, however, CorrNet's DA is 77.3%, and it is higher than BNF 74.98%. The Ivector method can help to reduce the mismatch to some extent, and the DA of it is 72.94% in $\tau = 0.6$.

From the experiment results, both BNF and CorrNet methods can reduce the influence of mismatch. The thresholds mean the degree of rigors. The higher the threshold, the stricter it is and CorrNet adaption method is more suit for the beginner of language learners. Due to the data limitation, we only use about 2 hours non-native data to adapt native data, the CorrNet requires the same number of frames per view, so if more non-naïve data can be used in training set, the adaptation result could be better.

V. CONCLUSIONS

This paper proposed use different methods to reduce the database mismatch for mispronunciation detection task. Adding non-native data to training set is a common way, but the experiment result need have a further improvement, so we try to use CorrNet to map the L2 and native acoustic feature

into a common subspace. Reduce the influence of channel, speaker and others and maximize the relationship between two kinds of corpora. Bottleneck feature was proposed to do the same thing, but in the training process, BNF's model all the data share the weights, it did not consider the difference and relations of input feature. Experiments result show CorrNet as feature extractor are outperform than TDNN extractor. We also compare the ivector feature, the experiment result shows neutral network methods are better than it. In this experiment, because of the limitation of model structure and experimental corpus, we only use 4 hours training corpus that includes native and non-native datasets. In the future, more unlabeled non-native corpora will be collected and participate in training CorrNet.

ACKNOWLEDGMENT

This work is supported by National social Science foundation of China (18BYY124), Advanced Innovation Center for Language Resource and Intelligence (KYR17005), the Fundamental Research Funds for the Central Universities (18YJ030004), the Graduate Innovation Fund of Beijing Language and Culture University (19YCX130), and the project of "Intelligent Speech technology International Exchange". Yanlu Xie is the corresponding author

REFERENCES

- [1] R. Tong, B. P. Lim, N. F. Chen, B. Ma, and H. Li, "Subspace Gaussian Mixture Model for Computer-Assisted Language Learning", in ICASSP, pp. 5347–5351, 2014.
- [2] H. Huang, H. Xu, Y. Hu, et al. "A transfer learning approach to goodness of pronunciation based automatic mispronunciation detection", Journal of the Acoustical Society of America, 142(5):3165, 2017.
- [3] Gales M J F. Maximum likelihood linear transformations for HMM-based speech recognition. Computer Speech and Language, 1998, 12(2):75-98.
- [4] C. Huang, F. Zhang, F K. Soong, et al. "Mispronunciation detection for Mandarin Chinese", IEEE International Conference on Acoustics. IEEE, 2008.
- [5] G. Huang, J. Ye, Z. Sun, Y. Zhou, Y. Shen and R. Mo, "English mispronunciation detection based on improved GOP methods for Chinese students", 2017 International Conference on Progress in Informatics and Computing (PIC), Nanjing , pp. 425-429, 2017.
- [6] D. Luo, C. Zhang, L. Xia and L. Wang, "Factorized Deep Neural Network Adaptation for Automatic Scoring of L2 Speech in English Speaking Tests", in INTERSPEECH, 2018.
- [7] Y. Gao, Y. Xie, W. Cao, and J. Zhang, "A study on robust detection of pronunciation erroneous tendency based on deep neural network," in INTERSPEECH, pp. 693–696, 2015.
- [8] Nicolao, Mauro, A. V. Beeston, and T. Hain. "Automatic assessment of English learner pronunciation using discriminative classifiers", IEEE International Conference on Acoustics, IEEE, 2015.
- [9] W. Li, K. Li, S. M. Siniscalchi, N. F. Chen, and C. Lee, "Detecting mispronunciations of 12 learners and providing corrective feedback using knowledge-guided and data-driven based decision trees," in INTERSPEECH, 2016.
- [10] W. Li, S. M. Siniscalchi, N. F. Chen, and C. Lee, "Improving Non-Native Mispronunciation Detection and Enriching

- Diagnostic Feedback with DNN-Based Speech Attribute Modeling," in ICASSP, pp. 6135–6139, 2016.
- [11] Kim, Yoon, H. Franco, and L. Neumeyer. "Automatic Pronunciation Scoring of Specific Phone Segments for Language Instruction", European Conference on Speech Communication & Technology DBLP, 1997.
- [12] S. M. Witt, and S. J. Young. "Phone-level pronunciation scoring and assessment for interactive language learning", Speech Communication, vol. 30, no. 2, pp. 95-108, 2000.
- [13] W. Hu, Y. Qian, F. K. Soong, and Y. Wang, "Improved mispronunciation detection with deep neural network trained acoustic models and transfer learning based logistic regression classifiers," Speech Communication, vol. 67, pp. 154–166, 2015.
- [14] S. Chandar, M. M. Khapra, H. Larochelle, and B. Ravindran, "Correlational Neural Networks." Neural Computation, vol. 28 pp. 257–285, 2016.
- [15] S. Gao, et al. "Update Progress Of Sinohear: Advanced Mandarin LVCSR System At NLPR." In proc. ICSLP, 2000.
- [16] W. Cao, et al. "Developing a Chinese L2 speech database of Japanese learners with narrow-phonetic labels for computer assisted pronunciation training", in INTERSPEECH, 2010.

Acoustic Cues for Sensory Modality and Referential Vagueness of Personal Pronouns: Me or Not? You or Not?

1st Aibao Zhou

School of Psychology
Northwest Normal University
Lan Zhou, China
zhoulabpapers@163.com

3rd Xiaoyong Lu

School of Psychology
Northwest Normal University
Lan Zhou, China
luxy@nwnu.edu.cn

2nd Yanbing Hu

School of Psychology
Northwest Normal University
Lan Zhou, China
hybpsy2018@163.com

4th Tao Pan

Department of information engineering
Lanzhou Resources and Environment Voc-Tech College
Lan Zhou, China
pant_rev@126.com

Abstract-Personal pronouns are of great pragmatic significance, and among their numerous functions, referential vagueness is the key to the exploration of the self. In Experiment 1, the relationship of first-, second-, and third-person personal pronouns with the self was discussed in acoustic condition, with gender consistency, pronoun type, and person type as independent variables. Experiment 2 records the frequency of the subjects on the SR button and the NSR button, and statistical analysis was performed on the SR button reaction. This study finds that the same pronouns show different self-cognitive processing under the different visual and acoustic stimuli, and the results support the dynamics of personal pronouns and possessive pronouns and self-relationships.

Keywords-personal pronouns; self-cognitive; sensory modality; acoustic cues

I. INTRODUCTION

As James (1890) put it, self is the center of the personal psychic universe. Self is the core of consciousness, different from non-self, and there is a self-referential effect of processing, that is, people preferentially process their own related stimuli [1]. Researchers often use self-related (SR) stimuli and non-self-related (NSR) stimuli to study human self-recognition to see if self-referential effects (SRE) are ubiquitous. In a study of face cognition, the self-referential effect of the face was also found. This study found that people who are familiar with the face of the person will have a larger volatility in the P300 when they identify themselves. It takes more attention to resources when you see your face[2]. In a study of speech cognition, using the Oddball paradigm, with the selective attention of voice as an indicator, explored the interaction between self (SR) and non-self (NSR) and stimulus types, and found that when the subjects were exposed to them When it comes to its own voice, its self-acoustic law is easier to extract and classify than other people's voices[3]. In a complex and changing social context, people can quickly and accurately notice self-related information with high social adaptability[4, 5].

In addition, the memory of self-related information is superior to information under other coding conditions [6]. Self-related information appears to be subject to more elaborate and organized processing[7]. In the study of the ownership of an item, when the item is associated with the possessive pronoun "my" and "his", the recall of the "my" item is generally better than the "his" item[8].So it is a good explanation that SRE still exists when studying the content of unfamiliar SR. In recent years, substantial neuroscience research has been devoted to finding out whether the two processing modes associated with the "I" and the "me" (My) are embodied in the human language. As an ability to reflect on one's own feelings, thoughts, and behaviors, and to distinguish oneself from others, the basic prerequisite for the emergence of the self is the development of self-awareness in the first two years after birth[9, 10], the use of pronouns[11, 12], and the proliferation of neural networks like mirror neurons[13, 14].

In previous studies, pronouns were presented visually either in a sentence context [2], or in pronominal phrases [13-15], or in the absence of a semantic context, for example in the oddball task[8]. However, these studies of pronouns focused more on the presentation of visual stimuli, and few studies introduced second-person pronouns. By using German SRPs (first-person perspective) "ich" and "mein" ("I" and "my"), the second-person pronouns "du" and "dein" ("you" and "your"), and-NSRPs-(third-person perspective) "er" and "sein" ("he" and "his"), Herbert, Blume, and Northoff [16] proved that, in audio conditions, second-person pronouns were related to the self, and they were given priority during processing and distinguishing personal pronouns from possessive pronouns. However, in their study, only female voice was used for the stimulus materials, and no distinction was made between genders.

In order to explore, in the context of Chinese culture, whether the processing of pronouns will be different and explore whether the pronouns will have processing differences under the two kinds of visual and audio stimulation. Specifically, the hypotheses of this study were as follows for the acoustic conditions: (1) second-person pronouns are associated with the self, and first- and third-

person pronouns are associated with others when a second-person pronoun is introduced; (2) There is a difference between the personal pronoun and the SRE of the possessive pronoun; (3) these associations, interference, and differences are affected by the gender of the participants, and the consistency or inconsistency between the gender of the stimulus materials and the participant. For the visual condition, (1) when a second-person pronoun is introduced, first-person pronouns will be associated with the self and third-person pronouns will be associated with others, but participants may have different perspectives and second-person pronouns may be associated with the self and others; and (2) There is a difference between the personal pronoun and the SRE of the possessive pronoun.

II. EXPERIMENT 1

A. Participants

Eighty participants (40 males; aged 20 to 23 years, mean age 21.5 years) who were native speakers of Mandarin participated in this acoustic experiment. All participants were right-handedness, and none of the participants had any hearing impairment. The study has been examined and

approved by the Ethics Committee of the School of Psychology of Northwest Normal University. Before the experiment, each participant signed an informed consent form and was sent a gift after completing the experiment.

B. Stimulus Materials

The voice was recorded by the same experimenter in a very quiet room by using a tape recorder (LLC-9000). The two people who recorded the voice (one man and one woman) were selected from the junior high school students of the Northwest Normal University, and the participants who participated in the experiment were strangers. They did not know the purpose of the experiment and did not participate in the subsequent experiments. The two students recorded the first, second and third personal pronouns "you", "I", "he" and the possessive pronouns "my", "your" and "his". The pronunciation is pronounced in a steady, non-emotional tone, and the experimenter can hear it. Cooledit pro 2.0 was used to intercept the audio, after which the durations of the words were utilized to ensure that all words had a duration of 500 ms. The basic frequency and intensity of the voice stimuli have been presented in Table 1.

TABLE 1.VOCAL PARAMETERS OF PRONOUNS IN DIFFERENT GENDER VOICES.

Gender of the voice	Vocal parameters	Pronouns					
		You	I	He	Your	My	His
Male	Basic frequency (Hz)	121.94	113.87	191.72	137.48	125.98	172.10
	Intensity (dB)	70.53	69.16	77.50	71.02	71.39	76.75
Female	Basic frequency (Hz)	235.57	217.82	167.47	217.59	218.19	193.02
	Intensity (dB)	64.42	67.94	69.09	67.19	64.23	62.90

C. Experimental Design

Experiment 1 employed a 2 (gender consistency: consistency or inconsistency of the gender of the participant and the stimulus voice) \times 2 (type of pronoun: personal or possessive pronouns) \times 3 (type of person: first, second, and third person) within-participants design. The types of pronouns and person, and gender consistency/inconsistency were within-participant variables, while the response types of the keystrokes made by the participants and the response time were the dependent variables.

D. Procedure

Each trial starts with a centering gaze point "+" time of 500ms. Then the participant will hear the voice stimulation from the earphone for 1000ms. Finally, let the subject respond as quickly as possible. SR. The experiment stimulates the difference in the presented material. In other words, the stimulus presented by the male voice is a procedure, and the stimulation presented by the female is a procedure, so the subject has to perform two procedures. In a program, there are a total of 5 blocks, one block is a practice experiment, the personal pronoun "ni", "wo" and "ta" contain two blocks, and the possessive pronouns "ni de", "wo de" and "ta de" respectively occupy Two blocks. In each block, each voice stimulus appears 15 times, for a total of 6 voice stimuli. So trials are a total of 180 ($2 \times 6 \times 15$). Each block is pseudo-randomly presented, ensuring that

two identical blocks appear consecutively, and each trial is also pseudo-randomly present, ensuring that two identical voice stimuli do not appear consecutively. Participants sit 60cm in front of the computer screen and are required to press two previously set buttons. One button is SR and one button is NSR. The reaction button is balanced between the subjects. Stimulation through the headphones (SHP9500) and presentation through the ears. The randomized presentation of the stimulus was run on a computer via E-prime 2.0 (Psychology Software Tool 1996–2012) and the frequency of the reaction buttons was recorded. In the course of the experiment, the response of the subjects due to preemptive judgment, keystroke errors, etc., that is, the number of responses of each individual to each pronoun is 30, if the subject responds to a certain pronoun in "self" or "other" If the number of times on the category is less than 4, the data on the reaction category is rejected.

III. Results

A. Frequency of Responses by Reaction Categories

The frequency of participants' perception of the relation of each pronoun to others or to the self has been presented in Fig 1 and Fig2. The frequency of the voice stimulus and the gender of the subject is related to the frequency of "others" or "self".

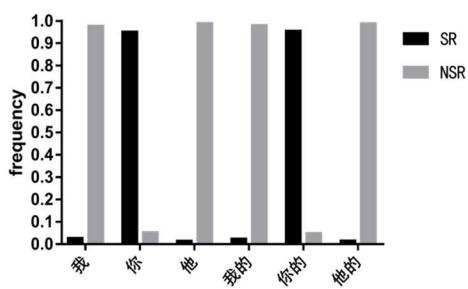


Figure1. Frequency of relating pronouns to others and to the self.
“我”=“I”, “你”=“you”,“他”=“he”;“我的”=“my”,“你的”=“your”,“他的”=“his”

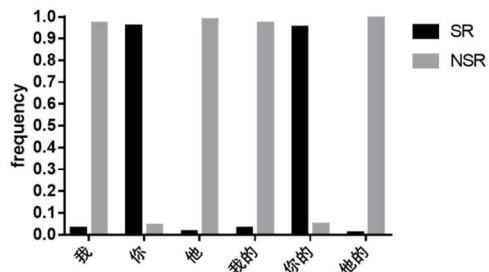


Figure2. Frequency of relating pronouns to others and to the self.
“我”=“I”, “你”=“you”,“他”=“he”;“我的”=“my”,“你的”=“your”,“他的”=“his”

B. Reaction Time for High Frequency Categories

The reaction time for the high frequency categories (self or others) of the personal and possessive pronoun was examined. The descriptive statistics for these reaction times have been presented in Table 2.

C. Repeated Measures Analysis of Variance

Participants' reaction time was submitted to a 2 (gender consistency: consistency or inconsistency between the gender of the participant and stimulus voice) \times 2 (pronoun category: personal and possessive pronouns) \times 3 (type of person: first, second, and third person) analysis of variance. The analysis revealed a significant effect of gender consistency, $F(1, 79) = 22.36$, $p < 0.01$, $\eta^2 = 0.957$, and of the person category, $F(2, 158) = 35.938$, $p < 0.01$, $\eta^2 = 0.981$; but no effect of the pronoun category, $F(1, 79) = 0.222$, $P = 0.639$. Further, there was no significant interaction effect.

IV. DISCUSSION

The results of Experiment 1 showed that, when a second-person pronoun was introduced in the acoustic condition, participants associated the first-person pronouns with others and the second-person pronouns with the self. This was because the relationship between the self and the first-person reference changed during listening to the stimuli. Specifically, the listener associated the second-person pronouns (you and your) with the self (listener) and the first-person pronouns (I and my) with the speaker [9, 10].

TABLE 2. REACTION TIME FOR PERSONAL AND POSSESSIVE PRONOUNS FOR EACH PERSON CATEGORY (MS).

Pronoun	Consistency of gender between participants and stimulus			Inconsistency of gender between participants and stimulus		
	voice			voice		
	First person	Second person	Third person	First person	Second person	Third person
Personal pronouns	701.59 \pm 157.10	637.74 \pm 141.90	638.93 \pm 157.49	747.65 \pm 188.63	673.46 \pm 153.36	666.06 \pm 164.79
Possessive pronouns	699.56 \pm 188.69	644.55 \pm 145.52	648.64 \pm 167.64	739.67 \pm 170.31	678.20 \pm 145.74	679.10 \pm 164.00

Further, in the acoustic condition, both first- and third-person pronouns were associated with others, but the reaction time for first-person pronouns was longer than that for third-person pronouns, with a statistically significant difference. Second-person pronouns were associated with the self and third-person pronouns were associated with others; however, there were no significant differences between their reaction times. This was because it is customary for people to associate “I” and “my” with the self, even if the participants associated “you” and “your” with the self in the acoustic condition. Further, the first- and second-person pronouns interfered with each, leading to longer reaction time for both as compared to that for third-person pronouns. Thus, no advantage of processing was observed.

In the gender consistent condition (male participants listened to the male voice and female participants listened to the female voice) the response bias was consistent as

compared to that in the gender inconsistent condition (male participants listened to the female voice and female participants listened to the male voice). That is, both first- and third-person pronouns were associated with others, and second-person pronouns were associated with the self. However, because the same-gender voice was similar with the self’s voice in terms of fundamental frequency and timbre, the familiarity led participants to adapting to and recognize the speech more quickly. Therefore, the reaction time of the participants for the same-gender voice was shorter than that for the opposite gender voice, for all personal pronouns.

In the acoustic condition, there were no significant differences in the response bias or reaction time between the personal and possessive pronouns, which may be caused by the limitations of behavioral experiments. This finding needs to be studied further.

V. EXPERIMENT 2

In order to confirm that the result of Experiment 1 is related to the acoustic way of presenting stimuli, Experiment 2 will explore by changing the sensory pathway, and the way of stimulating presentation will be changed to visual presentation, that is, when visually presenting the first person, second person and third person, people Whether the cognition of his pronoun is different from the cognition of acoustic presentation.

A. Participants

Sixty subjects with a native Chinese language (including 30 males; aged 20 to 25 years, average 22 years old) participated in this visual experiment. All subjects were right-handed and there were no visually impaired subjects. Before the experiment, each participant completed the informed consent form and sent a gift after completing the experiment.

B. Stimulus Materials

The stimuli included the first-, second-, and third-person pronouns "you," "I," "he," and the possessive pronouns "my," "your," and "his." All pronouns were presented in Chinese characters using Song Typeface, size 58.

C. Experimental Design

Experiment 2 uses the design of 2 (pronoun type: personal pronoun, possessive pronoun) \times 3 (personal type: first person, second person and third person). In the experiment, the person type and the pronoun type are all the variables in the test, and the reaction type and response time of the test button are the dependent variables.

D. Experimental Procedure

Each trial starts with a centering gaze point "+" time of 500ms. Then the subject will see the stimulus from the screen for 1000ms. Finally, let the subject respond as quickly as possible to SR. In this program, a total of 5 blocks, one block for practice experiments, the personal pronoun "you" "I" "he" contains two blocks, the possessive pronoun "your" "my" "his" contains two Blocks. In each block, each voice stimulus appears 15 times, for a total of 6 stimuli. So trials are a total of 180 ($2 \times 6 \times 15$). Each block is pseudo-randomly presented, ensuring that two identical blocks appear consecutively, and each trial is also pseudo-randomly present, ensuring that two identical voice stimuli do not appear consecutively. The subject sat in front of the computer screen 60cm, was asked to press two previously set buttons, one button is SR, one button is NSR, the reaction button is balanced in the test room. Stimulation through the headphones and presentation through the ears. The randomized presentation of the stimulus was run on a computer via E-prime 2.0 (Psychology Software Tool 1996–2012) and the frequency of the reaction buttons was recorded. In the course of the experiment, the response of the subjects due to preemptive judgment, keystroke errors, etc., that is, the number of responses of each individual to each pronoun is 30, if the subject responds to a certain

pronoun in "self" or "other" If the number of times on the category is less than 4, the data on the reaction category is rejected.

VI. RESULTS

A. Frequency of responses by reaction categories

The participants were divided into two categories based on their reactions to second-person pronouns (reacted as the self or as others), and the frequency of relating each pronoun with the self or others was computed for each participant (Fig 3).

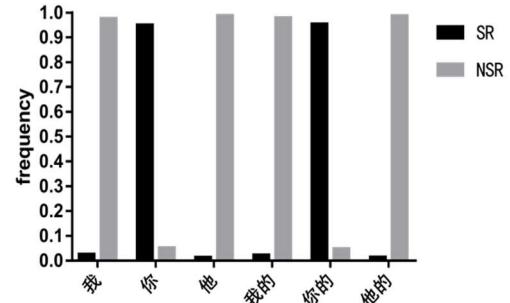


Figure3. Frequency of relating pronouns to others and to the self. "我"="I", "你"="you", "他"="he"; "我的"="my", "你的"="your", "他的"="his"

B. Statistics when SR Button Reacts

The participants were divided into two categories based on their reactions to second-person pronouns (reacted as the self or as others), and the reaction time for the personal and possessive pronouns was recorded for each person type. The descriptive statistics for these reaction times have been presented in Table 3. Repeated measures analysis of variance For the personal pronoun, the possessive pronoun performs the repeated variance analysis of the pronoun category (pronoun category: personal pronoun; possessive pronoun) \times 3 (personal type: first person, second person and third person) in the reaction of the "SR" button. The variance analysis of the SR button reaction shows that the pronoun type main effect $F(1,59)=0.38$, $p>0.05$, $\eta^2=0.006$, because there is no significant difference in the reaction of different pronoun types, the main effect F of the person type = 9.76 , $p < 0.05$, $\eta^2 = 0.142$, because the subjects had different cognitive processing for different personal pronoun types. The interaction between the personal pronoun and the pronoun type $F(2,118)=3.61$, $p<0.05$, $\eta^2=0.06$. Further simple effect analysis, under the personal pronoun type, the difference between the first person and the second person and the second person and the third person is significantly $p<0.05$, while the difference between the first person and the third person is not significant $p>0.05$; under the type of the pronoun, the first The difference between the first person and the second person and the first person and the third person was significantly different ($p<0.05$), but the difference between the second person and the third person was not significant $p>0.05$.

TABLE 3 PERSONAL PRONOUNS AND POSSESSIVE PRONOUNS IN RESPONSE TO THE "SELF" BUTTON (MS)

pronoun	First person	Second person	Third person
Personal Pronouns	653.19±207.43	697.46±220.18	655.12±193.41
Owner pronoun	632.61±174.33	705.35±236.08	700.23±267.45

VII. DISCUSSION

In Experiment 2, this study explores whether there is a difference in the self-cognitive processing of different person types in different pronoun types through visual presentation.

First of all, through the results of the button frequency in Experiment 2 (Figure 3), for different pronoun types, people's response to their buttons is not significant, indicating that there is no difference in the cognitive processing of personal pronouns and possessive pronouns. The results of the personal type indicate that people's cognitive processing of different person types is significantly different. According to (Figure 3), the participants classified the first person "I" and "My" into the NS component, and the third person "He" and "His" as the NSR. For the second person's button frequency, it is because people have different perspectives on "you" and "your", and people will have different processing. In a similar study, the 2PP pronouns combine aspects of both, self- and other-reference[17]. Explain that when people see the second person, they sometimes use the second person as the "self" component because of different cognitive perspectives. For example, when listening to pronouns, the second person "du" ("you") was preferentially processed in Very early time. This early processing bias is well in line with the notion that when the self is addressed as the subject of experience, self-reference can be established pre-reflectively without self-reflection[16]. It is like the object is saying "you" or "your" to the subject, so the second person refers to the subject himself. When the subject looks at the second person from another angle, he will be treated as a component of NSR for example, The personal and possessive pronouns "du" and "dein" ("you" and "your") were processed similarly To NSRPs during the "fast" passive silent reading condition in which stimuli were presented at a presentation rate of 2.5 Hz [17]. It is like the subject saying "you" and "your" to the object, so the second person refers to the object.

In the visual representation of the possessive pronouns we can see that the reaction of the first person is the shortest relative to the second person and the third person. We can see that the reaction of the third person is significantly increased compared to the third person in the personal pronoun. This may be because the second person is perceived as being more cognitive than the need to perceive the third person as a need for more cognitive resources. The contrast between the two has a side-suppressing effect, making the reaction time longer, shorter and shorter, and ultimately The difference between the first person and the third person disappears [16].

VIII. CONCLUSION

In the acoustic condition, the pronouns "you" and

"your" were related to the self, while "I" and "my" were related to others. Because of the mutual interference of first- and second-person pronouns, there was no obvious self-processing advantage of second-person pronouns. In addition, the influence of gender on the relationship between personal pronouns and the self was only reflected in the reaction time.

In the visual condition, first-person pronouns were related to the self, but owing to differing perspectives among participants, some participants associated second-person pronouns (you and your) with others, while others associated them with the self. When participants associated second-person pronouns (you and your) with the self, the introduction of second-person pronouns did not interfere with the processing of first- and third-person pronouns, and the advantageous effect of the self-processing of first-person pronouns persisted. When participants associated the second-person pronouns (you and your) with others, the difference between the processing of first-and third-person pronouns disappeared. In addition, in the visual condition, the reaction time for possessive pronouns was significantly longer than that for personal pronouns. This indicates that the cognitive processing methods of personal pronouns and possessive pronouns may be different, and further research is needed for their related research.

ACKNOWLEDGEMENTS

This research was completed as part of the academic requirements for the National Science Foundation of China (NSFC) under grant No. 31860285 and No. 31660281. Additionally, part of this work is performed in the Scientific Research Project in Higher Education Institutions of Gansu Province (Grant No. 2017A-165).

REFERENCES

- [1] C. T. Burris and J. K. Rempel, "'It's the end of the world as we know it': threat and the spatial-symbolic self," *Journal of personality and social psychology*, vol. 86, p. 19, 2004.
- [2] M. Esslen, S. Metzler, R. Pascual-Marqui, and L. Jancke, "Pre-reflective and reflective self-reference: a spatiotemporal EEG analysis," *Neuroimage*, vol. 42, pp. 437-449, 2008.
- [3] T. Conde, Ó. F. Gonçalves, and A. P. Pinheiro, "Stimulus complexity matters when you hear your own voice: Attention effects on self-generated voice processing," *International Journal of Psychophysiology*, vol. 133, pp. 66-78, 2018.
- [4] J. A. Bargh and F. Pratto, "Individual construct accessibility and perceptual selection," *Journal of Experimental Social Psychology*, vol. 22, pp. 293-311, 1986.
- [5] E. C. Cherry, "Some experiments on the recognition of speech, with one and with two ears," *The Journal of the acoustical society of America*, vol. 25, pp. 975-979, 1953.
- [6] T. B. Rogers, N. A. Kuiper, and W. S. Kirker, "Self-reference and the encoding of personal information," *Journal of personality and social psychology*, vol. 35, p. 677, 1977.
- [7] C. S. Symons and B. T. Johnson, "The self-reference effect in memory: a meta-analysis," *Psychological bulletin*, vol. 121, p. 371, 1997.

- [8] Z. Shi, A. Zhou, P. Liu, P. Zhang, and W. Han, "An EEG study on the effect of self-relevant possessive pronoun: Self-referential content and first-person perspective," *Neuroscience letters*, vol. 494, pp. 174-179, 2011.
- [9] P. Ruby and J. Decety, "Effect of subjective perspective taking during simulation of action: a PET investigation of agency," *Nature neuroscience*, vol. 4, p. 546, 2001.
- [10] K. Vogeley and G. R. Fink, "Neural correlates of the first-person-perspective," *Trends in cognitive sciences*, vol. 7, pp. 38-42, 2003.
- [11] K. B. Clark and M. K. Clark, "The development of consciousness of self and the emergence of racial identification in Negro preschool children," *The Journal of Social Psychology*, vol. 10, pp. 591-599, 1939.
- [12] M. Lewis and D. Ramsay, "Development of self-recognition, personal pronoun use, and pretend play during the 2nd year," *Child development*, vol. 75, pp. 1821-1831, 2004.
- [13] M. D. Giudice, V. Manera, and C. Keysers, "Programmed to learn? The ontogeny of mirror neurons," *Developmental science*, vol. 12, pp. 350-363, 2009.
- [14] J. F. Lepage and H. Théoret, "The mirror neuron system: grasping others' actions from birth?," *Developmental science*, vol. 10, pp. 513-523, 2007.
- [15] C. Herbert, B. M. Herbert, T. Ethofer, and P. Pauli, "His or mine? The time course of self-other discrimination in emotion processing," *Social neuroscience*, vol. 6, pp. 277-288, 2011.
- [16] C. Herbert, C. Blume, and G. Northoff, "Can we distinguish an "I" and "ME" during listening?—an event-related EEG study on the processing of first and second person personal and possessive pronouns," *Self and Identity*, vol. 15, pp. 120-138, 2016.
- [17] C. Blume and C. Herbert, "The HisMine-Paradigm: A new paradigm to investigate self-awareness employing pronouns," *Social neuroscience*, vol. 9, pp. 289-299, 2014.

Automated Prediction of Item Difficulty in Reading Comprehension Using Long Short-Term Memory

Li-Huai Lin, Tao-Hsing Chang

Dept. of Computer Science and Information Engineering
National Kaohsiung Univ. of Science and Technology
Kaohsiung, Taiwan.
{1105108143, changth}@nkust.edu.tw

Fu-Yuan Hsu

Institute for Research Excellence in Learning Sciences
National Taiwan Normal University
Taipei, Taiwan.
kevinhsu@ntnu.edu.tw

Abstract—Standardized tests are an important tool in education. During the test preparation process, the difficulty of each test item needs to be defined, which previously relied on expert validation or pretest for the most part, requiring a considerable amount of labor and cost. These problems can be overcome by using machines to predict the difficulty of the test items. In this study, long short-term memory (LSTM) will be used to predict the test item difficulty in reading comprehension. Experimental results show that the proposed method has a good prediction for agreement rate.

Keywords-sentence Reading comprehension; Item difficulty estimation; Long short-term memory

I. INTRODUCTION

The item response theory (IRT) is one of the main theories of modern educational assessment. Many tests such as Test of English as a Foreign Language (TOEFL) and Scholastic Assessment Test (SAT) are developed based on this method. The utilization of IRT involves an item bank, which must include the difficulty of each item. In the past, test item difficulty was estimated by one of two methods. In the first one, the difficulty of each item is collected in small-scale experiments, these experiments are also called pretest. However, the pretest method is very costly. The second method adopts expert evaluation, which costs less but is not as accurate as the pretest.

Using a computer program for automated item difficulty estimation solves the two problems above. There has been extensive research on automated machine evaluation for English content, but most of them classify the difficulty of test items based on manually defined features. By virtue of the development of deep learning, many classification models have had manually defined features replaced with those learned by neural networks. Long short-term memory (LSTM) [1] is a deep learning model commonly used in natural language processing (NLP). Compared with other deep learning models, LSTM features the utilization of context to improve the accuracy of prediction. Therefore, the purpose of this paper is to propose an automated item difficulty estimation method based on LSTM. However, because the test items may belong to various categories of subjects, and some contain pictures and tables, etc., that cannot be easily converted into a single data format, this study focuses only on Chinese reading comprehension items consisting only of words.

II. RELATED WORKS

Freedle and Kostin [2]-[3] pioneered the studies on difficulty evaluation for reading comprehension. The study

assumed many variables of the test items, such as those defining whether an item tests the examinees' understanding of the main idea or expecting the examinee to deduce a result from the item statement, and so on. The study used the calculation results based on these variables to evaluate the item difficulty. Using the method above, Chon and Shin [4] first selected a series of predictors, such as response time and paragraph length, that may affect the difficulty of test items based on relevant studies and statistical data obtained from the College Scholastic Ability Test (CSAT). After classification, the predictors were then compiled into a scale. As validated by the actual application in the preliminary CSAT in September 2009, the methods of the study can be used to effectively estimate the overall average score of the English section of the CSAT. The methods are also readily applicable to similar tests.

Boldt and Freedle [5] proposed a model based on a neural network to predict the difficulty of test items and applied a genetic algorithm (GA) to obtain improved prediction results. Based on the above research, Loukina et al. [6] proposed to predict the difficulty of listening tests according to the text complexity in language proficiency tests. The study used the TextEvaluator, a system for predicting the complexity [7]-[9], to extract the text complexity features of the items. Also, the linear regression classifier in SciKit-Learn Laboratory (SKLL) was used to evaluate the item difficulty.

Hsu et al. [10] utilized the word embedding technique proposed by [11]. First, the semantic space was constructed using the learning corpus. Then, the items and options were projected onto the semantic space to obtain the corresponding semantic vectors. Next, they computed the semantic similarities among the stem, answer, and distractors, which were then input to a support vector machine (SVM) for training to predict the item difficulty. Among the methods mentioned above, only the estimation model achieved automated calculation, and the predictors still required manual definition or evaluation.

LSTM can achieve automated extraction of the features in a text that can be used as the bases for prediction. Wu et al. [12] used LSTM as the computation unit and constructed an eight-layer LSTM encoder-decoder model for machine translation. Adopting a gated recurrent unit (GRU) [13], Song et al. [14] proposed a text-emotion detection system integrating a four-layer model. This system starts from the bottom layer. First, it converts the words into word vectors. Then it inputs each word vector of a sentence into a sentence level GRU layer, whose output is equivalent to the sentence vector of the corresponding

sentence. Next, it inputs the sentence vectors into the discourse level BiGRU layer. Finally, a multi-label layer outputs the results of emotional labeling. This method offers considerable assistance in essay analysis and scoring of compositions, etc. This study also adopts LSTM in designing the item difficulty prediction model.

III. METHOD

This paper focuses solely on the difficulty evaluation of Chinese reading comprehension items. Fig. 1 shows an example of the test items involved in this study. Unlike English, Chinese words and phrases are not separated from each other by spaces. Therefore, Chinese sentences must first undergo the word segmentation procedure so that the word boundaries within the sentences can be identified. This paper adopts WECA [15] for word segmentation, which utilizes both the lexicography-based bidirectional maximum matching and the conditional random field (CRF) as a second word segmentation method.

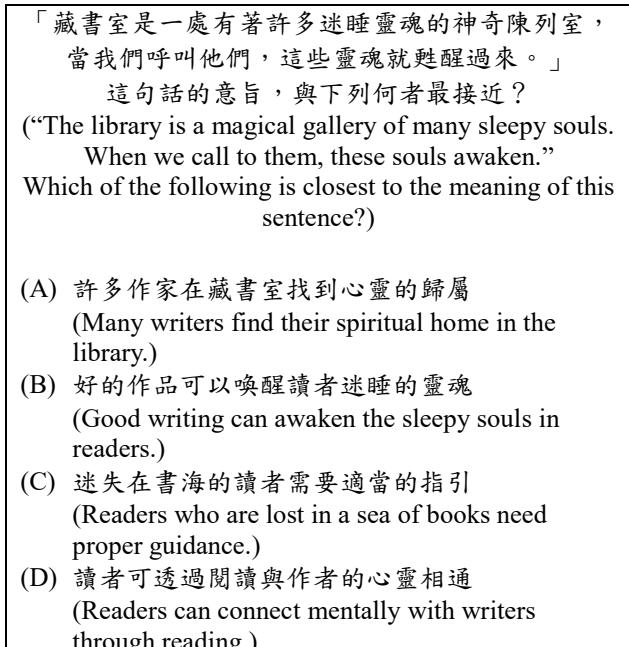


Fig. 1. Example of an item from the reading comprehension test, retrieved from the Chinese section of comprehensive assessment program (CAP) test in 2018.

This study adopts word embedding to transform words into semantic vectors. This technique is able to identify the coordinates of words in the semantic space, namely, the word vector. In other words, by examining the distance between two-word vectors, the degree of semantic similarity between two words can be determined. This study utilizes word embedding proposed by Mikolov et al. to train the transformation model to perform word-semantic vector transformation. When the training is completed, a word can be transformed into a word vector through this model. This study uses the Sinica Balance Corpus as the data source for model training. When the training is completed, each word is converted into a vector of 300 dimensions.

In this study, the LSTM proposed by Hochreiter and Schmidhuber is adopted as the prediction model, as shown

in Fig. 2. Using this as an example, C_{t-1} represents the current memory, h_{t-1} represents the previous LSTM output, and x_t represents the current input value. Through the forget gate and input gate, the decision can be made on whether to update the memory C_t thereby affecting the output value h_t .

If the current word or phrase is related to a new topic, the forget gate will filter out the previous memory. The input gate determines whether the current input and the newly-generated memory cell candidate should be added to the long-term memory. The two gates are usually sigmoid functions that indicate whether the previous memories have been forgotten or retained. The output gate is also a sigmoid function, determining whether to add the current word or phrase to the output.

This method is one of the improved models of RNN. Through appropriate gate design, the problems of vanishing and exploding gradients generated by RNN are avoided while retaining the same input and output methods as those for RNN.

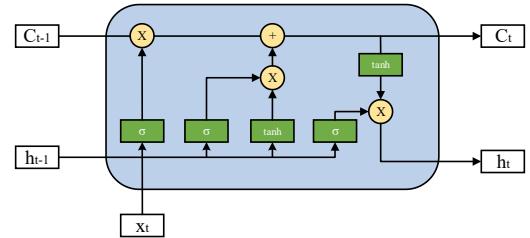


Fig. 2. LSTM

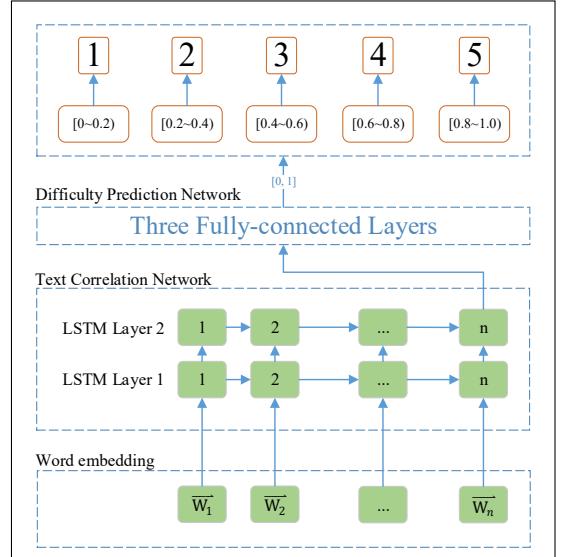


Fig. 3. The architecture of the proposed model.

Fig. 3 shows a multi-layer LSTM prediction model used in this study. This model is based on the architecture of model proposed by [14]. Our model consists of two networks: text correlation network (TCN) and difficulty prediction network (DPN). First, the proposed model converts the words of an item into semantic vectors using word embedding technique. For instance, the word W_1 in Fig. 3 is converted into the semantic vector $\overrightarrow{W_1}$ which

consists 300 dimensions. These vectors are input into the TCN.

The TCN contains two uni-directional LSTM layers to establish the correlation between the words of an item. Then, the output derived from the TCN are input into the DPN. DPN is composed of three fully-connected (FC) layers. It employs the features of inputs to evaluate the difficulty of an item. The DPN output a value which is between 0 and 1 finally. Since the difficulty of an item is defined from level 1 to level 5, the value is converted into a level by a conversion rule.

IV. EXPERIMENT

The training and test data collected for this experiment included 80 items from the basic competence test (BCTEST) and CAP TEST during 2013-2017. Due to the shortage of items, another 254 items designed by experts were added, resulting in a total of 334 items. The experts were professionals with over ten years' experience for analyzing items. The item difficulty was divided into five levels, with "difficulty 1" being the easiest item and "difficulty 5" the most difficult. The difficulty of the BCTEST and CAP TEST items were determined by the actual test results, whereas the items designed by experts were determined by the experts.

In the experiment, Adam was adopted as the optimizer of the proposed neural networks. The loss function was mean squared error (MSE). The number of repetitions of training was 5000 epochs. The initial learning rate was 0.0001, which decreased as the number of repetitions of training increased. Due to the small amount of experimental data, the 10-fold cross validation method was adopted for the experiment. Fig. 4 shows the learning curves of the proposed model. It suggests that the model have no overfitting.

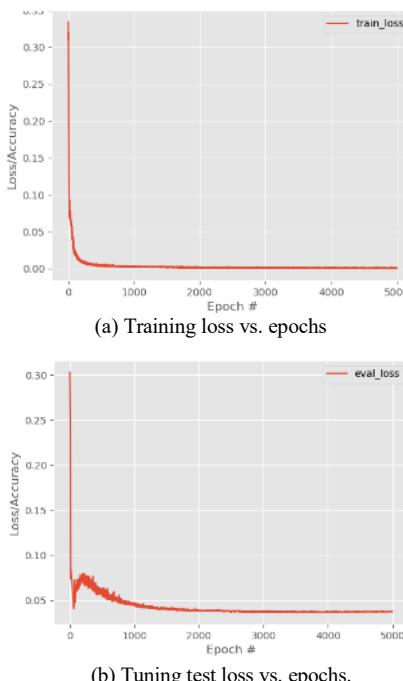


Fig. 4. Learning Curves

Because the automated difficulty evaluation methods in the previous studies were not applicable to Chinese reading comprehension items, the experiment only presents the results obtained via the method proposed in this study. Table 1 shows the confusion matrix produced by the experiment. From Table 1, three indices representing predictive effectiveness can be further calculated. The first is the exact agreement rate (EAR), which indicates the proportion of items whose predicted item difficulty exactly equals the actual difficulty. The second is adjacent agreement rate (AAR), which indicates the proportion of the items whose predicted item difficulty has a difference of one level from the actual difficulty. The serious error rate (SER) indicates the proportion of items whose predicted difficulty has a difference of over three levels from the actual difficulty.

As shown in Table 1, the EAR, AAR, and SER were 0.37, 0.84, and 0.01, respectively. According to the analysis of the difference between the predicted difficulty and the actual difficulty proposed by [10], the validity of the current results is very close to that given by pretests. This suggests that the model is indeed effective in assessing item difficulty.

TABLE I. CONFUSION MATRIX

	Actual difficulty					
	1	2	3	4	5	
Predicted difficulty	1	19	5	2	0	0
	2	34	27	13	12	0
	3	14	24	30	16	9
	4	3	9	19	21	33
	5	0	1	2	13	28

V. CONCLUSIONS AND FUTURE WORK

This study presents a difficulty prediction model for test items based on LSTM, which can effectively predict the item difficulty in Chinese reading comprehension as demonstrated by preliminary experiments. There are four potential topics for future work.

Firstly, the test items used in this experiment were in Chinese, thus the validity of the method in an English context cannot be confirmed. However, many studies have shown that word embedding and LSTM, which constitute the proposed method in this paper, perform identically in different languages. Therefore, it is appropriate to apply them to English items. Secondly, the method proposed in this paper was tested using the items in reading comprehension only. Its validity for other types of test items needs to be further tested and evaluated. Thirdly, the reasons for the difficulty in confirming the item features captured by LSTM still require investigation.

Therefore, it may be possible to further improve the accuracy of prediction by combining the manually defined characteristics that are known to be valid and whose results can be obtained by automated calculations. Finally, other deep learning models can be used as prediction models to explore the most suitable one for test item difficulty prediction.

ACKNOWLEDGMENT

This study was partially supported by the Ministry of Science and Technology (MOST), under the grant 107-2511-H-992-001-MY3 and 108-2813-C-992-017-H, and also financially supported by the “Institute for Research Excellence in Learning Sciences” of National Taiwan Normal University (NTNU) from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan.

REFERENCES

- [1] S. Hochreiter and J. Schmidhuber. “Long Short-term Memory.” *Neural computation*, vol. 9, no. 8, pp. 1735-1780, Nov. 1997.
- [2] R. Freedle and I. Kostin. “The prediction of SAT reading comprehension item difficulty for expository prose passages.” *ETS Research Report*, vol. 1991, no. 1, pp. i-52, Jun. 1991.
- [3] R. Freedle and I. Kostin. “The prediction of TOEFL reading item difficulty: implications for construct validity.” *Language Testing*, vol. 10, no. 2, pp. 133-167, Jul. 1993.
- [4] Y. V. Chon and T. Shin. “Item Difficulty Predictors of a Multiple-choice Reading Test,” *ENGLISH TEACHING(영어교육)*, vol. 65, no. 4, pp. 257-282, 2010.
- [5] R. F. Boldt, and R. Freedle. “Using a neural net to predict item difficulty.” *ETS Research Report*, vol. 1996, no. 2, pp. i-19, Dec. 1996.
- [6] A. Loukina, S. Yoon, J. Sakano, Y. Wei and K. Sheehan, “Textual complexity as a predictor of difficulty of listening items in language proficiency tests,” in *Proc. COLING*, 2016, pp. 3245-3253.
- [7] K. M. Sheehan, M. Flor, and D. Napolitano. “A two-stage approach for generating unbiased estimates of text complexity,” in *Proceedings of the Workshop on Natural Language Processing for Improving Textual Accessibility*, 2013, pp. 49–58.
- [8] K. M. Sheehan, I. Kostin, D. Napolitano, and M. Flor. “The TextEvaluator tool: Helping teachers and test developers select texts for use in instruction and assessment.” *The Elementary School Journal*, vol. 115, no. 2, pp. 184–209, Dec. 2014.
- [9] D. Napolitano, K. M. Sheehan, and R. Mundkowsky. “Online Readability and Text Complexity Analysis with TextEvaluator,” in *Proc. NAACL-HLT*, 2015, pp. 96–100.
- [10] F. Y. Hsu, H. M. Lee, T. H. Chang, and Y. T. Sung, “Automated estimation of item difficulty for multiple-choice tests: An application of word embedding techniques.” *Information Processing and Management*, vol. 54, no. 6, pp. 969-984, Nov. 2018.
- [11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. “Distributed representations of words and phrases and their compositionality,” in *Advances in neural information processing systems 26*, 2013, pp. 3111-3119.
- [12] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, ... and J. Klingner. “Google’s neural machine translation system: Bridging the gap between human and machine translation.” Oct. 2016. Internet: <https://arxiv.org/abs/1609.08144>.
- [13] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. “Empirical evaluation of gated recurrent neural networks on sequence modeling,” in *NIPS 2014 Deep Learning and Representation Learning Workshop*, 2014. Internet: <https://arxiv.org/abs/1412.3555>.
- [14] W. Song, D. Wang, R. Fu, L. Liu, T. Liu, and G. Hu. “Discourse Mode Identification in Essays,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017, pp. 112-122.
- [15] T. H. Chang. “The Development of Chinese Word Segmentation Tool for Educational Text,” in *Proceedings of the 7th International Conference on Information*, 2017, pp. 179-182.

Automatic Meta-evaluation of Low-Resource Machine Translation Evaluation Metrics

Junting Yu¹, Wuying Liu¹, Hongye He², Lin Wang^{3*}

1 Laboratory of Language Engineering and Computing, Guangdong University of Foreign Studies, Guangdong, CHINA

2 Training Department, Shaanxi provincial Party School of CPC, Shaanxi, CHINA

3 Xianda College of Economics and Humanities, Shanghai International Studies University, Shanghai, CHINA

junting_yu@163.com; wyliu@gdufs.edu.cn; hugh5945@163.com; lwang@xdsisu.edu.cn

Abstract—Meta-evaluation is a method to assess machine translation (MT) evaluation metrics according to certain theories and standards. This paper addresses an automatic meta-evaluation method of machine translation evaluation based on ORANGE - Limited ORANGE, which is applied in low-resource machine translation evaluation. It is adopted when the resources are limited. And take the three n-gram-based metrics - BLEUS, ROUGE-L and ROUGE-S for experiment, which is called horizontal comparison. Also, vertical comparison is used to compare the different forms of the same evaluation metric. Compared with the traditional human method, this method can evaluate metrics automatically without extra human involvement except for a set of references. It only needs the average rank of the references, and will not be influenced by the subjective factors. And it costs less and expends less time than the traditional one. It is good for the machine translation system parameter optimization and shortens the system development period. In this paper, we use this automatic meta-evaluation method to evaluate BLEUS, ROUGE-L, ROUGE-S and their different forms based on Cilin on the Russian-Chinese dataset. The result shows the same as that of the traditional human meta-evaluation. In this way, the consistency and effectiveness of Limited ORANGE are verified.

Keywords-Automatic Meta-evaluation; Limited ORANGE; BLEUS; ROUGE-L; ROUGE-S

I. INTRODUCTION

The evaluation of machine translation (MT) plays an important role in the field of machine translation. For system developers, it can assess system performance to learn the system problems and improve them in time. And for users, it can provide the basis for translation quality assessment. As is provided in the language specification released by the State Language Work Committee[1], there are mainly two kinds of machine translation evaluation: human evaluation and automatic evaluation. The human evaluation mainly determine the output quality through experts' subjective judgments of certain features such as adequacy and fluency. Automatic evaluation uses a computed sentence similarity to compare MT system outputs according to a set of reference translations. It is objective and not affected by external factors. However, the human method is very subjective, and will cost a lot of

resources. Also, it is easy to be affected by external factors, and need a long period to finish the evaluation. All of these disadvantages cause the human evaluation unable to adapt to the fast MT system modification progress and parameter adjustment. This inevitably extends the system development period. So that it is difficult to provide efficient evaluation results for both developers and users. Therefore, researchers are more inclined to use the automatic method to evaluate the system performance quantitatively. Consequently, automatic evaluation has become one of the research hot spots in the machine translation.

There are so many automatic evaluation methods, and it is worth to assess their performances. For example, BLEU[2] can only be used at the corpus level, but smoothing BLEU (BLEUS) can be applied to sentence-level evaluation; BLEU does not consider the matching between the discontinuous subsequences, while ROUGE[3], based on continuous n-gram as BLEU, considers the subsequence with maximum length for candidate and reference matching; and so on. The quality of these automatic evaluation methods needs to be measured by a unified standard, that is, meta-evaluation of machine translation evaluation.

Meta-evaluation of machine translation evaluation, that is, using a certain method to detect which machine translation evaluation metric performance is better, mainly divided into two categories: human meta-evaluation and automatic meta-evaluation. The human meta-evaluation is to obtain the correlation coefficient by the adequacy and fluency scores of the automatic evaluation and the human evaluation score. It is expensive and the subjective factors always cause results inconsistent. What's more, it is difficult to achieve consistent for adequacy and fluency scores. The larger the correlation coefficient, the better the evaluation method performance. While the automatic meta-evaluation is to use the computer to assess the evaluation methods automatically. This method is objective, convenient and easy to implement, and will not be affected by external factors to cause the deviation of the evaluation results. Human meta-evaluation has many shortcomings due to manual intervention, and it is difficult to adapt to the information processing needs of massive data. As a result, it is very important to explore a general

* Corresponding Author

and objective automatic meta-evaluation method for the assessment of machine translation evaluation methods.

As the limited corpus resources, we propose a automatic meta-evaluation method named Limited ORANGE, which is based on ORANGE (Oracle Ranking for Gisting Evaluation) proposed by Lin[4] in 2004. With this method, we evaluate the performance of the current three automatic n-gram-based evaluation methods, BLEUS[4], ROUGE-L[5] and ROUGE-S[5]. Also, we evaluate some other forms of these three metrics, which consider semantic analysis based on Cilin[6]. In order to maintain consistency with the human meta-evaluation, we propose to evaluate the automatic evaluation metric performance by scoring and sorting the candidate and the reference through the adequacy score, fluency score and translation similarity. Except for the artificial reference translations, the whole evaluation process does not require additional manual intervention, and can be well applied to sentence-level evaluation.

II. LIMITED ORANGE AUTOMATIC META-EVALUATION METHOD

A. Brief introduction of used machine translation evaluation metrics

a) Smoothing BLEU (BLEUS)

Since its introduction by Papineni et al. in 2002[2], BLEU has been widely used in various evaluation activities and has many variants[7]. BLEU is calculated through matching the continuous n-grams between the candidate and a fixed set of references, and then get the precision of n-gram, where n is set 1 to 4 usually. Then take the geometric mean of the modified n-gram precisions and then multiply the result by a brevity penalty factor (BP) to punish candidate that is shorter than reference. So BLEU is defined as:

$$\text{BLEU} = e^{\min(1-r/c, 0)} \times \exp\left(\sum_{n=1}^N w_n \log p_n\right) \quad (1)$$

Where N is the maximum number of n-gram, p_n is the n-gram precision, with the $w_n=1/N$ to be corresponding weight. In the brevity penalty case, r is the length of reference, c is that of candidate.

Although it is widely used as golden standard, BLEU lacks stable, reliable, and valuable sentence-level scores, which is critical to distinguishing system performance. Therefore, it is important for BLEU to avoid zero-precision by smoothing techniques.

In 2004, Lin has proposed smoothing BLEU (BLEUS) for the first time[4]. In this paper, we adopt +1 smoothing in each n-gram precision, as defined in equation (2), to solve the zero-precision.

$$p_n = \frac{\text{Count}_{\text{clip}(n\text{-gram})} + 1}{\text{Count}_{(n\text{-gram})} + 1} \quad (2)$$

$\text{Count}_{(n\text{-gram})}$ and $\text{Count}_{\text{clip}(n\text{-gram})}$ are respectively the minimum n-grams co-occurring in reference and candidate.

b) ROUGE-L

When matching the continuous n-grams in candidate and reference, BLEU does not describe well the relationship between long-distance discontinuous words. In this respect, in 2004, Lin proposed a method named Recall-Oriented Understudy for Gisting Evaluation (ROUGE) based on discontinuous n-grams to describe the

relationship between long-distance units, such as ROUGE-L and ROUGE-S.

ROUGE-L is defined to measure sentence-to-sentence similarity based on the longest common subsequence (LCS) statistics between a candidate and a set of references. This metric considers both the precision and the recall, and uses the LCS-based F-measure to calculate the similarity between the reference X, with length of m, and the candidate Y with length of n.

$$R_{lcs} = \frac{\text{LCS}(X, Y)}{m} \quad (3)$$

$$P_{lcs} = \frac{\text{LCS}(X, Y)}{n} \quad (4)$$

$$F_{lcs} = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}} \quad (5)$$

Where $\text{LCS}(X, Y)$ is the length of the LCS of X and Y, and when $\partial F_{lcs}/\partial R_{lcs} = \partial F_{lcs}/\partial P_{lcs}$, $\beta = P_{lcs}/R_{lcs}$, which is named relative weight. The equation (5) is ROUGE-L. And the ROUGE-L is 1 when $X=Y$, while the value is 0 when $\text{LCS}(X, Y)=0$, that is, the candidate Y and the reference X are completely different.

c) ROUGE-S

As ROUGE-L only records the longest common subsequence, and does not record the co-occurrences shorter than the longest one, this method cannot describe translation similarity well. ROUGE-S calculates the similarity between candidate and reference based on the discontinuous skip-bigram, which is any pair of words in their sentence order, allowing for arbitrary gaps. The skip-bigram is similar to the 2-gram in BLEU, but contains discontinuous subsequences. Thus the ROUGE-S can get the translation similarity well. Each sentence with the length of len has $C(len, 2)$ skip-bigrams, where C is combined function. For Chinese, len stands for the number of words after segmentation. Given reference translation X of length m and candidate translation Y of length n, we compute skip-bigram-based F-measure, which is ROUGE-S, as shown in equation (8):

$$R_{skip2} = \frac{\text{SKIP2}(X, Y)}{C(m, 2)} \quad (6)$$

$$P_{skip2} = \frac{\text{SKIP2}(X, Y)}{C(n, 2)} \quad (7)$$

$$F_{skip2} = \frac{(1+\beta^2)R_{skip2}P_{skip2}}{R_{skip2} + \beta^2 P_{skip2}} \quad (8)$$

$\text{SKIP2}(X, Y)$ is the number of skip-bigram matches between X and Y. Relative weight $\beta = P_{skip2}/R_{skip2}$, when $\partial F_{skip2}/\partial R_{skip2} = \partial F_{skip2}/\partial P_{skip2}$.

d) Semantic Space Transformation-based metrics

Machine translation is actually a different encoding of the "same semantics". When automatically evaluating the translation quality of the machine translation system, if the two words in the translations are not exactly the same, they are judged as having zero similarity, which reduces the similarity of the translations. Then this will affect the evaluation performance. So Junting Yu[8] proposed an improved Cilin-based smoothing BLEU (BLEUS-syn) metric with Semantic Space Transformation (SST) algorithm.

The SST algorithm is mainly based on the unigram of the reference. When the candidate and the reference are matched to form a mapping, the exact morphological

matching is first performed, and then the synonymy matching based on Cilin is performed. The two stages are carried out in order without overlapping.

BLEUS-syn mainly introduces the synonymy matching into BLEUS. When matching the unigram of candidate and reference, it uses exact matching first, and then synonymy matching based on Cilin, in which stage candidate unigram is replaced by reference unigram. In the 2~4-grams matching, the n-gram series are based on replaced unigram. So the evaluation performance maybe improved obviously.

Also, applying the SST algorithm into the ROUGE-L and ROUGE-S will obtain the ROUGE-L-syn and ROUGE-S-syn metrics respectively.

B. Limited ORANGE

A good evaluation metric should give a higher score to a good translation than a bad one. So a good translation should be ranked higher than a bad one according to their scores. Taking two assumptions that references are good translations and the more a candidate is similar to its references the better; references are usually better than candidates. Therefore, references should be ranked higher than candidates on average if a good automatic evaluation metric is used.

As the short of some data for specific tasks or data that can be applied to restricted fields, the training of statistical machine translation models still faces serious "data sparse" problems[9]. In view of the subjectivity, long period and weak consistency of the human meta-evaluation method, this paper proposes the Limited ORANGE, based on ORANGE, as a automatic method to assess evaluation metrics performances in the case of low-resource language. This method makes automatic meta-evaluation possible because of its objective, concise and convenient implementation, which reduces the contribution of human resources.

The Limited ORANGE is proposed in the absence of corpus resources. When the corpus resources are relatively limited and the scale of the training set is not large enough, the output n-best candidate translation list is poorly readable. So the Limited ORANGE utilizes limited corpus resources and assess the metric performance automatically with the help of online translation systems. In this paper, we intend to use the output Chinese translations of the widely used Russian-Chinese online translation systems on the network as a list of candidate translations, which is set as n-best translations, and Chinese translations in Russian-Chinese aligned corpus as reference translations. Then, we sort the list of candidates and reference, calculate the rank of the reference in the n-best list, and then get the ratio of the reference rank to the length of the n-best list as the Limited ORANGE score.

Given a source sentence S , select four online translation systems Google, Baidu, Bing, and Youdao, to assist implementing the meta-evaluation research. Take the Chinese outputs of above four online systems as candidates, the corresponding Chinese in Russian-Chinese bilingual news sentence alignment corpus as reference, to construct the experimental corpus. Then each automatic evaluation method is calculated as follows:

- (1) Select translation features and calculate translation scores of candidate list and reference;

- (2) Sort the reference and the candidate list based on the translation scores, and calculate an average rank of the reference;

- (3) Calculate the ratio of the average rank of the reference to the length of the translation list, which is the Limited ORANGE score shown as follows:

$$\text{Score} = \frac{\sum_{i=1}^S \text{Rank}(\text{Oracle}_i)}{S(N+1)} \quad (9)$$

Where, $\text{Rank}(\text{Oracle}_i)$ is the rank of the reference of the source sentence i in the n -best list, S is the number of source sentences in the corpus, N is the length of translation list. The smaller the ratio, the better the automatic evaluation metric performance.

C. Translation sorting method

Through the above analysis, the most important question is how to comprehensively sort the translations? We choose a set of features to represent the translation of the same source sentence. Each feature describes an attribute of the translation. The feature weight indicates the relative importance of the feature to the translation. Any information related to the translation can be encoded as a feature. In order to maintain consistency with traditional human meta-evaluation, we intend to select the 3 features to represent translation information, such as the translation adequacy score - ade, the fluency score - flu and the translation similarity based on the automatic evaluation method - sim. These features must be obtained with the help of linguistic knowledge.

The principle of statistical machine translation is to model, train and then decode through translation models and language models[10], as shown in Figure 1.

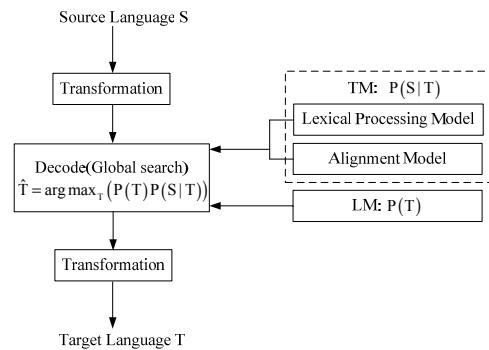


Figure 1. Framework of statistical machine translation system.

And then find the sentence with the highest probability in all possible target T as the translation result of the source S[10]. According to the Bayesian formula, it is to find a closest target sentence \hat{T} by decoding to the source S.

$$\bar{T} = \arg \max_T P(T)P(S|T) \quad (10)$$

Where, $P(S|T)$ is the translation model (TM), including lexical processing and word alignment. It indicates the degree to which the target language T is like the source language S . $P(T)$ is the language model (LM), independent of the source language, indicating the degree to which T is like a sentence in a target language, reflecting the fluency of the translation.

a) Translation adequacy score calculation

The translation adequacy score is expressed by the translation model probability $P(S|T)$, that is, the conditional probability of source language S when given

the target language T. The higher the probability value, the higher the translation adequacy score. We train the translation model with the Moses system[11], and use the GIZA++ to achieve word alignment.

Since the experiment corpus comes from four different online Russian-Chinese translation system translations and human reference translations, they are different from the n-best list from the same open source statistical machine translation system. Therefore, it is necessary to find a common "reference point" for the four candidates from different systems and the human reference to obtain their TM probabilities. We select the 1-best translation of the Moses system as the "reference point" to indirectly obtain the adequacy scores of the five translations.

(1) Obtain the 1-best of the Moses and its corresponding TM probability $P(S|T)$.

(2) Compare the similarity - m_i of the 1-best T with the five translations c_i ($i=1, 2, 3, 4, 5$) to be compared respectively. Chinese, as the target language, is a analytic language. So word order is important for similarity calculation. Then we take Levenshtein Distance to calculate the m_i . When using the Levenshtein distance to match, the exact matching is first performed. If the word forms are different, then the synonymy matching based on Cilin is considered. The two matching steps are performed in a sequence without overlap.

$$m_i = 1 - \frac{d}{\max(l_T, l_{c_i})} \quad (11)$$

Where d stands for the cost of the Levenshtein distance, l_T and l_{c_i} represent the length of the 1-best translation T and the translations c_i .

(3) Get the adequacy score- ade_i by multiplying step (1) and step (2), and this is the TM probability of the translation c_i .

$$ade_i = P(S|T) \cdot m_i \quad (12)$$

b) Translation fluency score calculation

The translation adequacy score is reflected by the LM probability $P(T)$, which indicates the likelihood that the sequence will be expressed in the target language. Currently the most widely used is n-gram to calculate the probability of the sequence $W=w_1, w_2, \dots, w_n$.

$$p(W) = p(w_1^n) \approx p(w_1)p(w_1 | w_2) \cdots p(w_n | w_{n-1}^{n-1}) \quad (13)$$

Due to the limited training corpus, there will be "data sparse". Smoothing techniques are needed to discount the visible events count and give them invisible events. Thus all probabilities are non-zero. We adopt the Katz smoothing[12] to train the language model on the basis of the SRILM toolkit. Then apply the 4-gram model obtained to the Moses system, and get the language model score - $P_{LM}(T)$. We can get the language model probabilities of the translation c_i by the same method as described in TM score.

$$flu_i = P_{LM}(T) \cdot m_i \quad (14)$$

c) Translation similarity calculation

The translation similarity calculation is based on the automatic evaluation metrics. Assume the 1-best of the Moses system as the reference r, and the other 5 translations are the candidate c_i ($i=1 \sim 5$), then the similarity of translation i is shown as follows.

$$sim_i = M_k(r, c_i) \quad (15)$$

Set an appropriate weight for each feature, the final scores of the five translations are:

$$Score_i = \lambda_1 \cdot ade_i + \lambda_2 \cdot flu_i + \lambda_3 \cdot sim_i \quad (16)$$

We set the three features the same weight, that is, $\lambda_1=\lambda_2=\lambda_3=1/3$.

III. PERFORMANCE ANALYSIS OF LIMITED ORANGE

A. Corpus and environment

In the experiment, we apply 52,892 Russian-Chinese bilingual alignment sentence pairs in the domain of news[13]. The corpus is divided into training set and test set by hierarchical sampling. The test set includes 1,057 Russian-Chinese pairs. At the same time, a total of 247,075 entries in Russian-Chinese dictionary were introduced in model training to optimize the bilingual phrase alignment model.

We select four online translation systems, such as Google, Baidu, Bing, and Youdao, which are widely used and have readable translations in Russian-Chinese translation. Take the Chinese outputs in December 2015 as candidate list, and the human Chinese sentences in corpus as references.

We run the experiment on the computer with 8.00GB memory and Intel(R) Core(TM) i7-6700HQ CPU to maintain consistency of results. The experiment is implemented using JAVA on the eclipse platform.

B. Result and discussion

First, on the test set, we take these methods on the four online translation systems, Google, Baidu, Bing, and Youdao, and get the BLEU values. Table I shows the experiment results.

TABLE I. THE BLEU VALUES OF BLEUS AND BLEUS-SYN ON TEST SET.

BLEU Value	Google	Baidu	Bing	Youdao
BLEUS	0.197001	0.214467	0.176529	0.190993
BLEUS-syn	0.209472	0.221200	0.194016	0.197361

Table I shows that, on the test set, (1) The evaluation value of BLEUS-syn is higher than that of the BLEUS for the 4 online systems; (2) When using the SST algorithm based on Cilin, the BLEU value of Baidu increases from 0.214467 to 0.221200, with an increase of 3.14%; the BLEU value of Google increases from 0.197001 to 0.209472, with an increase of 6.33%; the Youdao BLEU is increased by 3.33% from 0.190993 to 0.197361; the one of Bing is increased by 9.91% from 0.176529 to 0.194016; (3) After applying the SST algorithm, the performance of each system is improved with different degrees. All of the above analysis, BLEUS-syn performance is better than BLEUS.

In the same way, we get the F value of ROUGE-L and ROUGE-L-syn on the test set, shown as Table II.

TABLE II. THE F VALUES OF ROUGE-L AND ROUGE-L-SYN ON TEST SET.

F Value	Google	Baidu	Bing	Youdao
ROUGE-L	0.277583	0.277276	0.254985	0.257065
ROUGE-L-syn	0.339582	0.343511	0.312607	0.319314

As shown in Table II, on the test set, the F value of ROUGE-L-syn is higher than that of ROUGE-L. When using the synonymy matching, the performance of systems have different improvements obviously, for example, Google 22.34%, Baidu 23.89%, Bing 22.60%, and Youdao

24.22%. So the ROUGE-L-syn performance is better than ROUGE-L.

Then we experiment with ROUGE-S and ROUGE-S-syn on the test set to obtain the F values of 4 systems and corresponding improvements by using SST. The results are listed in the Table III.

TABLE III. THE F VALUES OF ROUGE-S AND ROUGE-S-SYN ON TEST SET.

F Value	Google	Baidu	Bing	Youdao
ROUGE-S	0.189192	0.206467	0.167508	0.168972
ROUGE-S-syn	0.233596	0.240100	0.210788	0.202358
Improvement after SST	23.47%	16.29%	25.84%	19.76%

As shown in Table III, the performance of ROUGE-S-syn, which uses SST algorithm is superior to that of ROUGE-S.

Then we achieve vertical comparison with the different forms of the same metric, such as BLEUS and BLEUS-syn, ROUGE-L and ROUGE-L-syn, ROUGE-S and ROUGE-S-syn. Table IV shows the Limited ORANGE scores of these metrics.

TABLE IV. THE LIMITED ORANGE SCORES OF THE VERTICAL COMPARISON.

Metric	BLEUS	BLEUS-syn
Limited ORANGE Score	0.003227	0.003112
Metric	ROUGE-L	ROUGE-L-syn
Limited ORANGE Score	0.003112	0.003097
Metric	ROUGE-S	ROUGE-S-syn
Limited ORANGE Score	0.003160	0.003153

The score of BLEUS is 0.003227, and its variant BLEUS-syn using the SST algorithm, has the score of 0.003112, smaller than that of BLEUS. This shows the performance of BLEUS-syn is better than that of BLEUS, which is consistent with the result demonstrated in Table I. And the metrics with SST algorithm, such as ROUGE-L-syn and ROUGE-S-syn, are better in performance than the baseline of ROUGE-L and ROUGE-S respectively. All of these results are the same as the above research shown in Table II and Table III.

The above sets of experimental analysis results show that the Limited ORANGE meta-evaluation method is consistent with the results of the traditional vertical evaluation to assess the different evaluation methods of the same type. This indicates that the Limited ORANGE method is effective for the performance evaluation of vertical comparison.

Secondly, we evaluate the performance of BLEUS, ROUGE-L and ROUGE-S with Limited ORANGE, which is called horizontal comparison. The scores of the three metrics are shown in Table V.

TABLE V. THE LIMITED ORANGE SCORES OF THE HORIZONTAL COMPARISON.

Metric	BLEUS	ROUGE-L	ROUGE-S
Limited ORANGE Score	0.003227	0.003112	0.003160

As shown in Table V, ROUGE-L has the lowest score of 0.003112. ROUGE-S is 0.003160, BLEUS is the highest 0.003234. As analyzed above, the lower the Limited ORANGE score, the better the evaluation metric performance. Therefore, the ROUGE-L performance is the best, the ROUGE-S is second, and the performance of BLEUS is the worst. The main reason is that during the

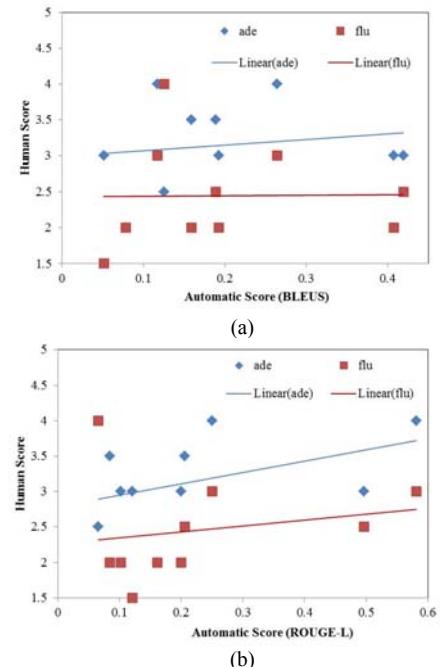
evaluation, ROUGE-L compares the translation similarity based on the longest common subsequence, and does not set the fixed length of the n-gram. The matching is flexible and not limited to the length of the n-gram. In this case, it will not affect the number of matching n-grams in the translations. ROUGE-S and BLEUS well reflect the matching of each n-gram. ROUGE-S combines the advantages of ROUGE-L and BLEUS, which includes both the continuous n-gram of BLEU and the discontinuous n-gram of ROUGE-L. ROUGE-S can not only ensure the adequacy of the translation, but also capture information of distant words. In a word, the performance of ROUGE-S is better than that of BLEUS.

To verify the utility of the automatic meta-evaluation method, we adopt the Pearson correlation coefficient r_{xy} to evaluate the three metrics, which is to measure the quality of evaluation metrics by determining the correlation between the scores of the evaluation metrics and scores of fluency and adequacy. Suppose the data point on the test set containing the variable automatic score x and the human score y is $\{(x_i, y_i)\}$, the Pearson correlation coefficient is defined:

$$r_{xy} = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}} \quad (17)$$

$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ and $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$ are the averages of sample X and Y respectively. The variable n is the source sentences number of test set. The higher the r_{xy} , the better the evaluation metric performance.

According to the results of vertical evaluation, the performance of Baidu system is the best. So we choose the automatic evaluation scores of Baidu to obtain the Pearson correlation coefficient r_{xy} . Figure 2 shows the correlation of automatic evaluation and human evaluation.



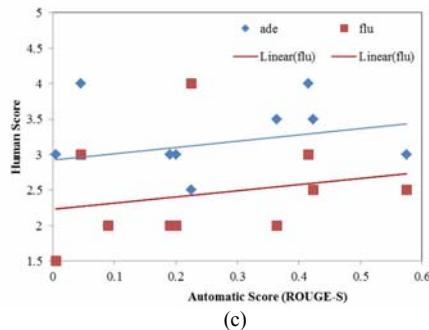


Figure 2. The correlation of automatic evaluation and human evaluation.

The x-axis indicates automatic scoring based on different automatic evaluation methods, and the y-axis is human score. The adequacy and fluency score are expressed by ade and flu respectively, which are added linear trends. (1) Both the correlation of BLEUS adequacy and fluency are the worst of the three methods. (2) The fluency correlation between ROUGE-L and ROUGE-S is basically the same. But the ROUGE-L adequacy correlation is better than that of ROUGE-S. (3) The correlation of ROUGE-S adequacy and fluency is equally good. It can be concluded that the performance of ROUGE-L is better than that of ROUGE-S and BLEUS, and the performance of BLEUS is the worst. So the experimental results of Limited ORANGE are consistent with the results of the human meta-evaluation. Thus, we verify the consistency of the Limited ORANGE method.

But at the same time, it can be seen that the human meta-evaluation involves optimization of the two objective functions of the adequacy and fluency correlation coefficients, which are always inconsistent difficultly. The comprehensive evaluation results based on the two objective functions have a lot to do with the subjective factors of people's scores. But the Limited ORANGE has only one objective function to optimize, which makes the meta-evaluation of MT evaluation operability perfectly reflected because of its objectivity. The meta-evaluation results are objective and concise. It can verify the improvement of the evaluation method performance more effectively and quickly. It can also be applied to research the languages with low-resource or in some specific fields or some special tasks. Thanks to its automation, it can free up some human resources.

IV. CONCLUSIONS

This paper mainly proposes a Limited ORANGE automatic meta-evaluation method based on ORANGE to assess the performances of the automatic MT evaluation metrics. Suppose this method works through the average rank of reference translations, without any manual intervention except the prepared references. The higher the ranking, the better the performance of metric. Compared with the traditional human meta-evaluation, the method proposed in the paper can significantly shorten evaluation time and save energy.

We take vertical comparison and horizontal comparison with BLEUS, ROUGE-L, ROUGE-S and their variants using SST algorithm on the test set. In order to be consistent with the traditional human meta-evaluation,

three factors of the adequacy score, the fluency score and the translation similarity based on the evaluated metrics are selected to represent the translation in the process of evaluation. Both the vertical comparison and horizontal comparison are verified consistent with traditional human meta-evaluation. So the Limited ORANGE automatic meta-evaluation method is proved to be effective and feasible.

Further research will concern that further optimizing the selection and calculation of translation features. And choose some other languages to experiment to increase the generalization of this method.

ACKNOWLEDGMENT

The research is supported by the Social Science Foundation of Shanghai (No. 2019BYY028), the Key Project of State Language Commission of China (No. ZDI135-26), the Natural Science Foundation of Guangdong Province (No. 2018A030313672) and the Key Project of Guangzhou Key Research Base of Humanities and Social Sciences: Guangzhou Center for Innovative Communication in International Cities (No. 2017-IC-02).

REFERENCES

- [1] Assessment Specifications of Machine Translation Systems. GF 2006
- [2] Papinen K., Roukos S., Ward T., et al. BLEU: a method for automatic evaluation of machine translation [C]. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics 2003 (ACL 2003), 2002: 311-318.
- [3] Lin C. Y.. ROUGE: A Package for Automatic Evaluation of Summaries [C]. In Proceedings of Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004. 2004.
- [4] Lin C. Y., Och F. J.. ORANGE: A Method for Evaluating Automatic Evaluation Metrics for Machine Translation [C]. In Proceedings of the International Committee on Computational Linguistics 2004 (COLING-2004), 2004.
- [5] Lin C. Y., Och F. J.. Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statistics [C]. In Proceedings of Association for Computational Linguistics 2004 (ACL 2004), 2004.
- [6] Mei J., Zhu Y., Gao Y., et al. Tongyici Cilin (Extended) [M]. HIT IR-Lab. 1996.
- [7] Yvette Graham, Barry Haddow, Philipp Koehn. Translationese in Machine Translation Evaluation. arXiv:1906.09833 [cs.CL]. 2019
- [8] Junting Yu, Wuying Liu, Hongye He, et al. BLEUS-syn: Cilin-Based Smoothed BLEU [C]. In the 12th China Workshop on Machine Translation (CWMT 2016), 2016: 102-112
- [9] ZHANG Bo. The Computational Models of Natural Language Processing [J]. Journal of Chinese Information Processing, 2007, 21(3): 3-7
- [10] Feng zhiwei. Formal Models of Natural Language Processing [M]. University of Science and Technology of China Press. 2010.1: 564-565
- [11] Koehn P.. Moses-Statistical Machine Translation System-User Manual and Code Guide [A]. 2015
- [12] Katz S. M.. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer [J]. IEEE Transactions on Acoustics, Speech and Signal Processing, 1987, 35(3): 400-401
- [13] Du W., Liu W., Yu J., et al. Russian-Chinese Sentence-level Aligned News Corpus [C]. In Proceedings of the 18th Annual Conference of the European Association for Machine Translation (EAMT 2015), 2015: 213.

Combination of Semantic Relatedness with Supervised Method for Word Sense Disambiguation

Qiaoli Zhou
School of Computer
Shenyang Aerospace University
Shenyang, China
zhou_qiao_li@hotmail.com

Yuguang Meng
School of Computer
Shenyang Aerospace University
Shenyang, China
465611370@qq.com

Abstract—We present a semi-supervised learning method that efficiently exploits semantic relatedness in order to incorporate sense knowledge into a word sense disambiguation model and to leverage system performance. We have presented sense relativity algorithms which combine neural model learned from a generic embedding function for variable length contexts of target words on a POS-labeled text corpus, with sense-labeled data in the form of example sentences. This paper investigates the way of incorporating semantic relatedness in a word sense disambiguation setting and evaluates the method on some SensEval/SemEval lexical sample tasks. The obtained results show that such representations consistently improve the accuracy of the selective supervised WSD system.

Keywords-Word Sense Disambiguation; Semantic Relatedness; semi-supervised learning; Neural Model;

I. INTRODUCTION

Word Sense Disambiguation (WSD) has been a hard nut ever since the earliest days of computer-based treatment of language in the 1950s. WSD is the task to identify the intended sense of a word in a computational manner based on the context in which it appears [1]. Many algorithms devote to WSD by exploiting two powerful properties of human language: “one sense per collocation” and “one sense per discourse” [2], [3]. In the “one sense per collocation”, the nearby words provide clues to the sense of the target word. “One sense per discourse” represents the sense that a target word is consistent with a given document. In the WSD research literature, currently, these two assumptions are widely accepted by natural language processing community. There are, however, several difficulties to WSD we need to face.

Firstly, one sense per collocation, the strong tendency for words which exhibits only one sense in a given collocation was observed and quantified in the paper [3]. Collocation refers to a group of practical words that habitually go together, whereas the sense of a word is figured out by accompanying words. In this case, words are to be classified in terms of co-occurrence relation as well as sense. The co-occurrence relation means the constraints shown in the sense combination relation, which is called collocation constraint or selection constraint. When we make use of these assumptions, it is easy to identify the sense of common expressions or idioms containing a target word. For example, the word “place” means general location. But, the meaning of the idiom “take place” is quite

different from the meaning of “take her place”. The idiom “take place” means that something occurs or happens at a particular time or place. Thus, an idiom is a group of words in a fixed order and has a particular meaning which is different from the meanings of the individual word regardless of the context of the word to be disambiguated. Although there are many researches aiming to solve WSD problem using the phrase in WordNet and idiom dictionary, when we take into consideration the overall occurrence in the target corpus, there still remain some cases where a dictionary may not cover some of the idioms that exist in the target corpus. Therefore, the effect of using collocations to resolve lexical ambiguities depends on the type of collocation. It is strongest for immediately adjacent collocations, and weakens with distance. It is much stronger for words in a predicate-argument relationship than for arbitrary associations at equivalent distance. It is very much stronger for collocations with content words than those with function words. Secondly, one sense per discourse, the sense of a target word is highly consistent within any given document. The observation that words strongly tend to exhibit only one sense in a given discourse or document was stated and quantified in Gale, Church and Yarowsky [2]. For example:

A The **stock** would be redeemed in five years, subject to terms of the company’s debt.

B Our soups are cooked with vegan **stock** and seasonal vegetables.

C In addition, they will receive **stock** in the reorganized company, which will be named Ranger Industries Inc.

Because the contexts between sentence A and sentence C are similar, we can identify that the “*stock*” in sentence 1 and in sentence C has the same meaning because they are both used in a “*company*” setting. Yet to date, the full power of this property has not been exploited for sense disambiguation. Many work derived from this assumption into statistical models based on local and topical features surrounding a target word to be disambiguated. However, even when we make use of these assumptions, it is difficult to identify the sense of common expressions or idioms containing the ambiguous term. For example:

- (1) In an age when personal grievances is all the **rage**, this tale is uninstructive.
- (2) In an age when **rage** powers personal grievances, this tale is uninstructive.

In these above two sentences, there are a word “*rage*” and an idiom “*all the rage*”, as the same time, the context of two sentences are almost the same. If we want to identify the meaning of “*rage*” in these two sentences using similarity context, the result of identification must have the same meaning. According to the above mentioned research, we find supervised systems for WSD often rely upon word collocations (i.e., sense-specific keywords) to provide clues on the most likely sense for a word within the given context. Collocation makes these features more obvious, so supervised learning techniques have generally been found to perform more accurately than knowledge-based methods in “*one sense per collocation*”. As for “*One sense per discourse*”, WSD, a knowledge-based method could obtain more accuracy than other methods. Similarity-based methods determine the sense of a polysemous word (a word with more than one possible meaning) by computing the relatedness between each of its possible senses and the terms in the surrounding context. The correct sense of the ambiguous term is then assumed to be that for which the relatedness is the greatest.

Finally, we find the supervised methods which use an annotated training corpus inducing the appropriate classification models in terms of “*one sense per collocation*”. We also find the relatedness-based method enables it to utilize a higher degree of semantic information, and is more consistent with the properties of “*One sense per discourse*”; that is, by considering the greater context in which the word appears. Because relatedness-based disambiguate all words in a text fragment simultaneously by exploiting semantic relatedness across word senses, it usually achieves higher performance than their supervised alternatives which usually do not consider the senses assigned to surrounding words. To overcome above problems and combine advantages, we propose a hybrid approach for WSD, which combines supervised and relatedness-based methods. This is achieved by combining supervised method and semantic relatedness measures. This approach integrates a diverse set of knowledge sources to disambiguate word sense, including Part-Of-Speech(POS), labeled training data, corpora of unlabeled data, salient neighboring words, and glosses of ambiguous words.

The rest of this paper is organized as follows. Section 2 presents a short review of some earlier works. Section 3 refer to Neural Language Models we have used for our study and which calculate semantic relatedness. Section 4 explains the approach used in this paper and presents the results and corresponding explanations. A discussion of the experimental results is given in Section 5. Finally, section 6 concludes the proposed method.

II. PREVIOUS WORK

For supervised based methods, studies have shown that the word, the word n-gram, the traditional orthographic features, and the POS are the bases for WSD, but they are poor at representing semantic background. In order to incorporate semantic knowledge into an ML model,

Semi-Supervised Learning (SSL) techniques have been applied to WSD. SSL is an Machine Learning (ML) approach that typically uses corpora of unlabeled data and a small amount of labeled data to build a more accurate classification model than would be built using only labeled data. SSL has received significant attention for two reasons. First, preparing a large amount of data for training requires a lot of time and effort. Second, since SSL exploits unlabeled data, the accuracy of classifiers is generally improved. There have been two different directions in SSL methods: 1) semi-supervised learning approaches, which are randomly select a subset of a large unlabeled dataset and classify these samples using one (self-training) or two (co-training) classifiers, trained on a smaller set of labeled samples. After assigning labels to the new samples, these methods select the samples that were classified with a high confidence (according to a selection criterion) and add them to the set of labeled data, and 2) supervised model induction with un-supervised, possibly unlabeled data, feature learning [4]. Yuan [5] also present an algorithm for semisupervised learning, using label propagation to label unlabeled sentences based on their similarity to labeled ones. The approaches in the second research direction induce better feature representation by learning from unlabeled data.

This study extends our previous work in the following ways. First, we propose a hybrid approach for WSD, which combines supervised methods and semantic relatedness. Second, we have presented context2vector which combine neural model learned from a generic embedding function for variable length contexts of target words on a POS labeled text corpus. Third, we explore semantic relatedness algorithms and approach of relatedness feature representation. Finally, we take a small amount of labeled data to build supervised models to avoid being entrapped in the problem of creating annotated corpora. These changes lead to outperforming the results obtained by other systems in the considered competitions.

III. NEURAL LANGUAGE MODELS-BASED SEMANTIC RELATEDNESS

Melamud et al. argues that since contexts induce meanings (or senses) for target words, a good context similarity measure should assign high similarity values to contexts that induce similar senses for the same target word [6]. More recently, in an unsupervised setting, word embeddings were used in measuring context-sensitive similarity to learn internal representations of wider sentential contexts [7]. Therefore neural language models were used to measuring context-sensitive similarity.

A. Context2vec’s Neural Model

Like target words, contexts are commonly represented via word embeddings. In an unsupervised setting, such representations were found useful for measuring context-sensitive similarity , word sense disambiguation [7], [8]. The context representations used in such tasks are commonly just a simple collection of the individual embeddings of the neighboring words in a window around the

target word, or an (sometimes weighted) average of these embeddings. We note that such approaches do not include any mechanism for optimizing the representation of the entire sentential context as a whole. In this work , we present context2vec-based semantic relatedness of method, an unsupervised model for efficiently learning generic context embedding of wide sentential contexts, using bidirectional LSTM. The main goal of our model is to learn a generic task-independent embedding function for variable-length sentential contexts around target words. To do this, we use a neural network architecture, which is based on word2vec’s CBOW architecture, but replaces its naive context modeling of averaged word embeddings in a fixed window, with a much more powerful neural model, using bidirectional LSTM [9]. We use a bidirectional LSTM recurrent neural network to obtain a sentence-level context representation [10]. However, that context representation has a drawback which has not been regarded for POS in input level. Though two words have different POS, they have the same word embeddings. For addressing this problem, in this paper, we proposed an improvement method which adds POS into input level to construct context2vec’s neural model. However, we don’t need fine-grained POS. We have combined some of the tags. In this paper, we tag five kinds of POS only as shown in table 1 right column.

Our proposed architecture is illustrated in Figure 1. This model learn context and target word representations at the same time, by embedding them into the same low-dimensional space, with the objective of having the context predict the target word via a log linear model. We utilize a much more powerful parametric model to capture the essence of sentential context. Figure 1 illustrates how context2vec represents sentential context. We use a bidirectional LSTM recurrent neural network, feeding one LSTM network with the sentence words from left to right, and another from right to left. The parameters of these two networks are completely separate, including two separate sets of left-to-right and right-to-left context word embeddings. To represent the context of an ambiguous term in a sentence (e.g. for “John/NP [submitted/VV] a paper/NN”), we first concatenate the LSTM output vector representing its left-to-right context (“John/NP”) with the one representing its right-to-left context (“a paper/NN”).

Let ILS be an LSTM reading the words of a given sentence from left to right, and let rLS be a reverse one reading the words from right to left. Given a sentence $w_{1:n}$, our ‘shallow’ bidirectional LSTM context representation for the target w_i is defined as the following vector concatenation:

$$biLS(w_{1:n}, i) = lLS(l_{1:i-1}) + rLS(r_{n:i+1}) \quad (1)$$

where l/r represent distinct left-to-right/right-to-left word embeddings of the sentence words.This definition is a bit different than standard bidirectional LSTM, as we do not feed the LSTMs with the target word itself (i.e. the word in position i). With this, we aim to capture the relevant information in the sentential context, even when

it is remote from the target word. Next, we feed this concatenated vector into a multi-layer perceptron to be capable of representing non-trivial dependencies between the two sides of the context.

$$MLP(x) = L_2S(ReLU(L_1(x))) \quad (2)$$

where MLP stands for Multi Layer Perceptron, ReLU is the Rectified Linear Unit activation function, and $L_i(x) = W_i x + b_i$ is a fully connected linear operation. Let $c = w_1, \dots, w_{i-1}, -w_{i+1}, \dots, w_n$ be the sentential context of the word in position i . We define context2vec’s representation of c as:

$$c = MLP(biLS(w_{1:n}, i)) \quad (3)$$

We consider the output of this layer as the embedding of the entire joint sentential context around the target word. At the same time, the target word itself (right-hand side of Figure 2) is represented with its own embedding, equal in dimensionality to that of the sentential context. We note that the only (yet crucial) difference between our model and word2vec’s CBOW (Figure 1) is that CBOW represents the context around a target word as a simple average of the embeddings of the context words in a window around it, while context2vec utilizes a full-sentence neural representation of context.

B. Context2vec-Based Semantic Relatedness

After obtaining context2vec, we need to obtain semantic similarity. We compute semantic similarity metrics in that space of context-to-context. All these are measured by the vector cosine value between the respective embedding representations. To classify a test word instance in context, we consider all of the tagged instances of the same word lemma in the training set, and find the instance whose context embedding is the most similar to the context embedding of the test instance using the context-to-context similarity metric. Then, we use the tagged senses of that instance. We note that this is essentially the simplest form of a k-nearest-neighbor algorithm, with $k = 4$.

$$\arg \max_{1 \leq i \leq k} \text{sim}(c_0, c_i) \quad (4)$$

C. Similarity-Based Feature Representation

Feature representation is very important to supervised methods. Therefore, semantic similarity was transformed into a special feature in this paper. To get context vector, some conditions are to be fulfilled in each of the experiments. These conditions are summarized as follows:

- (1) POS of the disambiguation sentence and instances.
- (2) Containing the POS of notional words and deleting the POS of function words.
- (3) POS of notional words mapping based on the table 1. Disambiguation sentence: *Sodalities have an important role in activating laity for what are judged to be religious goals both personally and socially* . Above sentence after the POS tagging: *Sodalities_NN have an important_JJ role_NN in activating_VV laity_NN for what are*

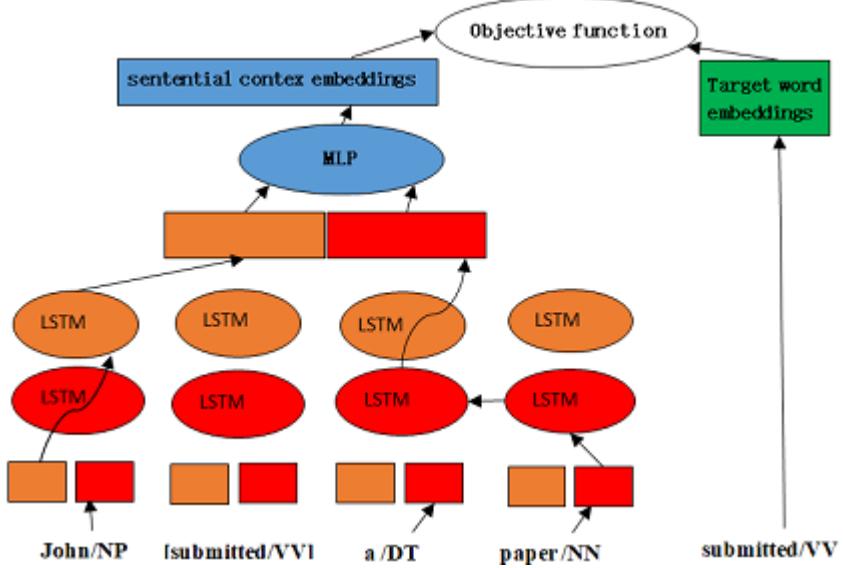


Figure 1. context2vec with POS

Table I
POS MAPPING LIST

RB,RBR,RBS	RB
NNS,NN	NN
NP,NPS	NN
JJS,JJR,JJ	JJ
VV,VVD,VVG,VVN,VVP,VVZ	VV

judged_VV to be *religious_JJ* *goals_NN* both *personally_RB* and *socially_RB*. We implemented the semantic disambiguation as a centroid-based classifier, which, given a disambiguation sentence, ranks all the instances by their relevance to the sentence. Given a disambiguation sentence, we first represent it as a vector using context2vec. To compute semantic relatedness of a pair of sentences we compare their vectors using the formulation 4. In table 2 the second column are instances tagged sense, the third column are senses id and the forth column are weight of every sense. If some word correspond to two senses, the weight would signed 0.5. We measure senses of disambiguation word in all instances based on formulation 5.

$$Score(S_i) = T * weight \quad (5)$$

Where S_i is one of sense of the disambiguation word, T is the number of times which sense appeared in the top 4 instances, and weight is the same as weight in the table 2. For example, “active” totally has five senses. Based on the formulation 5 the score of every sense as follow:

- $Score(S_1) = 1.5$
- $Score(S_2) = 2.0$
- $Score(S_3) = 0.5$
- $Score(S_4) = 0.0$
- $Score(S_5) = 0.0$

We take the score as sense feature in the supervised learning. As our supervised WSD dataset, we used the Senseval-3 lexical sample dataset, denoted SE-3, which

includes 7,860 train and 3,944 test instances. We used the training set for instance corpus and report accuracy results on the test set [11].

IV. METHODS

The machine learning tool that the supervised system used for word sense disambiguation is Conditional Random Fields (CRFs). We extract types of features and then use CRFs as the classifier. The features and templates implemented in our system are explained below section 4.2. After extracting these features, the classifier (CRFs) is used to train a model for the same number sense of target word. In the test phase, the model is used to classify test samples and to assign a sense tag to each sample.

A. Preprocessing

We used the two billion word ukWaC as our learning corpus [12]. We use TreeTagger to tag POS [13]. To speed-up the training of context2vec, we discarded all sentences that are longer than 64 words, reducing the size of the corpus by 10%. We lower-cased all text and considered any token with fewer than 100 occurrences as an unknown word. This yielded a vocabulary of a little over 180K words for the full corpus, and 160K words for the trimmed version.

B. Feature and Template

The word representation feature is essential to classifier, but it is poor because it only carries some morphological and shallow-syntax information of words. However, the sense representation features can be extracted by calculating semantic relatedness and may be capable of introducing sense knowledge background to the WSD model. We take the score of sense as sense feature in section 3.2, and then an ML algorithm is employed to build a model for WSD. We applied features and templates in ML algorithm as follows: In table3, t represents words in

Table II
FIRST 4 INSTANCES IN INSTANCE CORPUS

#	Input: disambiguation sentence	Sense id	weight
1	You step_VV on to activate_VV it .	38201	1.0
2	Which parts_NN of the sensory_JJ system_NN are activated_VV.	38202	1.0
3	Different_JJ genes_NN are activated_VV in different_JJ cells_NN .	38203	1.0
4	This clause_NN has never_RB yet_RB been activated_VV .	38201 38203	0.5

Table III
FEATURES AND TEMPLES OF CRFS TRAINING

feature	temple
t	t_0
score	$score(s_j) (1 \leq j \leq m)$
$t + score$	$t_0 score(s_j)$
POS	p_0
others	$t_{-4} t_{-3} t_{-2} t_{-1} t_0 t_1 t_2 t_3 t_4, p_{-4} p_{-3} p_{-2} p_{-1} p_0 p_1 p_2 p_3 p_4$

Table IV
TAGGING FORMAT OF CRFS TRAINING SET

term	POS	F1	F2	F3	F4	Fi	tag
You	PP	0	0	0	0	0	O
step	VV	0	0	0	0	0	O
on	IN	0	0	0	0	0	O
to	RB	0	0	0	0	0	O
activate	VV	1.5	2.0	0.5	0.0	0.0	F1
it	PP	0	0	0	0	0	O
.	.	0	0	0	0	0	O

a given sentence W , $score$ represents semantic similarity score(details see section 3.3), $t+score$ represents the combination feature, t_0 represents the disambiguation word, t_{-1} is the word preceding t_0 , t_1 is the word following t_0 , $score(s_j)$ represents one sense of t_0 , p represents POS of word.We compute semantic similarity metrics in the space of context-to-context. All these are measured by the vector cosine value between the respective embedding representations.

C. Semi-Supervised Learning

We apply the CRFs tools to build WSD model. In this paper, tagging format of CRFs training is shown in table4. In table4, F1, F2, F3 and F4 denote the feature of every sense based on the score . Fi denotes i^{th} score of sense.

D. Experiments Results

In this section, we study the performance of our classifiers on SE3 lexical sample task, SemEval-2013 Task 12 and SemEval-2015 task 13 [14], [15]. For experiment results, we report F1 score (Navigli, 2009). In table 5, we refer to Context2vec based sense relatedness as ‘CBSR’. In Table 5, we compared our overall F1 scores with different systems which include An-do [16], Rothe [17], Melamud [10], Grozea [18]. The best performance is achieved when we combine supervised learning (CRFs) and CBSR. Ando achieved high performance using Alternating Structure Optimization that is a very complicated method compared to ours. Rothe has a system to learn embeddings for synsets and lexemes. Mel-maud learn context embeddings

Table V
COMPARISON OF OUR SYSTEM WITH OTHER SYSTEMS ON SENSEVAL-3 LEXICAL

System	Micro-average
CRFs (baseline)	62.0
CBSR	75.3
CRFs+CBSR	75.9
Ando	74.1
Rothe	73.6
Mel-amud	72.8
Grozea	72.9

of a word and classify a test word instance with the sense of the training set word whose context embedding is the most similar to the context embedding of the test instance. The method of Mel-maud is similar to CBSR, but our method adds the POS feature into the context embedding. Grozea is the best system in Senseval-3. As can be observed from Table 6, we did see significant improvements using CBSR with the CRFs model. In Table 6, the improvements obtained by using Semcor with OMSTI as training data over Semcor only are significant.

V. DISCUSSION

Supervised word sense disambiguation systems usually treat words as discrete entities and consequently ignore the concept of relatedness between words. However, by adding sense relatedness, some of the samples that cannot be discriminated basing on the original features (surrounding words, long distance dependency) have more chances to be classified correctly. Moreover, sense relatedness contain valuable linguistic information too. Hence, adding representations of sense can provide valuable information to the classifier and the classifier can learn better discriminative criteria based on such information. Our approach to WSD does not rely on large labeled data sets. Instead, it leans on supervised models learned from small labelled data sets, on representations of sense relatedness learned from structured semantic resources and on medium unlabeled corpora. This enables us to exploit how to integrate the semantic knowledge of word in the framework of WSD.

VI. CONCLUSION

We have presented sense relativity algorithms which combine neural model learned from a generic embedding function for variable length contexts of target words on a POS labeled text corpus, with sense-labeled data in the form of example sentences. Meanwhile, corpora resource provides collocation occurrences of a target word that

Table VI
EVALUATION RESULTS OF DIFFERENT RUNS WITH VARIED APPROACH ON ALL-WORDS DATASETS OF SEMEVAL-2013 AND SEMEVAL-2015

Method	SemEval-13 Semcor	SemEval-13 Semcor+OMSTI	SemEval-15 Semcor	SemEval-15 Semcor+OMSTI
CRFs (baseline)	46.7	52.3	50.1	58.7
CBSR	64.4	65.8	69.5	70.3
CRFs+CBSR	64.5	66.0	69.9	70.6

cannot be gained from sense inventory resources and context embedding. Semi-supervised WSD systems generate high quality WSD result using CBSR, as it may carry more useful information learned from large corpora than other methods. Our system that solves WSD would enable corpora resource to take full advantage of their enlightened decision to incorporate sense knowledge into supervised learning method.

ACKNOWLEDGMENT

The authors would like to thank the Humanities and Social Sciences of Ministry of Education Planning Fund (18YJA870020) for their support in this research.

REFERENCES

- [1] R. Navigli, “Word sense disambiguation: A survey,” *ACM Computing Surveys*, vol. 41, no. 2, pp. 1–69, Feb. 2009.
- [2] W. A. Gale, K. W. Church, and D. Yarowsky, “One sense per discourse,” in *Proceedings of the Workshop on Speech and Natural Language*, ser. HLT ’91. Association for Computational Linguistics, 1992.
- [3] D. Yarowsky, “One sense per collocation,” in *Proceedings of the Workshop on Human Language Technology*, ser. HLT ’93. Association for Computational Linguistics, 1993.
- [4] K. Taghipour and H. T. Ng, “Semi-supervised word sense disambiguation using word embeddings in general and specific domains,” in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2010.
- [5] D. Yuan, R. Doherty, J. Richardson, C. Evans, and E. Altdorf. (2016) Word sense disambiguation with neural language models.
- [6] O. Melamud, I. Dagan, and J. Goldberger, “Modeling word meaning in context with substitute vectors,” in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2015.
- [7] E. Huang, R. Socher, C. Manning, and A. Ng, “Improving word representations via global context and multiple word prototypes,” in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2012.
- [8] X. Chen, Z. Liu, and M. Sun, “A unified model for word sense representation and disambiguation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2014.
- [9] T. Mikolov, K. Chen, G. Corrado, and J. Dean. (2013) Efficient estimation of word representations in vector space.
- [10] O. Melamud, J. Goldberger, and I. Dagan, “context2vec: Learning generic context embedding with bidirectional lstm,” in *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 2016.
- [11] R. Mihalcea, T. Chklovski, and A. Kilgarriff, “The senseval-3 english lexical sample task,” in *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, 2004.
- [12] A. Ferraresi, E. Zanchetta, M. Baroni, and S. Bernardini, “Introducing and evaluating ukwac, a very large web-derived corpus of english,” in *In Proceedings of the 4th Web as Corpus Workshop (WAC-4)*, 2008.
- [13] TreeTagger, “TreeTagger - a part-of-speech tagger for many languages.”
- [14] R. Navigli, D. Jurgens, and D. Vannella, “Semeval-2013 task 12: Multilingual word sense disambiguation,” in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Association for Computational Linguistics, 2013.
- [15] A. Moro and R. Navigli, “Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking,” in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, 2015.
- [16] R. K. Ando, “Applying alternating structure optimization to word sense disambiguation,” in *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*. Association for Computational Linguistics, 2006.
- [17] S. Rothe and H. Schütze, “Autoextend: Extending word embeddings to embeddings for synsets and lexemes,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, 2015.
- [18] C. Groza, “Finding optimal parameter settings for high performance word sense disambiguation,” in *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Association for Computational Linguistics, 2004.

Acquisition of Knowledge with Time Information from Twitter

Kohei Yamamoto

Department of Artificial Intelligence

Kyushu Institute of Technology

680-4 Kawazu Iizuka Fukuoka 820-8502 Japan

Email: k_yamamoto@pluto.ai.kyutech.ac.jp

Kazutaka Shimada

Department of Artificial Intelligence

Kyushu Institute of Technology

680-4 Kawazu Iizuka Fukuoka 820-8502 Japan

Email: shimada@pluto.ai.kyutech.ac.jp

Abstract—In this paper, we propose a knowledge acquisition method for non-task-oriented dialogue systems. Such dialogue systems need a wide variety of knowledge for generating appropriate and sophisticated responses. However, constructing such knowledge is costly. To solve this problem, we focus on a relation about each tweet and the posted time. First, we extract event words, such as verbs, from tweets. Second, we generate frequency distribution for five different time divisions: e.g., a monthly basis. Then, we remove burst words on the basis of variance for obtaining refined distributions. We checked high ranked words in each time division. As a result, we obtained not only common sense things such as “sleep” in night but also interesting activities such as “recruit” in April and May (April is the beginning of the recruitment process for the new year in Japan.) and “raise the spirits/plow into” around 9 AM for inspiring oneself at the beginning of his/her work of the day. In addition, the knowledge that our method extracts probably contributes to not only dialogue systems but also text mining and behavior analysis of data on social media and so on.

Keywords-Knowledge acquisition; Text mining; Hourly things; Daily things; Weekly things; Monthly things;

I. INTRODUCTION

Recently, dialogue systems, especially non-task-oriented dialogue systems, have been increasingly important and popular, such as Rinna¹ (Microsoft) and XiaoIce² (Microsoft). In addition, task-oriented dialogue systems, such as QA systems, often contain a chatting function for increasing user satisfaction: e.g., Siri³ (Apple) and Pepper⁴ (Softbank). The success of neural networks or reinforcement learning approaches improves performance dramatically. However, these models tend to generate simple responses and are not sufficient to satisfy users. The main problem is caused by a lack of knowledge that such dialogue systems retain. When we talk to someone about something, we often utter not only direct responses and answers to what he/she talks but also implicit information that we share, namely a matter of common sense. To realize a human-like conversation by dialogue systems, we need to acquire much knowledge related to experiences and events in real life. Narisawa

et al. [1] have proposed an automatic acquisition method about numerical common sense from the Web. Young et al. [2] have proposed a method for generating utterances by using common sense knowledge. Many researchers have studied knowledge acquisition methods from Web news [3], asynchronous dialogues from Ubuntu forums [4], [5] and Reddit [6], [7], and Wikipedia [8]. Some researchers have also handled Twitter as the resource for dialogue systems [9], [10], [11]. Such knowledge and approaches lead to the improvement of user satisfaction for dialogue systems.

In this paper, we propose a method for extracting knowledge about daily occurrences from social media. We focus on Twitter as the social media service. It is one of the most famous microblogging services and text-based posts of up to 140 characters. The posted sentences are described as “tweet”. In microblogging services such as Twitter, users tend to post tweets in real-time. It denotes that tweets often contain significant information about daily occurrences, as lifelog data.

The purpose of our study in this paper is to automatically extract daily, weekly, and monthly activities with time information from Twitter. In other words, our purpose is to link an event and time information, e.g., “snowfall” appears in winter (around December to February in Japan) and “going to bed” frequently occurs in approx. 10 p.m. to 12 p.m. Here we explain the importance of the knowledge for dialogue systems with some examples.

- If a system retains knowledge, “eating lunch” = around 12 a.m., the system can generate the following utterance by using the current time.

Ex1 (2/22 12:30)

System: Have you eaten lunch?

User: Yes, I have.

- If a system retains knowledge that “swimming” is an event in summer, the system can generate the following utterance by using the current month.

Ex2 (2/22 15:00)

User: I went swimming in the sea yesterday.

System: Oh! It is uncommon to swim in February, isn’t it?

- If a system retains knowledge about the peak of cherry blossoms, the system can generate the following utter-

¹<https://www.rinna.jp/>

²<https://www.msxiaobing.com/>

³<https://www.apple.com/siri/>

⁴<https://www.softbank.jp/robot/pepper/>

ance by using the current date.

Ex3 (3/15 12:00)

User: Spring is coming soon.

System: I would like to go on a cherry viewing picnic.

These utterances can be realized by the knowledge about relations between an event and the most relevant time.

One simple approach to realize the system is to construct rules by handwork. However, the construction is costly and often possesses a problem of the coverage of rules. Moreover, Higashinaka et al. [12] have reported that manual modification and extension of the rules do not always tend to lead the improvement of the system's performance. On the other hand, the contribution of this paper is that our method links a word/phrase with time information automatically. In addition, the knowledge that our method extracts probably contributes to not only dialogue systems but also text mining and behavior analysis of data on social media and so on. Furthermore, our method does not depend on specific languages although we handle Japanese tweets in this paper because the method is just based on rankings by frequency distribution and burst detection. To the best of our knowledge, this is the first research about knowledge acquisition with time information from Twitter

II. RELATED WORK

As a study about handling time information for dialogue systems, Sato et al. [13] have proposed a neural chatbot system with knowledge about season type information. In other words, they proposed a method handling a variety of situations that affect the system outputs. However, they used only seasons, namely spring (Mar. - May.), summer (Jun. - Aug.), autumn (Sep. - Nov.), and winter (Dec. - Feb.) as the time information. On the other hand, we apply various types of time information, such as daily and monthly, to our knowledge acquisition task.

For knowledge acquisition, many researchers have reported their approaches [1], [14], [15], [16]. As mentioned above, Narisawa et al. [1] proposed an automatic acquisition method about numerical common sense from the Web. It must be useful knowledge for dialogue systems. Mitsuda et al. [14] have collected and clustered information that humans perceive from each utterance by handwork. Machida et al. [15] and Otani et al. [16] have proposed knowledge acquisition methods via dialogue systems. On the other hand, we use tweets on Twitter for knowledge acquisition.

The purpose of our study is to extract knowledge about hourly, daily, weekly, monthly occurrences. Ge et al. [17] have constructed a resource, EventWiki, from Wikipedia. Their target is major events, such as earthquakes and Olympic events. It is different from the target in our task. As studies focusing on relations between human activity and time information, Tandon et al. [18] have proposed an acquisition method of knowledge about activities from narratives, such as movies. Yao et al. [19] have proposed a

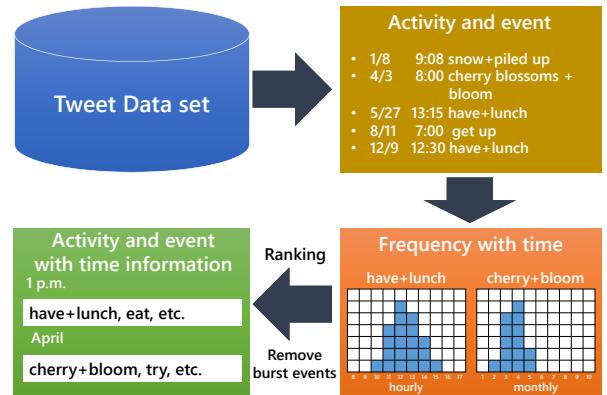


Figure 1. Overview of our method.

method for acquiring rich temporal “before/after” event knowledge across sentences in narrative stories. In this paper, we propose an acquisition method for a wide range of knowledge from Twitter as compared with the related work.

III. METHOD

In this section, we describe our acquisition method. Figure 1 shows the outline of the method. First, we extract activity and event word/phrases from a tweet data set (Section III-A.) Next, we generate the frequency distribution of each activity and event word/phrase on the basis of the timestamp of each tweet (Section III-B.) Then, we generate the rankings of each activity and event word/phrase (Section III-C.) Finally, we remove burst words by using a burst detection process (Section III-D.) As a result, we obtain activities and events that appear in a specific time range: e.g., “have lunch” at 1 p.m. in Figure 1.

A. ActEvn word extraction

Tweets tend to contain mention about users' real-time activities or events that users attend. In general, verbs have the most important role in such mention. In addition, the object of each verb also has an important role in recognizing the content of activities and events. Therefore, we extract verbs and verb-object pairs from tweets (hereinafter this is called “ActEvn words”).

We use a morphological analyzer, MeCab⁵, for the ActEvn word extraction. Tweets tend to contain new words and proper nouns that are not included in normal dictionaries. Therefore, we also use the NEologd dictionary⁶ to solve this problem.

B. Frequency distribution

Then, we generate frequency distribution tables of ActEvn words. In this paper, we focus on the appearance of ActEvn

⁵<http://taku910.github.io/mecab/>

⁶<https://github.com/neologd/mecab-ipadic-neologd>

Table I
TIME PERIODS.

Type	Unit for counting
Month	Monthly (Jan., Feb., Dec.)
Week	Weekday (Mon. to Fri.) and Weekend (Sat. and Sun.)
Day-of-Week	Each day of the week (Sun. to Sat.)
Day	Morning (3 to 8 a.m.), Daytime (9 a.m. to 5 p.m.), Night (6 p.m. to 2 a.m.)
Hour	Hourly (1 a.m., 2 a.m., ..., 12 p.m.)

Activity / Event (ActEvn word)	Freq. All	Monthly frequency (Ratio)					
		Jan.	Feb.	Mar.	Apr.	...	Dec.
Cherry blossoms + bloom	200	0 (0%)	10 (5%)	80 (40%)	80 (40%)	...	0 (0%)
snow + piled up	1000	200 (20%)	200 (20%)	10 (1%)	0 (0%)	...	200 (20%)
medicine + work	5	3 (60%)	1 (20%)	1 (20%)	0 (0%)	...	0 (0%)
...

- Filter out by the total frequency
 → Assume that the threshold is 100.
 In this case, “medicine+work” is deleted from the ranking
 • Ranking by the ratio

Time	Activity / Event (ActEvn word)
Jan.	Snow + fall, Snow + piled up, ...
...	...
Apr.	Cherry blossoms + bloom, Go + picnic, ...
...	...

Figure 2. Ranking by the ratio that computed from the total frequency.

words in several time periods for the knowledge acquisition with temporal information. For the purpose, we need a suitable definition of time periods. We utilize the time unit defined by the Japan meteorological agency.

Table I shows the definition and the time ranges: units that our method sums up by. We handle five types of time ranges. We compute the frequency of each ActEvn word for each type of time range. The upper part of Figure 2 shows an example of frequency distribution tables about the Month type, namely a monthly distribution.

C. Ranking

Typical activities and events closely related to a specific time lead to a massive amount of posts, namely tweets, about the activities and events. For example, the number of tweets that contain the ActEvn word “have+lunch” increases around noon. In a similar way, tweets with “Cherry blossom+bloom” frequently occur in March and April in Japan. Therefore, we generate rankings about each time range type from the distribution tables obtained in Section III-B.

Figure 2 shows an example of a ranking process. The ranking process is based on the ratio of each time range. Our method sorts each ActEnv word in descending order as

Activity / Event (ActEvn word)	Time	Freq. All	Freq. of each day (Ratio)					
			1/1	...	6/24	6/25	...	12/31
sleep	1 a.m.	1900	4 (0.20%)	...	6 (0.31%)	4 (0.20%)	...	5 (0.25%)
Watch+FIFA World Cup	1 a.m.	1000	0 (0%)	...	700 (70%)	100 (10%)	...	0 (0%)
...

Compute the variance of each ActEvn word on ratio

Activity / Event (ActEvn word)	Variance
sleep	0.0057
Watch+FIFA World Cup	13.95
...	...

Large variance indicates a burst event
 → Remove the activity / event from the list

- “sleep” is a daily activity around 1 a.m.
- “Watch FIFA World Cup” is a burst event because it frequently occurred in only a few days

Figure 3. Deletion of burst activities and events by using the variance value.

the ranking of each type: Year, Week, Day-of-Week, Day, and Hour explained in Table I. In the process, we use a filter based on the total frequency of each ActEvn word because low-frequency words usually should be ignored for the real ActEnv word detection: e.g., “medicine+work” in Figure 2. For example, the threshold in Figure 2 is 100. The threshold value of the filter is determined experimentally.

D. Burst detection and removal

In Twitter, a phenomenon in which many users post tweets occurs at the same time. It is called “burst”. In other words, the burst is a point in which the number of tweets suddenly increases. It usually occurs in out-of-ordinary events: e.g., FIFA World Cup and Olympic games. Such burst situations in Twitter should be ignored in our knowledge acquisition process because our purpose is to extract common activities of people and ordinary events in the world. Therefore, we need to remove ActEvn words related to the burst situation.

In this paper, we focus on distribution uniformity of each ActEvn word. If an ActEvn word is not a burst word, the distribution of the word frequency becomes uniform. In other words, the variance of ratios based on the frequency becomes small. On the other hand, if an ActEvn word is a burst word, that becomes non-uniform. In other words, the variance of ratios based on the frequency becomes large.

Figure 3 shows an example of burst and non-burst ActEvn words. The word “sleep” is an ActEvn word and a daily activity. Hence, the variance is sufficiently small. On the other hand, “Watch+FIFA World Cup” is not an ActEvn word because it is a temporary event at 1 a.m. However, monthly time range is excluded from the burst detection process due to a lack of enough yearly data because we need tweets of several years to compute the statistically unwarranted variance.

Table II

MONTH RANKING WITHOUT BURST DETECTION. THE THRESHOLD VALUES OF THE FREQUENCY IS 1000.

Month	1st	2nd	3rd	4th
Jan	積もる (snow cover)	雪+降る (snowfalls)	お願い+いたす (please)	お願い+致す (please)
Feb	積もる (snow cover)	雪+降る (snowfalls)	渡す (give)	受かる (pass)
Mar	卒業+する (graduate)	受かる (pass)	狙う (target)	染める (dye)
Apr	募集+する (recruit)	うつ (depression)	病む (worry about)	フォロー+する (follow)
May	黙る (be silent)	募集+する (recruit)	掘る (dig)	つづく (continue)
Jun	揺れる (shake)	勝つ (win)	追いつく (catch)	攻める (attack)
Jul	溶ける (melt)	浴びる (shower)	雨+降る (rain falls)	刺す (bite)
Aug	鳴る (sound/thunder)	掘る (dig)	雨+降る (rain falls)	刺す (bite)
Sep	発表+する (announce)	合わせる (fit)	刺す (bite)	晴れる (fine weather)
Oct	当選+する (get elected)	風邪+ひく (catch cold)	晴れる (fine weather)	風邪+引く (catch cold)
Nov	晒す (expose)	風邪+ひく (catch cold)	風邪+引く (catch cold)	冷える (get cold)
Dec	雪+降る (snowfalls)	掃除+する (clean up)	実家+帰る (homecoming)	迎える (to start the new year)

IV. DISCUSSION

We analyzed the outputs from our method. First, we discuss the longest and shortest time ranges in our setting: monthly and hourly. Table II shows the top 4th about monthly ranking. We can see intuitive seasonal events from the table, such as “snowfalls” in the winter season. In addition, we obtained many interesting ActEvn words for the monthly ranking. In Japan, February and March are an entrance exam period for high schools and universities. As a result, the ActEvn word “pass” frequently appeared in these months. April is the beginning of a new fiscal year and the beginning of the recruitment process for the next year in Japan. Therefore the word “recruit” became the 1st rank in April. The words “depression” and “worry about” probably express the sentiments and feelings of new employees. The word “bite” indicates that July, August, and September are mosquito-infested months. The words “clean up”, “homecoming”, and “to start the new year” in December express the Japanese culture: the whole house cleaning and staying in parents’ home at the end of the year. On the other hand, some unsuitable ActEvn words appeared on the list. The appearance of “get elected” in October as the 1st ActEvn word is caused by the national election that was held in this month of the year that we collected tweets from Twitter. It is not a habitual activity that we want to acquire. One reason is that we were not able to apply the burst detection process, namely removal by variance, due to a lack of yearly tweets. To improve the correctness and validity of the ranking list, we need to capture tweets with a long-term plan.

Table III shows the top 4th about hourly ranking with burst detection. The threshold values of frequency and burst

are 100 and 1, respectively. We can observe typical hourly actions from the table, such as “sleep” during midnight time, “eating lunch” around noon, and “eating dinner” in the evening. In other words, we obtained a good ActEvn word list that related to common-sense. In addition, we also obtained some interesting ActEven words. One example is the 1st ActEvn word at 9 AM, “raise the spirits/plow into”. It probably indicates that he/she posts the tweet for inspiring oneself at the beginning of his/her work of the day. The word “do the best today” at 7 and 8 AM also contains a similar meaning.

Here we imagine an application of the knowledge that was acquired by our method in a similar way to Section I, namely a chat dialogue system with the knowledge.

- If the system retains knowledge, “bite:mosquito-infested” = Jul. to Sep., the system can generate “Be careful about mosquitoes when you go out.”
- If a system retains knowledge that “raise the spirits/plow into” is around 9 AM and the current time is 8 to 9 AM, the system can generate “Plow into your work! Do your best and good luck.”

Thus, we might well be able to develop an extremely expressive dialogue system in the future.

Table IV, Table V, and Table VI show the results of the weekly ranking, the day-of-week ranking, and the daily ranking, respectively. We obtained similar tendencies between the week ranking and the day-of-week ranking. For example, “go to school/hospital/work” on weekdays and “go out” on weekends. The result of the daily ranking was also similar to the hour ranking shown in Table III.

Here we focus the effectiveness of the burst detection and the removal. Table VII shows a part of a result by our method without the burst detection for the hourly ranking. The table contained many noise words as compared with that using the burst detection process (see Table III.) In addition, the method without the bust detection acquired non-habitual activities, such as “watch fireworks” at 8 PM as the 4th. These results show the effectiveness of burst detection.

Words in each ranking often accompany auxiliary verbs about report and conjecture markers, such as “らしい (It is said that)” and “みたい (It seems that).” Some researchers have studied fact analysis methods [20], [21]. Incorporating the methods into our method is interesting future work.

In this paper, we evaluated our result qualitatively. To provide a quantitative evaluation by test subjects is the most important future work.

V. CONCLUSIONS

In this paper, we proposed a method for extracting knowledge about daily occurrences from social media. We focused on Twitter as the social media service. Users in Twitter tend to post tweets in real-time. We handled tweets as lifelog data by using this characteristic. First, we extracted event words (ActEvn words) from tweets. Then, we generated

Table III

HOUR RANKING WITH BURST DETECTION. THE THRESHOLD VALUES OF THE FREQUENCY AND BURST ARE 100 AND 1, RESPECTIVELY.

Hour	1st	2nd	3rd	4th
0 AM	お誕生+ござる (be birthday)	夢+見れる (be able to have a dream)	明日+起きる (wake up tomorrow)	5 時+起きる (wake up at five)
1 AM	ねむれる (be able to sleep)	仕事+寝る (sleep in working)	ねれる (be able to sleep)	ため+寝る (sleep for)
2 AM	これ+寝る (this sleep)	の+眠れる (be able to sleep)	ねれる (be able to sleep)	目+冴える (be wakeful)
3 AM	目+冴える (be wakeful)	ねれる (be able to sleep)	時間+起きる (wake up at)	目+さめる (wake up)
4 AM	時間+起きる (wake up at)	就寝+する (go to bed)	2 時間+寝る (sleep two hours)	の+眠れる (be able to sleep)
5 AM	表す (express)	お過ごし+くださる (stay)	指す (point)	4 時+起きる (wake up at 4)
6 AM	使用+する (use)	予測+する (predict)	お過ごし+くださる (stay)	結局+寝る (finally sleep)
7 AM	日+がんばる (do the best today)	今日+頑張る (do the best today)	朝+迎える (in the morning)	今週+頑張る (do the best this week)
8 AM	日+がんばる (do the best today)	今週+頑張る (do the best this week)	つづく (continue)	遅延+する (delay)
9 AM	はりきる (raise the spirits/ plow into)	遅延+する (delay)	病院+来る (go to hospital)	元気+過ごす (keep well)
10 AM	瞬+殺る (in a flash)	洗濯+終わる (finish washing)	電話+くる (get a phone call)	朝ごはん+食べる (have breakfast)
11 AM	病院+来る (go to hospital)	昨日+行う (do yesterday)	マック+食べる (eat McDonald's)	昼ごはん+食べる (have lunch)
12 PM	昼ご飯+食べる (have lunch)	弁当+食べる (have lunch)	飯+食べる (eat lunch)	瞬+殺る (in a flash)
1 PM	昼ごはん+食べる (have lunch)	昼飯+食う (have lunch)	昼ご飯+食べる (have lunch)	お昼+食べる (have lunch)
2 PM	昼ごはん+食べる (have lunch)	用事+済ませる (finish a job)	先+ある (be ahead)	買い+来る (go buy)
3 PM	モード+なる (go into work mode)	仕事+戻る (go back to work)	昼飯+食べる (have lunch)	バイト+休む (get off a part-time job)
4 PM	バイト+行く (go to a part-time job)	昼+食べる (have lunch)	夕方+なる (in the evening)	電車+座る (get a seat on a train)
5 PM	夕方+なる (in the evening)	定期+帰れる (leave work on time)	図+買う (buy books)	駆る (punch)
6 PM	選択+する (select)	夕飯+作る (make dinner)	夕飯+食べる (have dinner)	定期+上がる (leave work on time)
7 PM	閉じる (close)	ご飯+炊く (cooking rice)	自炊+する (cook own meal)	お仕事+終わる (finish the work)
8 PM	与える (give)	晩御飯+食べる (have dinner)	チケット+届く (get a ticket)	点+入る (get a score)
9 PM	明日+届く (get tomorrow)	収束+する (settle down)	明日+楽しむ (enjoy tomorrow)	ドラム+叩く (beat a drum)
10 PM	ふむ (step)	心+響く (touch a heart)	湯船+浸かる (get in a bath)	いらっしゃる (coming)
11 PM	明日+寝る (sleep for tomorrow)	人+飲む (drink with)	どっか+見る (watch something)	時間+食べる (eat late at night)

Table IV

WEEK RANKING (WEEKDAY AND WEEKEND) WITH BURST DETECTION. THE THRESHOLD VALUES OF FREQUENCY AND BURST ARE 1000 AND 1, RESPECTIVELY.

Week	1st	2nd	3rd	4th
weekday	学校+行く (go to school)	サボる (wag off)	病院+行く (go to hospital)	今日+頑張る (do my best today)
weekend	出かける (go out)	並ぶ (get in line)	参加+する (join)	向かう (go to)

frequency distributions for five different time divisions, e.g., a monthly basis. We introduced burst word detection on the basis of variance for obtaining refined distributions. As a result, we obtained not only common sense things such as in night but also interesting activities such as “raise the spirits/plow into” around 9 AM for inspiring oneself at the beginning of his/her work of the day. Although we handled Japanese tweets in this paper, our method does not

Table V

DAY-OF-WEEK RANKING WITH BURST DETECTION. THE THRESHOLD VALUES OF FREQUENCY AND BURST ARE 1000 AND 1, RESPECTIVELY.

DoW	1st	2nd	3rd	4th
Mon	学校+行く (go to school)	書ける (write)	揺れる (swing)	今日+寝る (sleep today)
Tue	使用+する (use)	掛かる (hang)	つづく (continue)	黙る (shut)
Wed	つづく (continue)	晒す (expose)	曇る (cloudy)	つぶやく (tweet)
Thu	相談+する (have a talk)	腹立つ (be angry)	サボる (wag off)	今日+寝る (sleep today)
Fri	いらっしゃる (come)	使用+する (use)	今日+頑張る (do my best today)	当選+する (get elected)
Sat	出かける (go out)	並ぶ (get in line)	売り切れる (sold out)	呑む (drink)
Sun	充実+する (fulfilling)	浸る (soak)	参加+する (join)	出掛ける (go out)

depend on specific languages. This is also the effectiveness of our method. The result often contained noise words in

Table VI
DAY RANKING (MORNING, DAYTIME, NIGHT) WITH BURST DETECTION. THE THRESHOLD VALUES OF FREQUENCY AND BURST ARE 1000 AND 1, RESPECTIVELY.

Day	1st	2nd	3rd	4th
3 to 8	今日+頑張る (do my best today)	寝坊+する (oversleep)	仕事+行く (go to work)	早起き+する (early rising)
9 to 17	昼寝+する (napping)	売り切れる (sold out)	出かける (go out)	混む (jam-up)
18 to 2	今日+寝る (sleep)	風呂+入る (take a bath)	眠れる (sleep)	酒+飲む (drink)

Table VII
HOUR-RANKING WITHOUT BURST DETECTION. THE THRESHOLD VALUES OF FREQUENCY IS 100.

Hour	1st	2nd	3rd	4th
4 PM	%+見る (watch)	白猫+遊ぶ (play a game)	それ+合わせる (fit)	抽選+当たる (win a lottery)
8 PM	今日+閉じる (close)	逢う (meet)	声+変わる (voice-change)	花火+見る (watch fireworks)

the list, namely non-habitual activities. The improvement of the method using another burst detection approach and fact analysis is important future work.

The current method only focused on five-time ranges. We need to combine the time ranges, such as 24 hours in a month, for extracting much rich knowledge. In addition, user attributions are interesting features for the knowledge acquisition process: e.g., the difference between 24 hours of young men and old men. It is our important future work. Applying our extracted knowledge to dialogue systems and text mining systems is also our future work.

REFERENCES

- [1] K. Narisawa, Y. Watanabe, J. Mizuno, N. Okazaki, and K. Inui, “Is a 204 cm man tall or small? acquisition of numerical common sense from the web,” in *Proceedings of ACL*, vol. 1, 2013, pp. 382–391.
- [2] T. Young, E. Cambria, I. Chaturvedi, H. Zhou, S. Biswas, and M. Huang, “Augmenting end-to-end dialogue systems with commonsense knowledge,” in *AAAI Conference on Artificial Intelligence*, 2018.
- [3] K. Yoshino and T. Kawahara, “Conversational system for information navigation based on pomdp with user focus tracking,” *Computer Speech & Language*, vol. 34, no. 1, pp. 275–291, 2015.
- [4] R. Lowe, N. Pow, I. Serban, and J. Pineau, “The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems,” in *Annual Meeting of SIGDIAL*, 2015, pp. 285–294.
- [5] R. T. Lowe, N. Pow, I. V. Serban, L. Charlin, C.-W. Liu, and J. Pineau, “Training end-to-end dialogue systems with the ubuntu dialogue corpus,” *Dialogue & Discourse*, vol. 8, no. 1, pp. 31–65, 2017.
- [6] P.-E. Mazare, S. Humeau, M. Raison, and A. Bordes, “Training millions of personalized dialogue agents,” in *Proceedings of EMNLP*, 2018, pp. 2775–2779.
- [7] A. C. Curry, I. Papaioannou, A. Suglia, S. Agarwal, I. Shalymov, X. Xu, A. Eshghi, I. Konstas, V. Rieser *et al.*, “Alana v2: Entertaining and informative open-domain social dialogue using ontologies and entity linking,” *Proc. Alexa Prize*, 2018.
- [8] K. Niina and K. Shimada, “Trivia score and ranking estimation using support vector regression and ranknet,” *Proceedings of PACLIC*, 2018.
- [9] J. Li, M. Galley, C. Brockett, G. Spithourakis, J. Gao, and B. Dolan, “A persona-based neural conversation model,” in *Annual Meeting of the ACL*, vol. 1, 2016, pp. 994–1003.
- [10] Y. Luan, C. Brockett, B. Dolan, J. Gao, and M. Galley, “Multi-task learning for speaker-role adaptation in neural conversation models,” in *Eighth International Joint Conference on Natural Language Processing*, vol. 1, 2017, pp. 605–614.
- [11] I. V. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. Courville, and Y. Bengio, “A hierarchical latent variable encoder-decoder model for generating dialogues,” in *AAAI Conference on Artificial Intelligence*, 2017.
- [12] R. Higashinaka, T. Meguro, H. Sugiyama, T. Makino, and Y. Matsuo, “On the difficulty of improving hand-crafted rules in chat-oriented dialogue systems,” in *APSIPA*. IEEE, 2015, pp. 1014–1018.
- [13] S. Sato, N. Yoshinaga, M. Toyoda, and M. Kitsuregawa, “Modeling situations in neural chat bots,” in *Proceedings of ACL Student Research Workshop*, 2017, pp. 120–127.
- [14] K. Mitsuda, R. Higashinaka, and Y. Matsuo, “What information should a dialogue system understand?: Collection and analysis of perceived information in chat-oriented dialogue,” in *Advanced Social Interaction with Agents*. Springer, 2019, pp. 27–36.
- [15] Y. Machida, D. Kawahara, S. Kurohashi, and M. Sassano, “Design of word association games using dialog systems for acquisition of word association knowledge,” in *Proceedings of AKBC*, 2016, pp. 86–91.
- [16] N. Otani, D. Kawahara, S. Kurohashi, N. Kaji, and M. Sassano, “Large-scale acquisition of commonsense knowledge via a quiz game on a dialogue system,” in *Proceedings of OKBQA*, 2016, pp. 11–20.
- [17] T. Ge, L. Cui, B. Chang, Z. Sui, F. Wei, and M. Zhou, “Eventtwiki: a knowledge base of major events,” in *Proceedings of LREC*, 2018.
- [18] N. Tandon, G. De Melo, A. De, and G. Weikum, “Knowlywood: Mining activity knowledge from hollywood narratives,” in *Proceedings of ACM CIKM*, 2015, pp. 223–232.
- [19] W. Yao and R. Huang, “Temporal event knowledge acquisition via identifying narratives,” in *Proceedings of ACL*, 2018, pp. 537–547.
- [20] R. Sauri and J. Pustejovsky, “Factbank: a corpus annotated with event factuality,” *Language resources and evaluation*, vol. 43, no. 3, p. 227, 2009.
- [21] M.-C. De Marneffe, C. D. Manning, and C. Potts, “Did it happen? the pragmatic complexity of veridicality assessment,” *Computational linguistics*, vol. 38, no. 2, pp. 301–333, 2012.

Research on Chinese Text Error Correction Based on Sequence Model

Jianyong Duan

College of Information Science
North China University of Technology
Beijing, China
e-mail: duanjy@hotmail.com

Abstract—When users input text, it will inevitably produce errors, and with the rapid development and popularization of smart devices, the situation becomes more and more serious. Therefore, text correction has become one of the important research directions in the field of natural language processing. As the grammatical error correction task, in this paper, the error correction process of Chinese text is regarded as the conversion process from wrong sentence to correct sentence. In order to adapt to this task, the (sequence-to-sequence) Seq2Seq model is introduced. The wrong sentence is used as the source sentence, and the correct sentence is used as the target sentence. Supervised training is carried out in units of characters and words. It can be used for correcting errors such as word of homophone, homotype, and near-sound, greatly reducing the artificial participation and expert support of feature extraction, improve model accuracy on specific errors. In order to solve the information loss caused by the conversion of long sequence to fixed length vector, the attention mechanism is introduced into the basic model. After adding the attention mechanism, the model's accuracy, recall rate and F1 value have been effectively improved.

Keywords- correction; Seq2Seq; attention

I. INTRODUCTION

Text proofreading technology is one of the important tasks of natural language processing. As early as the 1960s, some scholars conducted automatic proofreading research on English texts. IBM implemented a TYPO English spell checker by UNIX on IBM/360 and IBM/370 in 1960[1]; in 1971 Stanford University's Ralph Gorin implemented an English spell checker spell[2] on the EDC-10 machine. In recent years, with the continuous development of technology, Text proofreading research is also constantly making progress, and there are some commercialized results, such as Grammarly, Deal proof, Proofread and other special English word spell checking system. In the 1990s, Chinese scholars began to conduct research on Chinese text proofreading, but they developed rapidly. At present, many technology companies and universities or research institutions have invested a certain amount of human and material resources to carry out research in this area.

Among them, automatic error correction is an important part of automatic text proofreading. It provides suggestions for modification of error strings detected during automatic error detection, and assists users in correcting errors. The effectiveness of the proposed modification is the main indicator for measuring the automatic error correction performance. It has two requirements: the first proposed modification proposal should contain correct or reasonable suggestions, and the correct or reasonable recommendations should be arranged in front of all suggestions as much as possible. Therefore, the proposed algorithm and sorting algorithm are the two core topics of automatic error correction research. However, the Chinese text automatic

Yang Yuan, Hao Wang, Xiaopeng Wei, Zheng Tan

College of Information Science
North China University of Technology
Beijing, China
e-mail: yuanyangeli@163.com

proofreading theory is still not mature, and the discussion of automatic error correction is still rare.

At present, there are a number of commercial text proofing software products on the market. The representative products include the black horse proofreading system, the Founder Jinshan proofreading system, the woodpecker proofreading system, etc. They have obtained a certain degree of application in the publishing and printing industry and achieved good result. However, since most of the wrong language models are built based on the statistical features of words, the use of linguistic knowledge is still insufficient, and the error recovery and accuracy rate are still not up to the requirements of use. The efficiency of the error correction recommendation or the preferred correct rate is not high. It is difficult to find out the higher-level (such as sentence-level) errors, and there is still a big gap with the user's requirements. Therefore, the text proofreading technology needs to be further improved. And in the traditional Chinese text error correction field, the commonly used method is based on rules or statistical-based models. The disadvantage of the rule model is that it cannot fit all possible situations with simple rules, and new rules need to be added continuously. Both statistical models and rule models cannot take advantage of the semantic relationship of contexts.

With the successful application of deep learning in the NLP field, seq2seq has achieved good results in some end-to-end applications such as (automatic abstract, machine translation, text categorization, etc.). Different from the traditional method, the seq2seq model can learn the semantic relationship of the context, and can give more reasonable correction suggestions for the specific context.

II. RELATED WORK

The traditional text proofreading is mainly divided into two parts. First, it is necessary to check the error, that is, in a sentence or an article, find the wrong position, and then replace the word or character in the wrong position with the correct word or character. Generally, several optimal options are recommended for the user to select during the error correction process.

At present, the error detection algorithm includes a context-based local language feature method; the transition probability is used to analyze the continuation relationship between adjacent words[3][4]; using rules or linguistic knowledge, such as grammar rules, word collocation, etc.; Statistical method. There are no obvious boundaries between the different methods and they can be mixed.

Context-based local language features such as the multi-feature-based Chinese automatic proofreading method designed and implemented by Microsoft China Research

Institute, which takes into account the local language features of words, words and part of speech in Chinese texts and the long-distance language features, and uses the Winnow method. Feature learning, using these context features to select the words in the target word confusion set, the main difficulty is how to convert the target sentence into multiple effective features and the acquisition of the confusion set[5]. Harbin Institute of Technology will search for possible candidates for each word in the proofreaded sentence, and form the word candidate matrix of the sentence. On this basis, using the structural features and statistical features of the language itself, from the candidate matrix. Select the best word candidate sequence for the sentence, compare it to the original sentence, find the wrong word, and correct it with the first candidate.

In the rule method, Yi Rongxiang, He Kekang, etc. use the revised grammar rules to proofread the manuscript[6]. If the sentence satisfies the rule of correcting grammar, the corresponding words are marked incorrectly according to the rules, but the limited rules are difficult to cover a large number of unpredictable The error phenomenon, limited ability to check. Liu Ting et al. used the clauses as a unit to scan Chinese sentences three times. Through automatic word segmentation, automatic recognition of new words, and the use of phrase rules to separate single words into phrases, and gradually bundle the correct strings[7]. The remaining single string that cannot be bundled is determined to be an error. The downside is that limited phrase binding rules are difficult to cover a large number of linguistic phenomena. Wu Yan et al.[8] also proposed a proofreading method combining word matching and grammar analysis. Using the combination of rules and statistics, the large-scale corpus is not used, the hash is found by the inverse maximum matching and the local corpus statistical algorithm, and the word matching and syntax analysis are performed on the hash, and then the candidate error string is found. The interactive method automatically corrects the error string and achieves a high error detection rate.

In the statistical-based method, Zhang Zhaochuang[9]proposed a method of using comprehensive approximate word set replacement and using statistical language model to score. The shortcoming is that it can only correct the so-called word errors, multi-words, missing words, and Bit errors are hard to find. Yu Yu, Yao Tianshun proposed a hybrid text proofreading method HMCTC, using pattern matching method to find the longest matching participle and finding long word errors. Then according to the ternary grammar, the frequency of co-occurrence with the adjacent words is less than a certain threshold. The word is marked as an error; finally, the word is marked with a grammatical attribute, and the error is marked in the sequence of the impossible grammar labeling sequence[10]. The disadvantage is that the error-checking criterion based on the co-occurrence frequency of words is limited by the size of the training corpus and the field of corpus selection, and the acquisition of the co-occurrence frequency data requires a large-scale segmented corpus, and such corpus is Hard to get. Sun Cai, Luo Zhensheng used the corpus statistical knowledge to guide the text proofreading[11], taking the sentence as the unit, treating the sentence as a field and a segment, calculating the average word frequency of the field and the average transition probability of the field; calculating the word transfer between words

Probability, part-of-speech transition probability, the transition probability is used as the error-checking criterion, and the word or word whose transition probability is less than the threshold is taken as the detected error.

While error correction is another important component of text proofreading, the current mature theory is still rare. Yu Yu, Yao Tianshun, etc. used the pattern matching method to correct the long words[10], but did not make full use of the characteristics of the error string, the algorithm is computationally intensive. The IBM China Research Center[12] proposes an alternative word table combined with the main dictionary, which provides an error correction algorithm for modifying the detected error string by adding words and changing words, but the algorithm's error correction suggestions are limited to the replacement word. Table, without considering context-inspired information, mainly considers error correction for erroneous types, and has weaker error correction capabilities for missed words, multi-words, transpositions, multi-word substitutions, and English word spelling. Zhang Yangsen[13] proposed a candidate set generation algorithm for error correction based on likelihood matching, which greatly improved the error correction ability of missed words, multi-words, transpositions and multi-word substitutions.

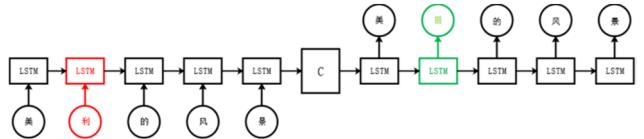


Figure 1 Figure of seq2seq model structure

However, it is worth noting that most of the above mentioned methods require complicated and cumbersome manual processing, as well as expert technical support, which consumes a lot of manpower and material resources. In the field of deep learning, some scholars try to use deep learning to correct grammatical errors[14][15]. To solve this problem, as the grammatical error correction task, this paper proposes a Chinese text error correction model based on sequence-to-sequence (hereinafter referred to as seq2seq), as shown in Fig 1, using the encoder-decoder structure to solve the conversion process of error text to the correct text, the left side is the encoding end, the right side is the decoding end, the encoding end and decoding. The LSTM[17] structure is adopted at the end, and the encoding end generates the semantic vector C of the entire sentence after the loop iteration, and the decoding end decodes the generated vector C into the corresponding text, and completes the conversion of the error text to the correct text. This model is different from the traditional rule-based and statistics-based methods, which effectively reduces manual participation and expert support, and can learn context-related semantic relations. When providing recommendations, it can be targeted for a certain sentence. Error correction scheme. At the same time, in order to alleviate the information loss caused by the seq2seq structure due to the long input sequence, the attention mechanism is introduced. The experiment proves that the seq2seq structure can effectively adapt to the text error correction task, and introduces the attention mechanism to effectively improve the evaluation indicators of the text error correction task.

III. MODEL

The text error correction task can be regarded as the transformation process between different sequences. The error sentence is regarded as the source language and the correct sentence is regarded as the target language. Therefore, the seq2seq model is introduced into the error correction task as a sequence conversion model, and seq2seq is used as The deep learning model can better fit the conversion of the wrong statement to the correct statement compared to the traditional rule model and statistical model. In addition, in order to strengthen the correspondence between the encoding end and the decoding end, attention mechanism is introduced.

The specific process of the Seq2seq+attention model is expressed as follows: Let an input statement be $s = a_1, a_2, a_3, \dots, a_t$, where a_i represents the i th character and m represents the length of the sentence. Each input statement will correspond to a correct sentence of the same correct length, and the content is one-to-one correspondence. In this paper, the error correction task is regarded as a machine translation task, the error sentence is treated as the source sentence, and the correct sentence is treated as the target sentence. Figure 2 shows the basic seq2seq+attention model structure, which rewrites the erroneous sentence “美利的风景 (Murray's landscape) ” into a “美丽的风景 (beautiful landscape) ” and completes the “translation process” from the wrong sentence to the correct sentence, in which the red part indicates the wrong word in the original sentence, and the green part indicates the correct word in the target sentence.

The Embedding layer encodes the characters $a_1, a_2, a_3, \dots, a_t$ into word vectors, inputs them into the LSTM layer of the encoder, and after loop iteration in order, obtains the semantic vector C of the entire sentence, and the hidden state $H = h_0, h_1, h_2, \dots, h_t$. Vector C is used to initialize the initial state of the decoder. The initial state of the input is fixed by "sos" and "0". After combining the two contents, as the initial input, the LSTM layer obtains an output s_1, s_1' at the attention layer. The calculation is performed with H to obtain the attention context. The context is merged with the character vector predicted in the previous step and then used as the input of the next LSTM cell at the decoding end to predict the information of the current node. The specific details of the Embedding layer, the LSTM layer, and the attention are described below.

A. Embedding layer

We set the size of the word vector to a fixed dimension. The size of the word vector in the source sentence is 500 dimensions, and the target sentence is also 500 dimensions. As shown in Figure 2, for each input character a_t , the corresponding word vector is represented as shown in equation(1), where e represents a lookup table of word vectors. That is, for the characters in the sentence, such as “利 (profit)”, we all randomly generate a 500-dimensional vector by uniform distribution. In the process of training, the weight of each reverse derivative is constantly changing. At the end of training, the weight of the model Learning the context of the statement and the general law, the vector of

the word “利 (profit) ” is the optimal expression of the wrong sentence.

$$X_t = e(a_t) \quad (1)$$

B. LSTM layer

The model's encoder and decoder sides use the LSTM[16] structure. The LSTM is the optimized structure

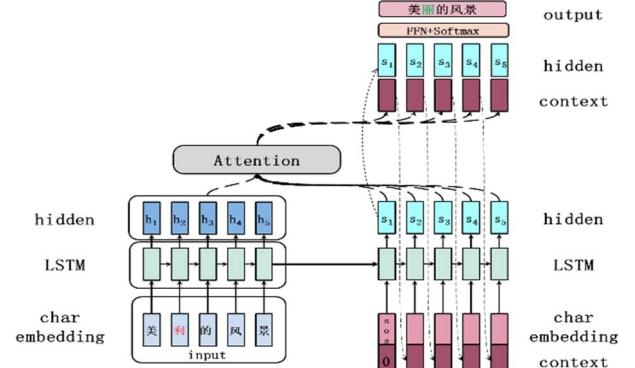


Figure 2 Seq2seq+attention structure

of the RNN. It uses an additional gate mechanism to effectively maintain long distances. The information and the problem of gradient disappearance and gradient explosion of the RNN structure are alleviated. As shown in equation (2)(3)(4):

$$\begin{bmatrix} o_t \\ f_t \\ \tilde{c}_t \\ i_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \tanh \\ \sigma \end{bmatrix} \left(W^T \begin{bmatrix} x_t \\ h_{t-1} \end{bmatrix} + b \right) \quad (2)$$

$$C_t = C_{t-1} * f_t + i_t * \tilde{c}_t \quad (3)$$

$$h_t = o_t * \tanh(C_t) \quad (4)$$

Where f_t, o_t and i_t are forgotten gates, output gates and input gates respectively, W^T and b are parameters of the model, σ represents the sigmoid function, \tanh represents the hyperbolic tangent function, \tilde{c}_t is the input data of the current node, C_{t-1} is the output information of the previous LSTM node. In order to make full use of the information in both directions, this paper uses a bidirectional LSTM structure. Each input sentence $a_1, a_2, a_3, \dots, a_t$, respectively Processing in both positive and negative directions to obtain past and future textual information.

$$\vec{h}_1, \vec{h}_2, \dots, \vec{h}_t = \overrightarrow{\text{LSTM}}(x_1, x_2, \dots, x_t) \quad (5)$$

$$\underline{h}_1, \underline{h}_2, \dots, \underline{h}_t = \overleftarrow{\text{LSTM}}(x_1, x_2, \dots, x_t) \quad (6)$$

Then merge the two hidden layers into the final output.

$$h_i = [\vec{h}_i; \underline{h}_i] \quad (7)$$

C. Attention layer

The During the conversion process from the wrong text to the correct text, we found that in the process of transforming “美利的风景 (Murray's Landscape) ” to the “美丽的风景 (beautiful landscape) ”, the source sentence “美利的风景 (Murray's Landscape) ” finally generates a fixed vector in the loop iteration, and the information is lost seriously. when decoding “利(profit)”, in the fixed vector generated by the encoding segment, the corresponding “利(profit)” information has been depleted, which will lead to

errors in the conversion process of the wrong text and the correct text.

In order to solve the problem of information loss caused by long sequence to fixed length vector transformation, this paper introduces the Attention mechanism, as shown in Figure 3. At the encoding end is a forward-propagating RNN that generates translation results by the equation(8):

$$p(y_t|y_{t-1}, X) = g(y_{t-1}, s_t, m_t) \quad (8)$$

$g()$ is a linear function, s_t and m_t represent the j th time step decoding state and the source text content, respectively, where s_t is calculated as equation (9).

$$s_t = f(s_{t-1}, y_{t-1}, m_t) \quad (9)$$

$f()$ is an activation function, such as the LSTM function. According to the principle of the attention mechanism, this article defines m_t as the weighted sum of the source output h_i :

$$m_t = \sum_{i=1}^I \alpha_{t,i} \cdot h_i \quad (10)$$

$\alpha_{t,i}$ indicates that the degree of matching between s_{t-1} and h_i is as equation (11)(12):

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{i'=1}^I \exp(e_{t,i'})} \quad (11)$$

$$e_{t,i} = v_a^T \tanh(W_a s_{t-1} + U_a h_i + b) \quad (12)$$

W_a U_a and V_a are the weight matrix of the attention matrix. With this model, the decoder automatically selects the vocabulary in the source statement associated with the target word being generated.

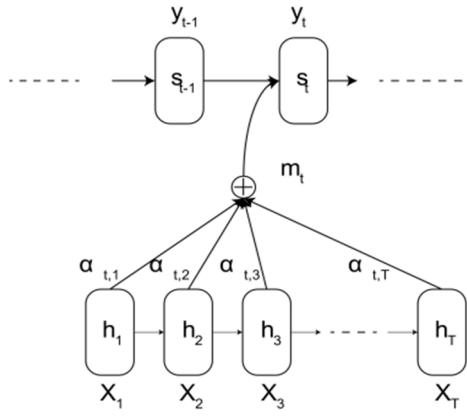


Figure 3 Attention structure

The key operation here is to calculate the weight of the correlation between the encoder and the decoder state, and obtain the Attention distribution, so that the weight of the input position that is more important for the current output position will account for a larger proportion when predicting the output. For example, in the process of transforming “美丽风景 (Murray's Landscape)” to “美丽的风景 (beautiful landscape)”, when the word “丽 (beautiful)” is decoded, the corresponding weight α of the word “利 (profit)” in the source sentence is gradually increased after training, that is, The decoding end adds the information input of the word “利 (profit)”, and corrects the error of the “利 (profit)” word to improve the accuracy of the model.

IV. EXPERIMENT

A. Data

The text errors generated by the input method mainly focus on homophone errors, homonym errors, and near-word errors. This paper mainly uses the above three types of errors as error correction objects. The dataset of this paper is derived from the tencent dataset[17], with 230,000 training sets and 10,000 validation sets and test sets. The training set verification set, the data format is the form of the wrong sentence corresponding to the correct sentence, in the test set, replace the four thousand error data into four thousand correct data, used to calculate the accuracy, recall rate and other evaluation values. In all data sets, each sentence has only one error, the same type, the same tone or the near sound. The data is shown in the following table:

TABLE 1: TRAINING DATA、VALIDATION DATA、TEST DATA

dataset	constitution	total_num	right/wrong	num
Train	src-train	230000	right	0
			wrong	230000
	tgt-train	230000	right	230000
			wrong	0
tencent	src-val	10000	right	0
			wrong	10000
	tgt-val	10000	right	10000
			wrong	0
Test	src-test	10000	right	4000
			wrong	6000

B. Parameter settings

As shown in Table 2, for the model, we set the word vector dimension to 500, the dropout to 0.3, and the parameter uniform distribution to (-0.1, 0.1). This article uses the ADAM optimizer[18] and sets the learning rate. For 0.001, a total of 100,000 rounds of training are set, and the model effect is checked every 10,000 rounds. After 100,000 rounds, the training is ended. The maximum batch of training is 64. In the coding segment, different coding modes RNN, BRNN, and different encode layer (2, 4) is set for different models, and the RNN node is uniformly set to the LSTM structure for whether to use the add attention mechanism to set the parameter to none or generate.

TABLE 2: MODEL SUPER PARAMETER

parameter	value	parameter	value
encoder_type	rnn/bmn	attention	none/generate
optim	adam	rn_type	lstm
learning_rate	0.001	dropout	0.3
enc_layers	2/4	enc_hidden_size	500
dec_hidden_size	500	embedding_size	500

C. Evaluation standard

In order to reasonably evaluate the model, this paper designs the following evaluation criteria, TP: number of correct corrections for incorrect sentences; FP: number of wrong corrections or not corrected for incorrect sentences; TN: no corrections for correct sentences; FN:

changes of correct sentences. The accuracy, recall and F1 are calculated using the equation(13)(14):

$$P = \frac{TP}{TP + FP} \quad R = \frac{TP}{TP + FN} \quad (13)$$

The F1 value is calculated as shown in the following formula:

$$F_1 = \frac{2PR}{P + R} \quad (14)$$

D. Experimental design

The experiment is divided into two parts. The first experiment is the model tuning part. In this part, five sets of comparative experiments are performed, namely seq2seq, seq2seq+attention(2 layers)/(4 layers), and seq2seq+attention+brnn (2 layers)/(4 layers), in which each set of experiments was set up for multiple experiments in order to obtain stable results, and finally averaged. Here 2 layers and 4 layers represent the number of layers of the LSTM layer at the encoding end. In the second part, the data is sent to the model in units of words and words for training, and the effects of different granularity data on the model's effect are compared and analyzed.

E. Experiment results and analysis

As shown in Table 3, the basic seq2seq model has the worst performance. After adding the attention mechanism, the scores are greatly improved. The results show that the attention mechanism is very helpful for the improvement of text correction.

Among all the experimental average results, Seq2seq+attention (4 layers) has the highest accuracy, recall and F1 values, seq2seq+attention+brnn (4 layers) accuracy, and the recall and F1 values are equally high, respectively, 58.3%, 68.2%, 31.4%. However, after comparing Seq2seq+attention(4 layers), seq2seq+attention+brnn (4 layers), it is found that after the BRNN is used, there is no improvement in the error correction effect, indicating that the BRNN structure does not help the error correction task.

TABLE 3:VALUATE SCORE OF DIFFERENT MODEL

	Model	P	R	F1
Enc 2 layers	Seq2seq	23.18	38.46	28.93
	Seq2seq+attention	57.47	73.37	64.30
	Seq2seq+attention+brnn	49.92	57.91	53.62
Enc 4 layers	Seq2seq+attention	58.51	81.46	68.11
	Seq2seq+attention+brnn	57.92	81.84	67.83

In order to make the data more accurate, this paper designed a number of experiments, and finally averaged, the results obtained confirm the original idea, the best model is Seq2seq+attention (4 layers). It can be found that in the error correction task, the BRNN may introduce noise and affect the effect of the model.

Comparing the two models of Seq2seq+attention(2 layers) and Seq2seq+attention(4 layers), we can find the four-layer LSTM structure, which is very helpful for the Chinese text correction task and can help to adapt to the error correction task. And effectively improve the accuracy,

recall rate, F1 value and other evaluation indicators. A large part of the reason for the data increase is that the size of the parameters has increased, and the model can better fit the text error correction task, thereby improving various evaluation indicators.

In order to better evaluate the model effect, we analyzed the results of the test data prediction. The analysis results are as Figure 4.

Figure 4 take the model's generated data for analysis. The model parameters are (adam lr(0.001) rnn attention 2layers), and the test data are all sentences with errors. Among all the data generated by the test data, there are 10049 pieces of data. The length of the sentence of the prediction result is the same as the length of the sentence of the standard answer, but there are also some pieces of data whose predicted answer is different from the sentence length of the standard answer. This type of data accounts for 2%. There are 11 numbers in the same length of data that have errors (that is, the wrong sentence is the same as the

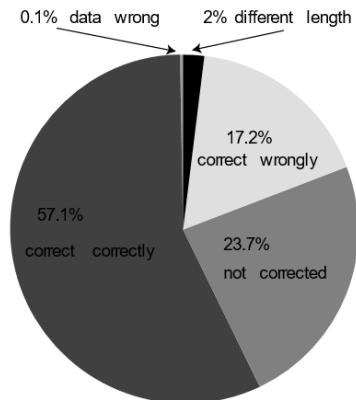


Figure 4 Analysis of prediction results

correct sentence), and this data needs to be filtered out. There are 5,857 correct correctly sentences, accounting for 57.1% of all data. There are 4248 uncorrected and corrected data, accounting for 41.43%, of which 1768 are correct errors, accounting for 17.2%. What can be found is that error correction and uncorrected errors account for a large

TABLE 4:RATING RESULT OF CHAR SIZE AND WORD SIZE MODEL

Model	P	R	F1
Char	58.51	81.46	68.11
Word	28.11	25.42	13.31

proportion of the total data, and the model has a lot of room for improvement.

The second experiment is based on the first experiment. The pre-processed Chinese text data is firstly segmented using the jieba word segmentation tool, and then sent to the model for training. The results are shown in Table 5.

The results show that the results obtained using the word segmentation data are generally inferior to the evaluation indicators of the character-based data. The reason is that when using the word segmentation tool for word segmentation, it is inevitable to introduce word

segmentation errors, and in the process of "translation", training in word units, the lexicon shared by both parties will be reduced, in "translation" Extra-territorial words (OOV) often occur, resulting in a decrease in error correction.

V. CONCLUSION

Based on the basic seq2seq model, the Attention mechanism effectively improves the error correction effect, and the multi-layer LSTM Layer helps improve the accuracy of the model. Training is required to use character-based data for training, which can reduce the impact of word segmentation. It can be seen that the Seq2seq model is well adapted to the Chinese text error correction task, and the accuracy, recall rate and F1 value can reach 58.4%, 68.7%, and 31.6%, respectively, and have achieved good results. It can be seen that the text-based error correction model based on the sequence model has a higher score on the error correction task. In t

he case of similar scores, this model can greatly reduce the workload of manually extracting features, and adopts the end-to-end approach directly. Solve the problem of error correction. However, there are still many improvements in the model, such as the problem of miscorrecting and correcting, how to make targeted changes to the attention mechanism to reduce the occurrence of miscorrecting and correcting errors. Modify the input mode of the model so that the model can use the information of the word and use the character information to help the model improve the effect. The next work is mainly to solve these two problems.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (61672040, 61972003), Beijing Urban Governance Research Center and the North China University of Technology Startup Fund.

REFERENCES

- [1] Kukich K. Techniques for automatically correcting words in text[J]. Acm Computing Surveys (CSUR), 1992, 24(4): 377-439.
- [2] James L. Peterson. 1980. Computer programs for detecting and correcting spelling errors. Commun. ACM 23, 12 (December 1980), 676-687. DOI: <https://doi.org/10.1145/359038.359041>
- [3] Riseman E M, Ehrlich R W. Contextual word recognition using binary digrams[J]. IEEE Transactions on Computers, 1971, 100(4): 397-403.
- [4] Golding A R, Schabes Y. Combining trigram-based and feature-based methods for context-sensitive spelling correction[C]//Proceedings of the 34th annual meeting on Association for Computational Linguistics. Association for Computational Linguistics, 1996: 71-78.
- [5] Lau R, Rosenfeld R, Roukos S. Trigger-based language models: A maximum entropy approach[C]//1993 IEEE International Conference on Acoustics, Speech, and Signal Processing. IEEE, 1993, 2: 45-48.
- [6] Yi Rongxiang, He Kekang. Computer Chinese proofreading system [D]. , 1997.
- [7] Liu Ting, et al. Principles of Chinese Computer Aided Proofreading System [J]. Chinese Information, 1997.
- [8] Wu Yan, Li Xiukun, Liu Ting, et al. Research and implementation of Chinese automatic proofreading system [J]. Journal of Harbin Institute of Technology, 2001, (2): 60-64.
- [9] Chang, C.H.. (1994). A pilot study on automatic Chinese spelling error correction. Communication of COLIPS. 4. 143-149.
- [10] Yu Wei, Yao Tianshun. A mixed Chinese text proofreading method [J]. Chinese Informatics, 1998, 12(2).
- [11] Sun Cai, Luo Zhensheng. Research on the Correction of Chinese Text Proofreading Words[A]. Proceedings of the Fourth Conference on Computational Linguistics (Language Engineering) [C]. Beijing: Tsinghua University Press, 1997. 319 -324
- [12] Guo Zhili, et al. Algorithm for modifying suggestions in Chinese proofreading system [A]. Proceedings of the 4th Conference on Computational Linguistics (Language Engineering) [C] . Beijing: Tsinghua University Press, 1997. 325-330.
- [13] Zhang Yangsen. The construction of error correction knowledge base in Chinese proofreading system and the generation of error correction suggestions [J]. Chinese Journal of Information, 2001, 15 (5): 33-39.
- [14] Grammatical error correction using neural machine translation. Zheng Yuan and Ted Briscoe. NAACL 2016.
- [15] A Multilayer Convolutional Encoder-Decoder Neural Network for Grammatical Error Correction. Shamil Chollampatt, Hwee Tou Ng. AAAI 2018.
- [16] Mnih V, Heess N, Graves A. Recurrent models of visual attention[C]//Advances in neural information processing systems. 2014: 2204-2212.
- [17] Wang D, Song Y, Li J, et al. A Hybrid Approach to Automatic Corpus Generation for Chinese Spelling Check[C]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018: 2517-2527.
- [18] Kingma D, Ba J. Adam: a method for stochastic optimization (2014)[J]. arXiv preprint arXiv:1412.6980, 2015, 15.

Integrating Knowledge in End-to-End Automatic Speech Recognition for Mandarin-English Code-Switching

Chia-Yu Li

*Institute for Natural Language Processing (IMS)
University of Stuttgart
Stuttgart, Germany
licu@ims.uni-stuttgart.de*

Ngoc Thang Vu

*Institute for Natural Language Processing (IMS)
University of Stuttgart
Stuttgart, Germany
nthangvu@ims.uni-stuttgart.de*

Abstract—Code-Switching (CS) is a common linguistic phenomenon in multilingual communities that consists of switching between languages while speaking. This paper presents our investigations on end-to-end speech recognition for Mandarin-English CS speech. We analyze different CS specific issues such as the properties mismatches between languages in a CS language pair, the unpredictable nature of switching points, and the data scarcity problem. We exploit and improve the state-of-the-art end-to-end system by merging nonlinguistic symbols, by integrating language identification using hierarchical softmax, by modeling subword units, by artificially lowering the speaking rate, and by augmenting data using speed perturbed technique and several monolingual datasets to improve the final performance not only on CS speech but also on monolingual benchmarks in order to making the system more applicable on real life settings. Finally, we explore the effect of different language model integration methods on the performance of the proposed model. Our experimental results reveal that all the proposed techniques improve the recognition performance. The best combined system improves the baseline system by up to 35% relatively in terms of mixed error rate and delivers acceptable performance on monolingual benchmarks.

Keywords-end-to-end speech recognition, Mandarin-English Code-Switching speech, language model integration

I. INTRODUCTION

Code-switching (CS) speech is a common phenomenon in multilingual countries and defined as speech which contains more than one language [1]. From a grammatical point of view, Poplack [2] proposed three types of CS: extra-sentential, inter-sentential and intra-sentential. Extra-sentential switching is inserting tag elements from one language into an otherwise monolingual language. Inter-sentential switching is characterized by a switch from one language to another outside the sentence or the clause level, whereas intra-sentential switching is switching from one language variety to another at the clause, phrase, or word level within a single utterance. This paper aims at improving end-to-end (E2E) automatic speech recognition (ASR) system on the SEAME corpus (South East Asia Mandarin-English) [3] which is intra-sentential dominant [4].

A first ASR system for Mandarin-English CS conversational speech was proposed in [5] investigating different merged acoustic units for acoustic modeling, artificial CS data for language modeling, and the use of language identification in the decoding process. Recent studies show

that deep learning has boosted the performance of ASR [6], [7] and the state-of-the-art ASR architecture - hybrid TDNN-HMM - has shown incredible performance on many LVCSR tasks [8]. Despite hybrid ASR having state-of-the-art performance, building this system remains a complicated and expertise-intensive task. First, it requires various resources such as pronunciation dictionaries and phonetic questions for acoustic modeling. Second, it relies on GMMs for frame-level alignments. In the context of CS, creation of a pronunciation dictionary for two languages might require expertise knowledge, e.g., generating pronunciation variants, and this process is error prone. Recently, some studies proposed a single neural network architecture to perform speech recognition in an end-to-end manner to resolve the issues in hybrid ASR. There are two types of E2E frameworks: CTC based [9], [10] and attention based [11], [12]. Although the attention model has been shown to improve the performance over CTC based, it is difficult to learn in the initial training stage with long input sequences and has poor performance in noisy conditions. Joint CTC-attention based E2E framework was proposed to improve noise robustness, achieving fast convergence and mitigating the alignment issue [13]. The experimental results on many benchmarks (WSJ, CHiME-4, etc.) demonstrate its advantages over both the CTC and attention based frameworks and comparable results to hybrid ASR systems.

Mandarin and English have many significant differences [14], [15]. First, Mandarin uses a logographic system, in which symbols represent the words' meaning and not their pronunciation. Second, Mandarin is a tone language that uses the pitch to distinguish word meaning, whereas English uses the pitch to express emotion or emphasize words. Third, the syntactic structure. e.g. in English, things are usually modified by the words that come after them, while in Mandarin, things are usually modified by the words that precede them. Furthermore, it is difficult to predict the CS points which is entirely up to the individual speakers [2]. Take the below sentences S1 and S2 as examples, the English word 'GO' has similar pronunciation as the Mandarin character '够', but both have totally different meanings. Both sentences can possibly occur in CS environment. If the ASR system is mainly trained with monolingual Mandarin data, then it is more likely to predict the next character to be '够' given the

history '我'. Some might argue that the ASR systems can learn the conditional probability $P(GO|我)$ from the CS data. However, collecting CS data is time-consuming and financially expensive. Besides, the CS points highly depend on the speakers [2], and it is not easy to cover all possible CS points.

- S1: 我 GO 了(Translation: I go.)
- S2: 我 够 了(Translation: It is enough for me.)

In this paper, we analyze several issues of Mandarin-English CS speech which might cause recognition errors and inject knowledge derived from the analysis into the development process of the E2E ASR system. We merge discourse particles and nonlinguistic signals, integrate language identification into the prediction process, utilize English subword modeling, artificially lower speaking rate, and use data augmentation to solve these issues. Furthermore, we investigate the effect of these techniques and their combinations on the ASR performance in terms of Mixed Error Rate (MER). Finally, we explore different language model integration methods in order to interpolate the knowledge of the language model into our best combined systems.

II. SEAME DATASET

SEAME is a 99 hours of spontaneous Mandarin-English CS speech corpus recorded from Singaporean and Malaysian speakers. All recordings are performed by close-talk microphone in quiet room. The speakers are aged between 19 and 33, almost balanced in gender (49.7% of female and 50.3 % of male). The total number of distinct speakers is 157 (36.8% are Malaysian while the rest are Singaporean) [16]. 16.96% of utterances are English (ENG), 15.54% are Mandarin (MAN) and the rest (67%) are CS utterances. In each transcript, they use the following categories for labeling: target language (English word and Mandarin character), discourse particle or hesitation ('lah', 'hmm', etc.) and nonlinguistic signal (people laughing, coughing, etc.), other languages (Japanese or Korean words).

III. IDENTIFICATION OF SUBPROBLEMS

A. Discourse particle and nonlinguistic signal

There are 430 unique discourse particles, hesitations and nonlinguistic signals (people laughing, coughing). These signals might be informative for sentiment analysis or emotion detection not for speech recognition.

B. Code-Switching points prediction

There are two language switching directions: one is from English to Mandarin and another is from Mandarin to English. SEAME has 12.24% switching points (6.04% are from English to Mandarin and 6.2% are from Mandarin to English). Previous studies state that the code-switching points are indeterminate because the code-switching decision is entirely up to the individual speakers [17] and there are some code-switching patterns across speakers [2]. Moreover, in over 80% of cases, speakers directly switch language without any short pause and discourse

particle between two adjacent different languages [3]. It is a challenge for conventional ASR system to predict the switching points due to insufficient acoustic information.

C. Out-of-Vocabulary (OOV)

OOV is a common problem in the context of speech recognition and would be accumulated due to the recognition of two languages. For example, there are around 370,000 Mandarin Chinese words and 172,000 English words. If we just combine two dictionaries to a lexicon for the ASR systems, the tedious lexicon would make ASR hard to be trained due to huge memory and time consumption. Not to mention it does not contain the new words being created in daily life or social media.

D. High speaking rate

A rate of clear speech ranges between 140-160 words per minute (wpm) and a rate higher than 160 wpm can make it difficult for the listener to absorb the material. Reference [3] reports that Singaporean speakers have an average speaking rate of 181 wpm and Malaysian speakers 151 wpm. Note that there are 72 hours of speech from Singapore, and 27 hours from Malaysia. Therefore, around 70% of the utterances have high speaking rates.

E. Data scarcity

Although there are many multilingual countries, only few countries do CS between Mandarin and English. Besides, CS speech normally occurs in casual conversation and it is not possible to record it for free due to the privacy concern. Especially in the context of E2E ASR, data scarcity might be a large problem to build a good system.

IV. PROPOSED METHODS

A. End-to-End speech recognition

It has been shown that the joint CTC-attention model within the multi-task learning framework [13] is able to outperform CTC-based or attention-based E2E ASR systems due to its robustness, fast convergence, and mitigation of the alignment issues. Furthermore, it allows building ASR systems without the use of a pronunciation dictionary, which is convenient for CS ASR because combining two languages' pronunciation dictionaries requires expertise knowledge. The overall architecture contains the shared encoder which is trained by both CTC and attention model objectives simultaneously and transforms the input sequence \mathbf{x} into high level features \mathbf{h} , and the location-based attention decoder generates the character sequence \mathbf{y} [18]. The multi-task learning (MTL) objective, is represented in Eq.1, follows by using both CTC and attention model.

$$\mathcal{L}_{MTL} = \lambda \mathcal{L}_{CTC} + (1 - \lambda) \mathcal{L}_{Attention} \quad (1)$$

where \mathcal{L}_{CTC} is the loss function of the CTC model, $\mathcal{L}_{Attention}$ is the loss function of the attention model, and a tunable parameter $0 \leq \lambda \leq 1$.

B. Merging discourse particles and nonlinguistic signals

In III-A, we mention the huge amount of labels for discourse particles, hesitations, and nonlinguistic signals. The system should put effort on learning language instead of nonlinguistic symbols. Therefore, we group all the discourse particles and hesitation pauses into the same class, e.g., "lah" and "hmm" are labeled as "<dispar>", and all the nonlinguistic signals are labeled as "<nlsyms>". The goal is to let neural network focus on learning language (English and Mandarin) characters because they will be the main factors to the loss function.

C. Language identification using hierarchical softmax

To predict the CS points mentioned in III-B, we exploit language identification to predict if the current word (character) is English or Mandarin given the history in terms of the high level features h (or the output of Encoder). The language identification is integrated into E2E attention model with the output layer factorized by class layer (hierarchical softmax), proposed in [19]. The probability of character at the i -th time step $y(i)$ given *history* is defined as

$$P(y(i)|\text{history}) = P(s(i)|\text{history})P(y(i)|s(i)) \quad (2)$$

where $s(i)$ denotes the type of language (English or Mandarin) at the i -th time step.

Furthermore, the overall E2E system is trained using multi-task learning objective represented in Eq.3 using CTC, attention, and language identification models.

$$\mathcal{L}_{MTL} = \lambda \mathcal{L}_{CTC} + (1 - \lambda)(\mathcal{L}_{Attention} + \mathcal{L}_{ld}) \quad (3)$$

D. English subword modeling

III-C identifies the OOV problems for both languages. For Mandarin, we could use characters as unit to mitigate the OOV problem because our E2E system only needs to recognize 50,000 characters instead of 370,000 words. For English, the subword model can solve the OOV problem and offer a capability in modeling longer context than using characters [20], [21].

E. Lower speaking rate

In III-D, we mentioned the problem of high speaking rate causing the difficulty for recognition. In a daily conversation, when people do not understand what others say, normally they ask others to say it again, and they will repeat it with a lower speaking rate. Motivated by this observation, we propose to artificially lower the speed of the entire dataset.

F. Data augmentation

We use '3 way speed-perturbed' method proposed in [22] to generate more CS data. In particular, two additional copies of the original training data are generated by modifying the speed to 0.9 and 1.1 of the original rate and added to the original data. Furthermore, we add several monolingual datasets for the training process allowing our system to learn more pronunciation variants.

G. Language model integration methods

There are several ways to integrate E2E ASR systems with an external language model. In a conventional decoding paradigm with an external language model, shallow fusion (SF) computes the score by linearly interpolating the score from a Sequence-to-Sequence (S2S) model and an external language model to maximize the following criterion:

$$y^* =_{y \in \Omega^*} \{\ln P_{S2S}(y|x) + \beta P_{LM}(y)\} \quad (4)$$

where x is acoustic features and y is the sequence made of English words and Mandarin characters. Where β is a tuneable parameter to define the importance of the external LM. Unlike SF uses the language model in the decoding state, cold fusion (flat-start fusion) (CF) uses the pre-trained language during the training of the S2S model to provide effective linguistic context [23]. The fine-grained element-wise gating function is equipped to flexibly rely on the language model depending on the uncertainty of predictions:

$$s_t^{LM} = DNN(d_t^{LM}) \quad (5)$$

where d_t^{LM} is the hidden states of RNNLM, s_t^{LM} is a feature from the external LM. The S2S models' hidden states s_t^{ED} is defined as:

$$s_t^{ED} = \sigma(W_{ED}[d_t; c_t] + b^{ED}) \quad (6)$$

CF uses a fine gating mechanism, and the gating function g_t takes features from the S2S model and the external LM.

$$g_t = \sigma(W^g[s_t^{ED}; s_t^{LM}] + b_g) \quad (7)$$

$$s_t^{CF} = [s_t^{ED}; g_t \circ s_t^{LM}] \quad (8)$$

where \circ is element-wise multiplication.

$$r_t^{CF} = DNN(s_t^{CF}) \quad (9)$$

$$P(y_t|h, y < t) = softmax(W_{CF}r_t^{CF} + b^{CF}) \quad (10)$$

V. EXPERIMENTAL SETUP

The corpus is split into train, development, and evaluation sets. The statistics of the three sets are shown in Table I. CS, MAN, and ENG represent Code-switching, Mandarin, and English utterances, respectively. Based on the ratio of CS, MAN, and ENG utterances in the three sets, the evaluation set is Mandarin dominant and the development set is relatively bilingual balanced.

Table I
THE STATISTICS OF THE TRAIN, DEV AND EVAL SETS

Sets	# spk	# utt	#hrs	Ratio(%)		
				CS	MAN	ENG
train	141	93782	96	67.19	15.45	17.36
dev	8	6549	1.8	66.63	17.03	16.34
eval	8	5037	1.4	74.08	15.53	10.38

A. Baseline system

The baseline system is trained by the best recipe from Espnet [18]. The encoder network is represented by bidirectional long short-term memory (BLSTM) with subsampling and has 5 layers with 1024 units. The decoder is represented by 1 layer of BLSTM with 1024 units. The hybrid CTC/attention parameter (λ) (Eq.1) is 0.5. The beam size is 20 and the CTC weight is 0.5 for decoding. The dictionary is character based.

B. Additional monolingual data

For Mandarin Chinese dataset, we utilize Aishell-1 which contains 170 hours of speech contributed by 400 people from different accent areas in China [24], THCHS30 containing 30 hours of Mandarin Chinese speech database [25], and Free ST Chinese Mandarin Corpus (ST-CMDS) having 110 hours (855 speakers) of speech recorded in a silent indoor environment using a cellphone [26]. The English datasets in the experiment are 1000 hours of Librispeech [27] and 425 hours of Common Voice [28], and 5 hours of Ted talks extracted from the TEDxSingapore website[29].

VI. RESULTS & ANALYSIS

This section presents a performance comparison in terms of MER (%) between the baseline (joint CTC/attention E2E) and all proposed solutions that are denoted as E2ELD (baseline with language identification), E2ESW (baseline with subword modeling), SL (slowing down the speed of utterance), 3W (3-way speed perturbation), and F (adding monolingual data). Label 1 denotes the dataset using the original labels and label 2 denotes the dataset using our proposed labels (merging discourse particles and nonlinguistic signals). Moreover, we examine the systems on the test sets with and without nonlinguistic symbols (discourse particles and nonlinguistic signals). The test set without nonlinguistic symbols will show how well the system recognizes actual language.

A. Merging discourse particles and nonlinguistic signals, and language identification

Table II shows that the baseline system trained with label 2 does not outperform the one trained with label 1. However when integrating language identification information in the output layer using hierarchical softmax, the system trained with label 2 data improves the performance especially on the test sets without nonlinguistic symbols (No nlssyms). The hypothesis of baseline E2E and E2ELD for one utterance in eval set are shown in Table III. The output of baseline has mistakenly recognize English word 'initiative' as the sequence of English characters and Mandarin characters in blue ink while E2ELD model has better identification between languages.

B. English subword modeling

We use two different texts (SEAME and Librispeech) to train the English subword model and add different amounts (100~5000) of subwords to the dictionary. The result shows that E2ESW with 500 subwords trained from

Table II
THE MER(%) ON SEAME TEST SET OF BASELINE AND ALL PROPOSED MODELS

Systems	Label	No nlssyms			
		type	dev	eval	dev
E2E	1	39.8	31.7	39.8	31.0
E2ELD	1	39.1	31.4	38.7	30.6
E2ESW(SEAME500)	1	34.6	27.9	34.1	27.1
E2E+SL(0.7)	1	36.5	29.2	36.1	28.4
E2E+3W	1	34.8	27.7	34.4	26.8
E2E	2	40.3	31.6	41.1	31.4
E2ELD	2	38.1	29.3	38.7	28.9
E2ESW(SEAME500)	2	33.6	26.4	33.9	25.7
E2E+SL(0.8)	2	37.3	29.0	38.0	28.6
E2E+3W	2	34.4	26.7	34.3	26.2

Table III
THE HYPOTHESIS OF BASELINE AND E2ELD AND E2E-SL MODELS FOR ONE UTTERANCE IN EVAL SET

Systems	Hypothesis
Ground-truth	then 你 不 可 以 take initiative 去 讲 么
E2E	then 你 不 可 以 that in 你 学 tive 就 讲 嘛
E2ELD	then 你 不 可 以 tat initiative 就 讲
Ground-truth	why you want to be the head of your of your group of friends
E2E	why want to be the head of your group of friends
E2E-SL (0.7)	why you want to be the head of your group of friends

SEAME text has better performance than the one with 500 subwords trained from Librispeech. The reason is that the frequent subwords in Librispeech and SEAME are not similar. To be more specific, SEAME has many conversation style English words and proper names related to South Asia while Librispeech mainly contains literary words. Therefore, the words in Librispeech are not likely to be used in a casual conversation in South Asia.

C. Lower speaking rate

We examine different factors (0.6~0.9) to lower the audio speed. Table II shows that lowering the speaking rate to 0.7 has the best performance for label 1 while the speaking rate of 0.8 works the best for label 2. Table III shows one example which has high speaking rate (14 words in 3 seconds), it reveals that baseline E2E model fail to recognize some words when the speaking rate is high.

D. Data augmentation: 3 way speed-perturbed (3W) and monolingual data (F)

Table II shows that 3 way speed-perturbed (E) improves the performance significantly on both label 1 and label 2 data. Again, E2E+3W performs better with label 2 than label 1. In order to observe the effect of adding different amount of monolingual data to the ASR performance, we create four different mixed datasets Table IV shows the performance on SEAME test sets (without non language symbols) of systems trained with each of the four datasets, separately. It also presents the performance on ENG, MAN, and CS utterances. The system trained with F1 (adding 100 hours of Mandarin data) mostly improves the performance on the MAN speech. The one trained with F2 (adding 100 hours of English data) improves the performance on the ENG speech and interestingly also MAN and CS speech. When trained

Table IV

THE MER(%) ON SEAME TEST SET OF BASELINE AND E2E+F APPROACH WITH LABEL 1

	No nlsyms	dev			eval			
		dev	eval	ENG	MAN	CS	ENG	MAN
E2E	39.8	31.0	58.5	31.0	38.8	45.9	28.6	30.2
E2E+F1	39.5	30.1	61.1	29.5	38.4	46.0	27.3	29.2
E2E+F2	37.7	29.9	56.0	30.1	36.6	43.3	28.3	29.0
E2E+F3	37.5	28.8	56.5	28.6	36.5	43.2	26.5	28.0
E2E+F4	39.2	30.7	62.8	31.6	38.2	46.9	29.4	29.5

with F4 (adding 1000 hours of Mandarin and English data), they get worse MER than F3 (adding 200 hours of Mandarin and English data) because monolingual data becomes dominant in the train set. However, F4 has better performance on monolingual benchmarks (the system trained with F4 has 12.6% WER on WSJ test and 10.1% CER on AISHELL-1 test set, whereas the system trained with F3 has 26% WER and 35% CER on them). Note that the baseline system which trained only with SEAME data has over 100% WER and CER on both monolingual benchmarks. The results indicate that optimizing the performance on both CS and monolingual test sets is an important trade-off which needs future investigations.

E. To combine all the approaches?

Note that all the combined systems are trained with label 1 data because we want to firstly find the best combined system, then apply it with label 2 data. Table V shows that all the combinations except E2ELD+SW improve the MER and especially the combinations involving E2ESW, SL and 3W improve the most. The reasons why E2ELD+SW performs worst could lie in the fact that by introducing English subword containing 3 to 4 phones, it is much harder for the system to estimate the language identification. Overall, adding monolingual data improves the performance of all the combinations. E2ESW+3W+F3 achieves the best performance with 25.0% MER on the SEAME evaluation set. As mentioned before, we apply this best combination with label 2 data and achieve 23.7% MER on the SEAME evaluation set.

F. Language model fusion methods

Results in Table VI show the comparison between the baseline models and the best improved models without an external language model. For label 1 and label 2, the

Table V

THE MER(%) ON SEAME TEST SET OF BASELINE AND POSSIBLE COMBINATIONS WITH LABEL 1.

Systems	dev	eval	No nlsyms	dev	eval
E2ELD+SW	37.0	29.9	36.7	29.1	
E2ELD+SL	36.2	29.7	35.8	28.9	
E2ELD+3W	34.3	27.4	33.9	26.5	
E2ESW+SL	34.4	27.8	34.0	27.0	
E2ESW+3W	32.5	25.9	32.0	25.1	
E2E+SL+3W	32.8	26.1	32.3	25.2	
E2ESW+SL+3W	31.9	26.0	31.5	25.1	
E2ELD+3W+F3	34.0	26.8	33.5	25.9	
E2ESW+SL+F3	32.8	26.4	32.3	25.5	
E2ESW+3W+F3	31.4	25.0	30.8	24.2	

Table VI

THE MER(%) ON SEAME TEST SET OF BASELINE, IMPROVED COMBINED MODEL W/O OR W/ LM AND CONVENTIONAL TDNN-HMM HYBRID SYSTEM

Systems	LM fusion	label type	No nlsyms	
			dev	eval
E2E		1	39.8	31.7
		1	31.4	25.0
E2ESW+3W+F3		2	40.3	31.6
		2	30.8	23.7
TDNN-HMM	SF	2	35.9	30.7
	SF	2	29.8	22.8
	CF	2	29.9	23.0
E2ESW+3W+F3		2	29.7	22.2

improved models achieve up to 35% relative performance to the baseline models. Again, the model trained with label 2 data has the lowest MER. The second row in the Table shows the comparisons between the state-of-the-art TDNN-HMM [8] which applies i-vector and 3-way data perturbation techniques followed by a Kaldi chain recipe [30]. Kaldi exploits a bilingual pronunciation dictionary, which does not contain the pronunciation of discourse particles, hesitations' nor nonlinguistic signals, to train the TDNN-HMM chain model and integrates the language model using SF. The best improved model with SF and CF outperforms TDNN-HMM chain model with SF.

Table VII shows how our external language models can improve the best E2E model. As mentioned before, E2ESW+3W+F3 is not trained with the language identification loss function since subword modeling will harm the performance of the language identification. Therefore, it sometimes misrecognizes the English signal as a Mandarin signal. For example, it recognizes 'take' as '帶'. The external language model can help to increase the score of 'take initiative' in order to output the correct sentence. In this case, the models using shallow fusion or cold fusion to inject language model knowledge (E2ESW+3W+F3+SF and E2ESW+3W+F3+CF) do not make a mistake of recognizing languages.

Note that 'so' and '所以' have similar pronunciation and meaning. The interesting example in Table VII leads to the question of whether Mixed Error Rate is always reliable metrics in the context of CS speech recognition. From the perspective of automatic evaluation, E2ESW+3W+F3+CF performs worse than E2ESW+3W+F3+SF in this case since E2ESW+3W+F3+CF has longer Levenshtein distance (one substitution plus one insertion) to the ground truth sentence than E2ESW+3W+F3+SF's distance (one substitution). However from the perspective of human evaluation, E2ESW+3W+F3+CF performs better than E2ESW+3W+F3+SF with regard to the completion and meaning of entire sentence.

VII. CONCLUSIONS

We analyze several subproblems of Mandarin-English CS speech based on SEAME dataset, and provide solutions to each subproblem within the E2E ASR framework. We explore different combinations of the proposed solutions in order to reach the optimal ASR performance. The experimental results reveal that each solution improves the MER little by little, and the appropriate combination

Table VII
THE HYPOTHESIS OF THE BEST IMPROVED MODEL W/ OR W/O LM FOR
UTTERANCES IN EVAL SET

Systems	Hypothesis
Ground-truth	then 你 不 可 以 take initiative 去 讲 么
E2ESW+3W+F3	then 你 不 可 以 带 initiative 就 讲
E2ESW+3W+F3+SF	then 你 不 可 以 take initiative 去 讲 么
E2ESW+3W+F3+CF	then 你 不 可 以 take initiative 去 讲
Ground-truth	所 以 我 就 去 apply job
E2ESW+3W+F3	所 以 我 就 去 apply job
E2ESW+3W+F3+SF	所 以 我 就 去 ply job
E2ESW+3W+F3+CF	so 我 就 去 apply job

achieves great improvement (up to 35% relatively). Our best combined system with an external language outperforms the baseline and the state-of-the-art hybrid system (TDNN-HMM).

REFERENCES

- [1] P. Auer, “Code-switching in conversation: language, interaction and identity,” *The Modern Language Review*, vol. 95, 2000.
- [2] S. Poplack, “Sometimes I’ll start a sentence in Spanish y termino en español: toward a typology of code-switching 1,” *Linguistics*, vol. 18, no. 7–8, 1980, pp. 581–618.
- [3] D.-C. Lyu, T. P. Tan, C. E. Siong, and H. Li, “An analysis of a Mandarin-English code-switching speech corpus: SEAME,” in Proc. of INTERSPEECH, 2010.
- [4] D.-C. Lyu, T. P. Tan, C. E. Siong, and H. Li, “SEAME: a Mandarin-English code-switching speech corpus in south-east Asia,” in Proc. of INTERSPEECH, 2010.
- [5] N. T. Vu et al., “A first speech recognition system for Mandarin-English code-switch conversational speech,” in Proc. of ICASSP, 2012.
- [6] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,” *IEEE/ACMTrans.Audio, Speech, Language Process*, vol. 20, 2012, pp. 30–42.
- [7] G. Hinton et al., “Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups,” *IEEE Signal Processing Magazine*, vol. 29, 2012, pp. 82–97.
- [8] V. Peddinti, D. Povey, and S. Khudanpur, “A time delay neural network architecture for efficient modeling of long temporal contexts,” in Proc. of INTERSPEECH, 2015.
- [9] A. Graves and N. Jaitly, “Towards end-to-end speech recognition with recurrent neural networks,” in Proc. of ICML, 2014.
- [10] A. L. Maas, Z. Xie, D. Jurafsky, and A. Y. Ng, “Lexicon-free conversational speech recognition with neural networks,” in Proc. of NAACL, 2015.
- [11] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, “End-to-end attention-based large vocabulary speech recognition,” in Proc. of ICASSP, 2016.
- [12] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, “Opennmt: Open-source toolkit for neural machine translation,” in EAMT User Studies and Project/Product Descriptions, 2017.
- [13] T. Hori, S. Watanabe, and J. Hershey, “Joint CTC/attention decoding for end-to-end speech recognition,” in Proc. of ACL, 2017.
- [14] F. Zhang and P. Yin, “A study of pronunciation problems of English learners in China,” in *Asian Social Science*, vol. 5(6), 2009.
- [15] P. Roach, “English phonetics and phonology: a practical course,” in Cambridge University Press, 2000.
- [16] Mandarin-English code-switching in south-east Asia. [Online]. Available: <https://catalog.ldc.upenn.edu/LDC2015S04>. [Accessed: May- 2019].
- [17] P. Auer, “From code switching via language mixing to fused lects toward a dynamic typology of bilingual speech,” in *International Journal of Bilingualism*, vol. 3, 1999, pp. 309–332.
- [18] S. Watanabe et al., “ESPnet: end-to-end speech processing toolkit,” in Proc. of INTERSPEECH, 2018.
- [19] T. Mikolov, S. Kombrink, L. Burget, J. H. Černocký, and S. Khudanpur, “Extensions of recurrent neural network language model,” in Proc. of ICASSP, 2011.
- [20] Z. Xiao, Z. Ou, W. Chu, and H. Lin, “Hybrid CTC-attention based end-to-end speech recognition using subword units,” in Proc. of ISCSL, 2018.
- [21] Z. Zeng et al., “On the End-to-End Solution to Mandarin-English Code-switching Speech Recognition.” in Proc. of INTERSPEECH, 2019.
- [22] T. Ko1, V. Peddinti, D. Povey, and S. Khudanpur, “Audio augmentation for speech recognition,” in Proc. of INTERSPEECH, 2015.
- [23] S. Toshniwal, A. Kannan, C.-C. Chiu, Y. Wu, T. N Sainath, K. Livescu, “A comparison of techniques for language model integration in encoder-decoder speech recognition,” in Proc. of SLT, 2018.
- [24] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, “AIShell-1: an open-source Mandarin speech corpus and a speech recognition baseline,” in Proc. of Oriental COCOSDA, 2017.
- [25] THCHS-30. [Online]. Available: <http://arxiv.org/abs/1512.01882>. [Accessed: May- 2019].
- [26] Free ST Chinese Mandarin corpus. [Online]. Available: <http://www.openslr.org/38/> [Accessed: May- 2019].
- [27] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: an ASR corpus based on public domain audio books,” in Proc. of ICASSP, 2015.
- [28] Mozilla common voice. [Online]. Available: <https://voice.mozilla.org/en>. [Accessed: May- 2019].
- [29] TedXSingapore. [Online]. Available: <https://www.ted.com/tedx/events/25530>. [Accessed: May- 2019].
- [30] D. Povey et al., “The Kaldi speech recognition toolkit,” in Proc. of ASRU, 2011.

Improving Question Classification with Hybrid Networks

Yichao Cao^{1,2}, Miao Li¹, Tao Feng^{1,2}, Rujing Wang¹, Yue Wu³

1. Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei, 230031, China
2. University of Science and Technology of China, Hefei, 230026, China
3. Institute of Physical Science and Information Technology, Anhui University, Hefei 230601, China
{cycaco,ft2016,wuyue16}@mail.ustc.edu.cn, {mli,rjwang}@iim.ac.cn

Abstract—Question classification is a basic work in natural language processing, which has an important influence on question answering. Due to question sentences are complicated in many specific domains contain a large number of exclusive vocabulary, question classification becomes more difficult in these fields. To address the specific challenge, in this paper, we propose a novel hierarchical hybrid deep network for question classification. Specifically, we first take advantages of word2vec and a synonym dictionary to learn the distributed representations of words. Then, we exploit bi-directional long short-term memory networks to obtain the latent semantic representations of question sentences. Finally, we utilize convolutional neural networks to extract question sentence features and obtain the classification results by a fully-connected network. Besides, at the beginning of the model, we leverage the self-attention layer to capture more useful features between words, such as potential relationships, etc. Experimental results show that our model outperforms common classifiers such as SVM and CNN. Our approach achieves up to 9.37% average accuracy improvements over baseline method across our agricultural dataset.

Keywords-question answering; question classification; hybrid networks;

I. INTRODUCTION

QA is a challenging task in Natural Language Processing (NLP), which has drawn significant attention from the past few decades. As an important sub-module of QA, question classification can effectively narrow the space of candidate answers and impacts the quality of QA. To a certain extent, question classification can be regarded as a special case of text classification without sufficient lexical context [1].

However, question sentences with massive amounts of exclusive vocabulary are usually complicated in some particular domains. It is relatively difficult to extract features from question sentences using general classification algorithms, that results in poor classification performance. To address the above specific challenge, we build a novel hierarchical hybrid deep network to improve the accuracy of question classification.

In particular, we first apply denoising and THULAC's Chinese word segmentation [2] to question sentences to improve the quality of data. Then we use word2vec [3] combined with a synonym dictionary to train the word embeddings on the Question-Answer corpus, which can express the semantic information of words. Second, we utilize self-attention mechanism to capture useful features between words of a sentence, and then feed the attention representations to LSTM [4] to compute semantic representations of the question sentence. The sentence

representations obtained by the above approaches contain more relationships and latent semantic information than just using word embeddings. Intuitively, it is a better way to handle the sentence features extraction issue in the particular domain. Finally, we exploit CNN [5] to extract the special features from the semantic representations of question sentences, and produce the final classification results through the fully-connected layer and the softmax layer. The experimental results demonstrate that the proposed hierarchical hybrid deep network for question classification can improve the average accuracy remarkably.

II. BACKGROUND

The goal of question classification is to classify the question to the anticipated type of the answer. In previous works, empirical rule-based and statistical-based approaches have been applied to the question classification tasks for decades. The rule-based approach generally exploits large-scale pre-defined rules to determine the type of question, which needs a lot of expertise [6,7]. The statistical-based approach is more scalable and versatile than the rule-based approach. [8] combined lexical, syntactic and semantic features to train three different classifiers: K-Nearest Neighbors classifier, naive Bayes classifier, and SVM to classify the question. Besides, many available means like HowNet [9] were introduced to improve the performance of the SVM-based method in the work. However, all of these use syntactic constituency parsing on the input when a trained classification model is applied. Meanwhile, the feature vector space is usually sparse [10].

Thanks to recent advances in deep learning, sentence classification has reached impressive performance. [11] defined a one-layer CNN architecture that uses pre-trained word vectors as inputs and had achieved state-of-the-art results across several datasets. [12] proposed a CNN architecture with multiple convolution layers, positing latent, dense and low-dimensional word vector as inputs. The most important point is that his model adopted a k -max pooling strategy, in which the maximum k values were extracted from the entire feature map and the relative order of these values was preserved. [13] combined high-order n -grams with CNN which contains multiple convolution layers and multiple pooling units associated with different regions. State-of-the-art performances on sentiment classification and topic classification were achieved using this approach. Unlike the above studies only use CNN, we build a hierarchical hybrid deep network consists of self-attention, bi-directional LSTM and CNN layers for question classification.

Self-attention, sometimes called intra-attention is an attention mechanism relating different positions of a single

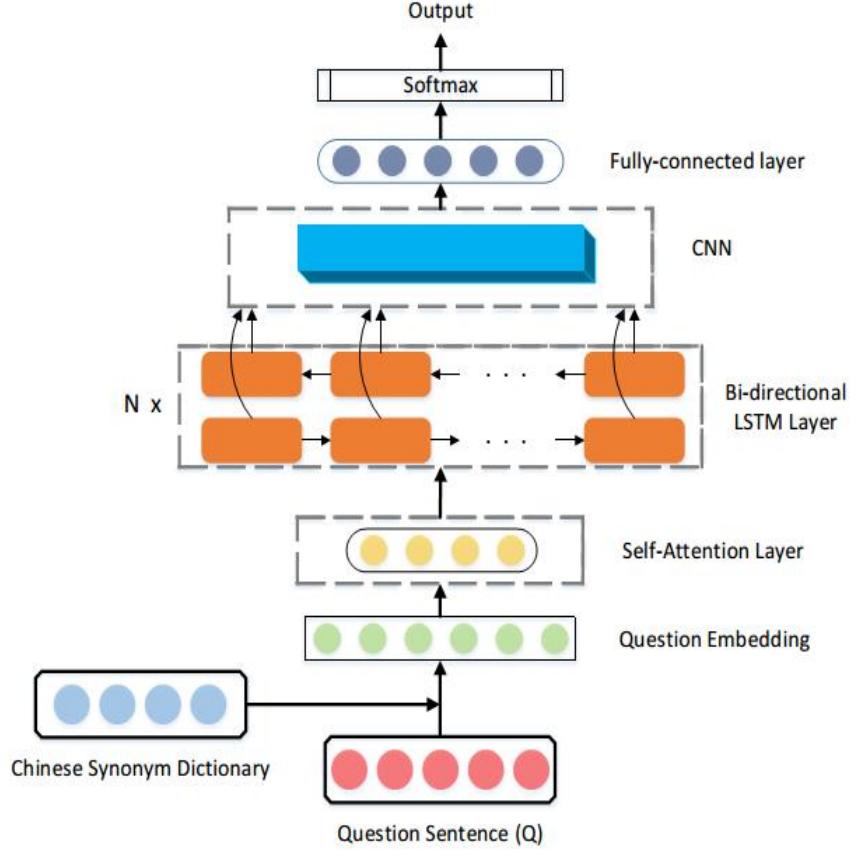


Figure 1. The overview of our model

sequence to achieve features between elements of the sequence. It has been used successfully in a variety of tasks such as learning sentence representations [14], reading comprehension [15], textual entailment [16] and abstractive summarization [17]. Bi-directional LSTM is able to encode the temporal information into representations. Besides, bidirectional architecture can access sequential and complete information about context before and after each time step in a sequence, and obtain the deep bidirectional context representations [18]. CNN is firstly proposed by LeCun for image processing [5], and has also been introduced to address sequential data. For instance, extracting robust and informative features from the sequential inputs [19-22].

It is worth noting that, research on question classification using deep learning methods is still in its infancy. Accordingly, we propose a novel network architecture to classify questions of QA. In the following sections, we describe the details of the proposed model, experiments and related discussions.

III. MODEL ARCHITECTURE

A QA text is composed of question and answer, which is different from normal text. Consequently, we have to process QA text and transform questions from sentences into embeddings which are fed into the hybrid network model for question classification. The self-attention layer of the proposed model is capable of achieving latent relationships between word vectors of the question embeddings. Multiple bi-directional LSTM layers are applied to compute a representation of a question sentence using self-attention outputs. CNN layer plays the role of feature

extractor to feed better sequential feature representations into the fully-connected layer, and then we utilize the softmax layer to obtain the final classification results.

The proposed model follows this overall architecture which uses self-attention, stacked bi-directional LSTM, CNN, fully-connected and softmax layers are shown in Figure 1.

A. Question Embeddings

Pre-trained word embeddings are considered to be a part of modern language models, offering significant improvements over embeddings learned from scratch [23]. Similarly to other sequence models, we use learned embeddings to convert the input tokens to vectors of dimension d_m . Specifically, we first use THULAC to perform Chinese word segmentation in question texts which have been denoised, and then we utilize word2vec to train the word embeddings on the QA corpus. Inspired by the work [24], we add Chinese synonym dictionary called *Tongyici cilin* [25] as additional information to the model which can improve the quality of the word embeddings. More specifically, we turn the word embedding of one pair of synonyms into the same one which randomly chooses from these two embeddings. Finally, each sentence in question texts can be converted into a question embedding composed of trained word representations.

B. Self-Attention Layer

In order to capture the internal features of the question sentences for achieving higher classification accuracy, the

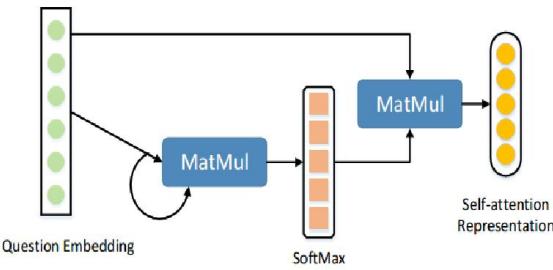


Figure 2. Question self-attention mechanism

self-attention mechanism is applied in our model. As described in Figure 2, assume that a question sentence $S_q = \{w_{q1}, w_{q2}, \dots, w_{qn}\}$ of QA corpus contains n words. And then we can convert the i -th word w_{qi} into word embedding $C_i^q = \{c_{i1}^q, c_{i2}^q, \dots, c_{id_m}^q\}$, where c_{ij}^q denotes the j -th element of i -th word embedding $C_i^q \in \mathbb{R}^{dm}$. Further, we can obtain a question embedding $S_q^e \in \mathbb{R}^{n \times dm}$ for question S_q . We compute the importance degree vector α of different words as:

$$T_q = \tanh(W_t(S_q^e(S_q^e)^T) + b_t) \quad (1)$$

$$\alpha = \text{softmax}(T_q) \quad (2)$$

where $T_q \in \mathbb{R}^{n \times n}$, W_t is the weight matrices, b_t is the bias.

After obtaining the self-attention weight of each word in a question sentence, we can compute the attention representation A_q of S_q^e by the following formula,

$$A_q = \alpha S_q^e \quad (3)$$

where $A_q \in \mathbb{R}^{n \times dm}$.

It is believed that the self-attention mechanism is able to capture latent information between words to get a higher quality context vector in the next layer.

C. Bi-directional LSTM Stack

LSTM can capture long-term dependencies in time series data and encode temporal information, which has been designed to address text classification [26, 27]. However, the major limitation is that the standard LSTM model is unidirectional, and this restricts the utilization of contexts. In contrast, bi-directional LSTMs can model the input sequence forward and backward, and then capture dependencies of past and future contexts. Figure 3 shows the details of one bi-directional LSTM layer, and this structure includes a forward LSTM and a backward LSTM. The forward LSTM reads the input $A_q = [a_{q1}, a_{q2}, \dots, a_{qn}]$ from left to right, where a_{qi} denotes the i -th vector of the attention representation A_q . At each time step t , the hidden state h_t is updated by the following formula.

$$h_t = f_{enc}(a_{qt}, h_{t-1}) \quad (4)$$

Similarly, the backward LSTM reads the input from right to left:

$$h'_t = f'_{enc}(a_{qt}, h'_{t-1}) \quad (5)$$

where f_{enc} and f'_{enc} are some nonlinear functions, h_t and h'_t are the hidden states of forward and backward LSTM at time t , respectively.

In this work, we employ $l=2$ bi-directional LSTM layers. For each of them, we apply $L2$ regularization to improve performance significantly. Then the sentence representation with more context semantic information would be obtained by the above structure, which is used as the input of the next CNN layer.

D. CNN Layer and Classification

In our approach, the CNN layer plays the role of feature extractor and the adopted CNN consists of two sub-layers: one convolutional layer and one pooling layer. The convolutional layer slides the filters over the whole inputs to generate feature mapping. In our model, the one-dimensional convolution operation is used along consecutive sentence representations, and we also employ a set of filters of varying widths to extract informative character patterns. Specifically, we exploit $N_f=3$ filters with filter size $p=3, 4$ and 5 . And then the pooling layer is applied to extract the most vital features from each feature mapping. In order to obtain the most significant features, we apply the *max-pooling* strategy to the outputs from the convolutional layer. This procedure selects the most salient features to give a final feature vector v . In short, we can capture robust semantic features of a question through the CNN layer and prepare for the final classification.

After obtaining the feature vector for a question, we then feed it into the fully-connected dense layer to seek a higher-level representation. The computation in this layer is given by:

$$u = g(W_{fc}v + b_{fc}) \quad (6)$$

where u is the output of dense layer, the function $g()$ is set to be *ReLU*, W_{fc} and b_{fc} denote the transformation matrix and the bias term, respectively.

Finally, we put the representation u into the *softmax* layer to achieve the conditional probability distribution:

$$y = \text{softmax}(W_s u + b_s) \quad (7)$$

where W_s and b_s are parameters of this layer. The label corresponding to the value with the highest probability in y stands for the final predicted category for a question sentence. Based on the above, we can try to predict and generate the real classification result considered by our model.

E. Training

Generally, the cross-entropy loss function is used to measure the similarity between predictions and actual values. We minimize the cross-entropy loss function to train the proposed model:

$$J(\omega) = - \sum_i^N [\Delta(y_i, y'_i)] + \frac{\lambda}{2} \|\omega\|_2^2 \quad (8)$$

where ω is training parameter, N is the number of training samples, y_i and y'_i are real label vector and output probability vector of the i -th sample respectively, Δ is a measure of discrepancy between these two vectors, the sum of token-level cross-entropy losses in our case. λ is a parameter of $L2$ regularization.

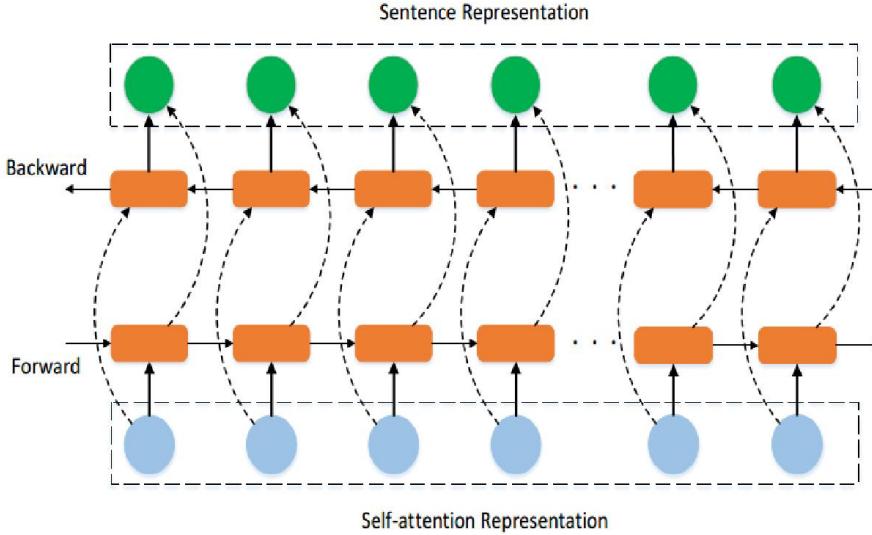


Figure 3. One bi-directional LSTM layer

Furthermore, we used the *Adam* optimizer [28] with $\beta_1 = 0.93$, $\beta_2 = 0.97$ and $\epsilon = 10^{-9}$. We also varied the learning rate over the course of training according to the formula:

$$lrate = slrate \times 0.5^{(global_step / decay_steps)} \quad (9)$$

where *slrate* is initial value for the learning rate, and *decay_steps* is the rate of learning rate decay. We also employ *dropout* to relieve the possible *over-fitting* problem during training.

IV. EXPERIMENTS

A. Descriptions of Dataset

To have a comprehensive understanding about the capacity of our proposed approach, we adopt the agricultural domain dataset to carry out experiments. More specifically, we select five categories of data from the website "Nongye Wenwen"¹ (NW) as the train and test sets. Our agricultural dataset has a total of 10139 samples and five labels, including "Livestock breeding", "Agricultural machinery", "Fruiter planting", "Agricultural materials" and "High-tech related". Statistics of data are shown in Table I.

Besides, we perform denoising and the word segment using THULAC to ensure the quality of the dataset. And then we replace related synonyms of questions using *Tongyici cilin* and produce new question sentences by combining synonyms with original questions, respectively. Finally, we utilize the open source toolkit word2vec to train word embeddings as inputs of our model. Considering the balance of data, 15% of the samples are randomly selected from each category as the test set, and the remaining data are used as the training set.

B. Experimental Setup

We evaluated our approach by comparing with several baseline methods. Both SVM [29] and CNN are typical classification approaches, and we perform them on the dataset as two baselines. However, the above two methods can not address sequential data. Thus we combine LSTM and CNN to encode temporal information of data and then classify questions. Sometimes question sentences are complicated, we have to consider the previous and future contexts of each

time step. Bi-directional LSTM can capture dependencies of past and future contexts, we put it on the CNN layer as another baseline method. Besides, the pooling layer can compress generated feature mapping to produce significant features, and affect the final predicted results. We change *max pooling* strategy to *k-max pooling* [12] in the CNN layer of the above approach, i.e. combination of bi-directional LSTM and CNN which uses *k-max* pooling.

For our proposed model, we exploit self-attention and bi-directional LSTM to compute the representations of question sentences. And then CNN, the fully-connected layer and the *softmax* layers are used to produce the final classification result. Similarly, we utilize *k-max* pooling strategy to obtain more significant features. Furthermore, the performance is evaluated by using *Accuracy* for all the methods.

The following methods will be compared in our experiments:

- SVM: Support Vector Machine based approach;
- CNN: Convolutional Neural Network based method;
- LMPC: This baseline method employs both a one-layer LSTM and CNN with max-pooling;
- BLMPC: This baseline approach puts question sentences into a two-layer Bi-LSTM and then CNN with *max-pooling* is used to classify;
- BLKMPC: This baseline method combines a two-layer Bi-LSTM and CNN with *k-max* pooling together for question classification task;
- Our Model: This is our proposed approach contains the self-attention layer, a two-layer Bi-LSTM and CNN with *k-max* pooling.

C. Results and Discussions

In this section, we show a comparison of our model with several benchmark methods and experimental results on the NW dataset are shown in Table II. In our work, experimental results of all models are evaluated using the classification accuracy.

¹ <http://wenwen.yl01.com/list-1.html>

TABLE I. "NONGYE WENWEN" DATA STATISTICS

Category	Label	Train	Test	Total
livestock breeding	0	1701	300	2001
agricultural machinery	1	1713	302	2015
fruiter planting	2	1723	304	2027
agricultural materials	3	1707	301	2008
high-tech related	4	1775	313	2088
Sum	-	8619	1520	10139

From the results, we observe that:

- Our model obtains a very competitive result compared to the baseline systems. More specifically, the classification accuracy of our proposed approach achieves an improvement of 9.37 percent compared to SVM and is 7.57 percent higher than CNN based method. Although the accuracy of our model is not much higher than other baseline methods except SVM and CNN, it has increased by at least 0.2%. The experiment results demonstrate that deep models are able to learn meaningful and discriminative representations from question sentences.
- From Table II, the accuracy of the LMPC method is higher 6.81 percent than CNN based method, indicating that LSTM can enable the model to capture long-term dependencies of sentences and encode them into the abstract representations. We also can find that the BLMPC method performs better than the LMPC method. It is obvious that bi-directional LSTM performs slightly better than LSTM, which owing to LSTM can only access the previous contexts but bi-directional LSTM can encode the question sentences in two directions to obtain better sentence representations.
- Moreover, comparing the classification accuracy of the BLKMPc method with the BLMPC method, we can observe that *k-max* pooling is more effective to question classification than *max* pooling strategy in the CNN layer. Noted that, our proposed hybrid deep network approach combining self-attention mechanism, bi-directional LSTM stack and CNN layers outperforms the above several outstanding baseline methods. Meanwhile, it also indicates that the attention mechanism can effectively capture the latent information between words dynamically and compute a meaningful representation of a question sentence.

It is interesting to observe that our proposed model significantly outperforms all baseline methods across this little training data. Intuitively, it is reasonable to believe that our hybrid deep network model can perform better on large-scale dataset.

V. CONCLUSION AND FUTURE WORK

In this paper, we presented a novel hierarchical hybrid deep network architecture, taking the place of general classification algorithms such as SVM most commonly used in question classification tasks. In our model, self-attention is firstly designed to capture useful features

TABLE II. CLASSIFICATION RESULT ON *NW* DATASET.

Methods	Accuracy(%)
SVM	81.63
CNN	83.43
LMPC	90.24
BLMPC	90.77
BLKMPc	90.80
Our Model	91.00

between words of question sentences, and then stacked bi-directional LSTM layers are applied to extend question embeddings depending on the obtained attention representations. Finally, the CNN layer can enable the model to extract question sentence features, and then classification results are achieved by the fully-connected layer and the *softmax* layer. The evaluation of the results on our agricultural dataset shows that the proposed hybrid deep network approach significantly outperforms all baseline methods.

In future work, we plan to verify our method with more datasets from different domains. Besides, joint learning may be more effective and efficient in question classification tasks.

ACKNOWLEDGMENT

This work is supported in part by the National Natural Science Foundation of China under Grant No. 61572462 and the 13th Five-year Informatization Plan of Chinese Academy of Science, Grant No. XXH13505-03-203.

REFERENCES

- [1] Li X, Roth D. Learning Question Classifiers [J]. Proc. COLING-2002, Taipei, Taiwan, 2002, 12(24):556—562.
- [2] Maosong Sun, Xinxiang Chen, Kaixu Zhang, Zhipeng Guo, Zhiyuan Liu. THULAC: An Efficient Lexical Analyzer for Chinese. 2016.
- [3] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality [C]// International Conference on Neural Information Processing Systems. Curran Associates Inc. 2013:3111-3119.
- [4] Hochreiter S, Schmidhuber J. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8):1735-1780.
- [5] Cun Y L, Boser B, Denker J S, et al. Handwritten digit recognition with a back-propagation network[J]. Advances in Neural Information Processing Systems, 1990, 2(2):396--404.

- [6] Magnini, B., Negri, M., Prevete, R., Tanev, H.: Mining Knowledge from repeated Co-occurrences: DIOGENE at TREC-2002 Proceedings of the Eleventh Text Retrieval Conference (TREC-2002), Gaithersburg, MD. (2002a).
- [7] Hovy E, Gerber L, Hermjakob U, et al. Toward semantics-based answer pinpointing [C]// International Conference on Human Language Technology Research. Association for Computational Linguistics, 2001:1-7.
- [8] Mishra M, Mishra V K, Sharma H R. Question classification using semantic, syntactic and lexical features [J]. International Journal of Web & Semantic Technology, 2013, 4(3): 39.
- [9] Xu S, Cheng G, Kong F. Research on question classification for automatic question answering [C]//Asian Language Processing (IALP), 2016 International Conference on. IEEE, 2016: 218-221.
- [10] Zhen L, Wang X, Yang S. Overview on question classification in question-answering system [J]. Journal of Anhui University of Technology (Natural Science), 2015, 32(1): 48-54.
- [11] Kim Y. Convolutional neural networks for sentence classification [J]. arXiv preprint arXiv:1408.5882, 2014.
- [12] Kalchbrenner N, Grefenstette E, Blunsom P. A convolutional neural network for modelling sentences [J]. arXiv preprint arXiv:1404.2188, 2014.
- [13] Johnson R, Zhang T. Effective use of word order for text categorization with convolutional neural networks [J]. arXiv preprint arXiv:1412.1058, 2014.
- [14] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need [C]//Advances in Neural Information Processing Systems. 2017: 5998-6008.
- [15] Cheng J, Dong L, Lapata M. Long Short-Term Memory-Networks for Machine Reading[J]. 2016.
- [16] Paulus R, Xiong C, Socher R. A Deep Reinforced Model for Abstractive Summarization[J]. 2017.
- [17] Parikh A P , Täckström, Oscar, Das D , et al. A Decomposable Attention Model for Natural Language Inference[J]. 2016.
- [18] Schuster M, Paliwal K K. Bidirectional recurrent neural networks [J]. IEEE Transactions on Signal Processing, 1997, 45(11): 2673-2681.
- [19] Gehring J, Auli M, Grangier D, et al. Convolutional Sequence to Sequence Learning[J]. 2017.
- [20] He H, Gimpel K, Lin J. Multi-perspective sentence similarity modeling with convolutional neural networks[C]//Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 2015: 1576-1586.
- [21] Bai S, Kolter J Z, Koltun V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling[J]. arXiv preprint arXiv:1803.01271, 2018.
- [22] Zhang X, Zhao J, LeCun Y. Character-level convolutional networks for text classification[C]//Advances in neural information processing systems. 2015: 649-657.
- [23] Turian J, Ratinov L, Bengio Y. Word representations: a simple and general method for semi-supervised learning[C]//Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics, 2010: 384-394.
- [24] Mei L, Zhou Q, Zang L, et al. Merge information in hownet and TongYiCi CiLin[J]. Journal of Chinese Information Processing, 2005, 19(1): 63-70.
- [25] Jiaju M, Yiming Z, Yunqi G, et al. Tongyici cilin [J]. Shanghai Dictionary Publication, 1983.
- [26] Zhou C, Sun C, Liu Z, et al. A C-LSTM neural network for text classification[J]. arXiv preprint arXiv:1511.08630, 2015.
- [27] Lai S, Xu L, Liu K, et al. Recurrent convolutional neural networks for text classification[C]//Twenty-ninth AAAI conference on artificial intelligence. 2015.
- [28] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.
- [29] Chang C C, Lin C J. LIBSVM: A library for support vector machines[J]. ACM transactions on intelligent systems and technology (TIST), 2011, 2(3): 27.

A Study on the Robustness of Pitch Range Estimation from Brief Speech Segments

Wenjie Peng, Kaiqi Fu, Wei Zhang, Yanlu Xie, Jinsong Zhang

Beijing Advanced Innovation Center for Language Resource

Beijing Language and Culture University

Beijing, China

{wenjayep, kaiq.fu}@gmail.com, wei_zhang_mail@163.com, {xieyanlu, jinsong.zhang}@blcu.edu.cn

Abstract—Pitch range estimation from brief speech segments is important for many tasks like automatic speech recognition. To address this issue, previous studies have proposed to utilize deep-learning-based models to estimate pitch range with spectrum information as input [1-2]. They demonstrated it could still achieve reliable estimation results when speech segment is as brief as 300ms. In this work, we further investigate the robustness of this method. We take the following situation into account: 1) increasing the number of speakers for model training hugely; 2) second-language(L2) speech data; 3) the influence of monosyllabic utterances with different tones. We conducted experiments accordingly. Experimental results showed that: 1) We further improved the accuracy of pitch range estimation after increasing the speakers for model training. 2) The estimation accuracy on the L2 learners is similar to that on the native speakers. 3) Different tonal information has an influence on the LSTM-based model, but this influence is limited compared to the baseline method. These results may contribute to speech systems that demanding pitch features.

Keywords-component; *pitch range estimation; LSTM; brief speech segments; L2; tone*

I. INTRODUCTION

In human speech communication, pitch carries important information, and different people have different pitch range. Pitch range estimation could benefit many speech systems. Previous research showed that human beings could have an accurate estimation on pitch range from a brief (<50ms [3]) speech segment [4-7]. Inspired by these findings, W. Zhang et al. proposed to employ a LSTM model with spectrum information as input to estimate speaker's pitch range targets [1]. Experimental results showed that they could achieve a reliable pitch level estimation result with low (<2.5%) mean absolute error rate (MAPE) with speech segments as brief as 300ms (about 1~1.5 syllables). Different from [1], Q. Zhang et al. have utilized multi-feature and multi-task learning deep neural network (MTL-DNN) to estimate pitch range [2]. Both studies have demonstrated that pitch range estimation could make use of deep learning technique with spectrum information as input. These findings could benefit speech systems that demanding pitch features.

In this paper, we further investigate the robustness of the method in [1] under some challenging circumstances. One of them is the influence of big scale dataset, since they trained the model on a relative small dataset previously. From the view of speech production, when the number of speakers increased, pitch range will have a larger diversity due to its speaker-dependent nature, which may put more challenges to the estimation task. From the view of speech

technology, deep-learning-based models often gain from large scale dataset. Considering these two aspects, the influence of a larger scale dataset on this model seems to be unclear.

Another scenario is applications of speech technology involved with pitch features, say computer-assisted pronunciation training(CAPT) system [8-9]. In CAPT system, it is often hard to obtain massive L2 data for model training. In addition to build a special L2 corpus (i.e. [10]), it is common to train models on a native corpus then evaluate them on the L2 data. The acoustic space between native speakers and L2 learners, however, differs hugely. For CAPT systems demanding pitch features, it is well worth investigating whether this method could still provide reliable estimation results in this situation.

Pitch range estimation is intrinsically difficult from brief speech segments especially when speech samples could not contain enough pitch variations. There are four basic lexical tones (high-level, mid-rising, low-dipping and high-falling) in Chinese. In Chinese monosyllabic utterances, the phenomenon of co-articulation between syllables will disappear while the influence of tonal information on F0 values will be much more prominent. [11] found that F0 range increases gradually from high-level to high-falling. Thus it is reasonable that speech samples with more high-fallings will result in a larger F0 range. Pitch range measures based on the distribution of F0 will be affected by different tones a lot. The LSTM-based method, however, takes spectrum information rather than F0 values as input. It is unclear whether or to what extent different tonal information affect the estimation results.

Considering the above three aspects, we conducted several experiments accordingly. We first increased speakers hugely compared to previous experiment for LSTM model training. To investigate the effect of L2 and different tonal information on the LSTM-based model, we then evaluated it on a Chinese L2 of Japanese corpus and a corpus with Chinese monosyllabic utterances respectively.

This paper is organized as follows. We first review the proposed method in section 2. In section 3 and section 4, we give details about the experiments for model training and test results. Two evaluation experiments will be presented in section 5 and section 6. Discussion and conclusions will be given in section 7.

II. METHOD REVIEW

A. Pitch Range

There is a common consensus that pitch range varies along two dimensions: pitch level and pitch span [12]. Pitch level refers to the overall pitch height of voice while pitch span represents the range within which pitch varies.

To quantify pitch level, many studies suggested using the mean F0 or median F0. As for pitch span, some long term distributional(LTD) measures have been adopted based on an analysis of F0 distribution within a speaker's voice, which includes the difference between the 95th and 5th percentile, difference between the 90th and 10th, maximum minus minimum F0, four standard deviations around the mean. In addition to LTD measures, 'linguistic' measures have also been proposed and adopted [13-14], which make use the specific landmarks in the F0 contour.

B. LSTM Model

Recurrent Neural Networks(RNNs) have been successfully applied in speech processing due to their ability to use the contextual information when mapping between the input and output [15]. Unfortunately, the range of context for standard RNN that can be accessed is very limited due to the influence of a given input on the hidden layer. This will affect the output, either decays or blows up exponentially as it loops through the whole network's recurrent connections.

To address the above issues, many attempts have been proposed. Among these methods, LSTM could provide an effective way to make output reliable by adding extra memory blocks. These blocks enable the model store and access information over long periods of time, thereby preventing the outputs from vanishing or exploding.

III. EXPERIMENTAL SETUP

A. Speech Data

We used the open-source AISHELL-2 [16] Mandarin Corpus for model training. This corpus is split into three parts, namely training, evaluation and test. The details about the speakers and content of speech are shown in Table I. All the speech data was recorded in a quiet environment.

We first tried to train the model with all the speakers in the training dataset, but ended up with poor performance for pitch range estimation. After observing the data, we found that there exists an unbalanced distribution of pitch range targets in the training dataset.

To solve this issue, we did preprocessing on the dataset. We only chose part of the whole data for model training instead. For pitch level estimation, we first located the max and min pitch level targets, and then split speakers within the range (max-min) into 100 groups equally. We then chose speakers within each group randomly. Preprocessing for pitch span estimation was the same as pitch level, except that the number of speakers we chose within each group differed, at most 12 for pitch level and 10 for pitch span respectively. Finally, we got 899 speakers for pitch level training and 693 speakers for pitch span training with 100 utterances per speaker. For testing, we used the test part in AISHELL-2 corpus, including 10 speakers (5 male and 5 female) with 1000 utterances in total. There is no overlap between training and test dataset neither at the speaker-level nor the utterance-level.

At the stage of training, 20% of training data will be used as validation set, which is a common approach in machine learning.

TABLE I. DESCRIPTIONS ON THE TRAINING DATASET

Content of speech	voice commands, places of interest, entertainment, finance, technology, sports, English spellings and free speaking without specific topic
Speaker information	There are 845 males and 1146 females 1991 speakers in total in the training dataset. The age of speaker varies from 11 to 40. As for the accents, there are 678 speakers using Southern ones, 1293 using Northern ones and 20 speakers using other ones while recording.

B. Features

We first did intensity normalization on the raw audio files using Praat [17] with default settings. We then utilized Kaldi [18] toolkit to extract time-spliced-40-dimensional Fbank features per utterance with 25ms frame length and 10ms frame shift. The raw Fbank features were further processed using voice activity detection technique to remove the silent interval segments.

F0 tracking was performed using STRAIGHT [19] algorithm at 1ms interval with a specific range of F0. We set F0 range of 50-300Hz for male and 75-500Hz for female. In this study, we first transformed the extracted F0 into logarithmic domain with base 10 rather than the raw F0 values, then we calculated the mean and standard deviation of the transferred results at the speaker-level to represent pitch level and pitch span targets respectively.

We conducted the above operations all the same on the dataset we used in this work.

C. Evaluation criteria

We adopt MAPE to evaluate the performance of LSTM-based model. The formula of MAPE is defined as below:

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{(y_i - \tilde{y}_i)}{y_i} \right| * 100 \quad (1)$$

where N is the total number of input samples, y_i is the i-th pitch range target (pitch level or pitch span) to be estimated, and \tilde{y}_i is the corresponding model prediction. The less value MAPE is, the more coherence that pitch range is estimated with measurements from logF0.

D. LSTM model setting

We used Keras [20] toolkit for model training. Following previous model settings in [1], the main parameters of our model are listed as below:

TABLE II. MAIN PARAMETER SETTINGS IN LSTM

Parameter	Batch length	Batch size	Layer size	Hidden units
value	30	50	3	100

We kept the main parameter settings the same with [1], but increased the hidden size from 50 to 100 due to the increased size of training data. Each input to the model consists of 300ms speech sample, model will not give an output until the end of last time step. The gradient will update once per 50 samples. We trained two separate LSTM models with 3 LSTM layers for pitch level and pitch span estimation. The two models have the same parameter settings but differ in pitch range estimation targets, say the mean logF0 and the standard deviation of logF0.

IV. RESULTS

The results of evaluation on our test dataset are as below.

TABLE III. EVALUATION RESULTS OF PITCH RANGE ON AISHELL-2 TEST DATASET

	Pitch level	Pitch span
MAPE	1.09%	19.37%

It can be seen from Table III that we achieved a fairly low estimation error rate of 1.09% for pitch level, while 19.37% for pitch span. Although pitch level estimation result seems more promising than that in [1] (1.09% vs 2.3%), it does not make sense to compare these two directly due to the different datasets we used. As for pitch span, the error rate was much larger than that of pitch level, which may suggest that spectral structure has higher correlation with pitch level than with pitch span [1].

To verify whether increasing training dataset could decrease the error rate, we split the above two training dataset into subsets to see the effect of training data size. We randomly split the above two training datasets into three subsets, namely 25%, 50%, 75% of the original ones. We trained models on these subsets with the same parameters settings. To get a more stable results, we repeated training on each subset 6 times, and calculated the average MAPE as the corresponding result. Table IV show the average of MAPE within each training set.

TABLE IV. MAPE OF PITCH RANGE ESTIMATION RESULTS ON DIFFERENT SIZE OF TRAINING DATASET

	25%	50%	75%	100%
Pitch level	1.35%	1.25%	1.17%	1.09%
Pitch span	24.53%	19.54%	19.25%	19.37%

It can be seen from Table IV that when training dataset increases, the estimation error rate will decrease especially for pitch level estimation. For pitch span estimation, the error rate drops at first, and get the lowest error rate with 75% of the whole training dataset. The error rate increase slightly when training on the whole dataset, which indicates that a relative large dataset is good enough for pitch span model training. Both results showed that large-scale dataset for model training will bring performance gain.

With the trained model on the whole dataset, we did two other evaluation experiments.

V. EVALUATION ON L2 CORPUS

A. Data

We first evaluated this method on Conversational Chinese 301 [21]. 19 native Japanese speakers (10 males, 9 females) were taken part in this study and they were told to read the Mandarin materials in a natural way. The average number of utterance per speaker is 301.

B. Evaluation Results and Analysis

To make the method comparable, we calculated the mean and standard deviation of logF0 within a speaker as the ground-truth labels for pitch level and pitch span respectively. We compared estimation results from the model with the ground-truth labels to investigate model performance under the effect of second-language, which are shown as below in Fig. 1 and Fig. 2 respectively.

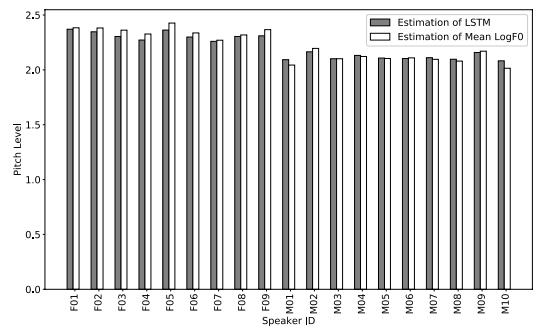


Figure. 1 Measurements of pitch level by LSTM and Mean logF0 (i.e. M02 means Japanese male with id 02, while F02 means Japanese female with id 02)

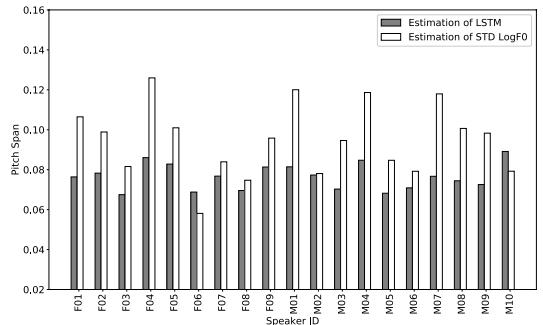


Figure.2 Measurements of pitch span by LSTM and Standard Deviation of logF0

It can be seen from Fig. 1 that there was a high coherence between model prediction and the measures of mean logF0. Pitch level varies across different speakers, but model's estimation results are very close to the ground-truth labels per speaker. Paired t-test shows that there is no significant difference between these two results ($t=-1.476$, $df=18$, $p=0.157$, Estimation of LSTM – Estimation of Mean LogF0). As for pitch span showed in Fig. 2, there seems to be a large difference between these two methods.

To quantify such difference, we calculated MAPE for pitch level and pitch span estimation respectively. And the MAPE for pitch level is 2.11% while 22.28% for pitch span, which is slightly higher than the error rate tested on the AISHELL-2 test dataset. The overall estimation MAPE of pitch level and pitch span tested on native Japanese

speakers seems to be close to that tested on the native Mandarin speakers. These results suggested that this method could still provide a relative reliable estimation results under the influence of second-language.

VI. EVALUATION ON MONOSYLLABIC-UTTERANCE CORPUS

A. Data

For this evaluation, we used part data from the BLCU-SAIT corpus [22]. We chose 35 native speakers (15 males, 20 females) with monosyllabic utterances per speaker for evaluation. The average number of monosyllabic utterances is 1085. The speech content covers the four basic tones in Mandarin (we removed some utterances with a neutral tone)

B. Baseline Method

We first established a baseline method to evaluate the performance of LSTM model under the effect of different tonal information. In this evaluation experiment, we used the mean and standard deviation of logF0 calculated at the monosyllabic-utterance-level as estimation results for baseline. We did not adopt ‘linguistic’ measures as baseline because it is hard to spot specific landmarks in monosyllabic utterances especially in utterances with Tone1.

C. Ground-truth Labels

In order to compare our model with baseline method, we then need to specify the ground-truth of pitch range per speaker. Unlike the ground-truth setup in the former evaluation experiment, we did this based on extra data that speaker produced. In practice, we did the calculation on two other part data belonging to that speaker, which includes 103 declarative sentences, 237 bi-syllable utterances together with the above monosyllabic utterances.

D. Evaluation Results and Analysis

Table V and Table VI show the MAPE results of pitch level and pitch span estimated from the baseline method and the LSTM-based method respectively.

TABLE V. COMPARISONS OF MAPE ON PITCH LEVEL ESTIMATION BETWEEN BASELINE AND LSTM MODEL

	Tone1	Tone2	Tone3	Tone4	Overall
Baseline	3.83%	1.25%	3.82%	1.71%	2.75%
LSTM	2.89%	1.74%	2.12%	2.19%	2.23%

TABLE VI. COMPARISONS OF MAPE ON PITCH SPAN ESTIMATION BETWEEN BASELINE AND LSTM MODEL

	Tone1	Tone2	Tone3	Tone4	Overall
Baseline	78.75%	44.06%	38.68%	25.28%	47.04%
LSTM	32.79%	27.14%	22.99%	19.08%	25.59%

As is shown in Table V, LSTM-based method achieved a slightly lower error rate for pitch level estimation compared against baseline method (2.23% vs 2.75%). The top1 and the lowest error rate came from the cases of Tone1 and Tone2 respectively in both methods. These may be associated with the characteristics of 4 different

Mandarin lexical tones, among which the production of Tone1 is at high level while Tone2 is at a relative median level.

As for pitch span showed in Table VI, LSTM-based model also outperformed the baseline method with overall MAPE of 25.59% compared against 47.04%. Besides, both methods achieved the lowest error rate in the case of Tone4. This may be due to the fact that Tone4 is involved with a larger range of F0 values compared to the rest three tones.

The varied estimation error rate among four tones indicated that different tonal information has an influence on the estimation results. Compared to the baseline method, however, LSTM-based model’s lower variance of error rate among different tonal information suggested that such influence was relative limited than that in the baseline method. This difference may be due to the fact that LSTM-based model takes spectrum information as input rather than F0 values.

VII. DISCUSSIONS AND CONCLUSIONS

In this study, we did the following three attempts to further investigate the robustness of LSTM-based pitch range estimation method.

First, the large-scale data. We show large-scale data will decrease the estimation error rate especially for pitch level estimation. With the increased speakers, we further improved the accuracy of pitch range estimation.

Second, the effect of second-language. Evaluation results on the Mandarin L2 of Japanese speakers showed that LSTM-based method could still maintain a high accuracy for pitch level estimation under the influence of second-language. The MAPE for pitch range estimation on L2 data is similar to that on the native data.

Third, the influence of different tonal information. Experimental results of Chinese syllabic utterances showed that LSTM-based method could be affected under such effect, but the influence is relatively limited comparing to the baseline method.

These results further verified that the robustness of pitch range estimation by utilizing LSTM with spectrum information as input. The experimental results suggest that it could contribute to speech systems involved with pitch features. However, the MAPE tested on monosyllabic utterances is larger than that on AISHELL-2 native corpus, which may be due to the mismatch between different corpus. Besides, the error rate of pitch span estimation is much larger than that of pitch level in our experiment. Future work should focus on the above two issues.

ACKNOWLEDGEMENT

This study was supported by Advanced Innovation Center for Language Resource and Intelligence (KYR17005), the Fundamental Research Funds for the Central Universities (16ZDJ03, 18YJ030006), and the project of "Intelligent Speech Technology International Exchange". Jinsong Zhang is the corresponding author.

REFERENCES

- [1] W. Zhang, et al. "LSTM-Based Pitch Range Estimation from Spectral Information of Brief Speech Input." *2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, 2018.
- [2] Q. Zhang, et al. "Pitch Range Estimation with Multi features and MTL-DNN Model." *2018 14th IEEE International Conference on Signal Processing (ICSP)*. IEEE, 2018.
- [3] C. Y. Lee, "Identifying isolated, multispeaker Mandarin tones from brief acoustic input: A perceptual and acoustic study," *Journal of the Acoustical Society of America*, vol. 125, no. 2, pp.1125-1137, 2009.
- [4] Moore, B. Corinne, and A. Jongman, "Speaker normalization in the perception of Mandarin Chinese tones." *The Journal of the Acoustical Society of America* 102.3: 1864-1877, 1997.
- [5] D. N. Honorof and D. H. Whalen, "Perception of pitch location within a speaker's F0 range." *The Journal of the Acoustical Society of America*, vol. 117, pp. 2193–2200, 2005.
- [6] J. Bishop and P. Keating, "Perception of pitch location within a speaker's range: Fundamental frequency, voice quality and speaker sex," *Journal of the Acoustical Society of America*, vol.132, no. 2, pp. 1100-1112, 2012.
- [7] J. Kuang and M. Liberman, "Influence of spectral cues on the perception of pitch height," *Proceeding of ICPHS, Glasgow, UK*, 2015.
- [8] D. Chun, Signal analysis software for teaching discourse intonation, *Language Learning and Technology* 2, 61-77, 1998, <http://lilt.msu.edu/vol2num1/article4/index.html>
- [9] J. Kommissarchik, E. Komissarchik, (2000) Better Accent Tutor-Analysis and visualization of speech prosody, *Proceedings of InSTILL, Dundee, Scotland*, 86-89, 2000.
- [10] N. F. Chen et al., "Large-Scale Characterization of Non-Native Mandarin Chinese Spoken by Speakers of European Origin: An Analysis on iCALL," *Speech Communication*, 2016.
- [11] H. Liu, "The acoustic-phonetic characteristics of infant-directed speech in Mandarin Chinese and their relation to infant speech perception in the first year of life," pp. 3687-3687, 2003.
- [12] D. R. Ladd, "Intonational Phonology," *Cambridge: Cambridge University Press*, 1996.
- [13] I. Mennen, F. Schaeffler, and G. Docherty, "A methodological study into the linguistic dimensions of pitch range differences between German and English," *Proceedings of the 4th Conference on Speech Prosody, Campinas*. 2008.
- [14] D. J. Patterson, "Linguistic approach to pitch range modelling," 2000.
- [15] A. Graves, M. Abdel-rahman, and H. Geoffrey, "Speech recognition with deep recurrent neural networks," *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013.
- [16] J. Du, X. Na, X. Liu and H. Bu, "AISHELL-2: Transforming Mandarin ASR Research Into Industrial Scale," *arXiv preprint arXiv*, pp. 1808:10583, 2018.
- [17] P. Boersma, "Praat: doing phonetics by computer." <http://www.praat.org/> (2006).
- [18] D. Povey, et al. "The Kaldi speech recognition toolkit.", *No. CONF. IEEE Signal Processing Society*, 2011.
- [19] H. Kawahara, A. Chevigne and R. D. Patterson. "An instantaneous-frequency-based pitch extraction method for high-quality speech transformation: revised TEMPO in the STRAIGHT-suite," *Fifth International Conference on Spoken Language Processing*, 1998.
- [20] P. W. D. Charles, "KERAS," GitHub Repository, <https://github.com/charlespwd/keras>, 2013.
- [21] Conversational Chinese 301[M]. Beijing Language and Culture University Press, 2005.
- [22] W. Wang, X. Wei, J Yu, W. Wei, Y. Xie and J Zhang, "The BICU-SAIT Speech Corpus Of Non-Native Chinese," *Oriental COCOSDA 2018, Miyazaki, Japan, May*, 2018.

Are Scoring Feedback of CAPT Systems Helpful for Pronunciation Correction? --An Exception of Mandarin Nasal Finals

Rui Cai, Wei Wei and Jinsong Zhang*

Beijing Advanced Innovation Center for Language Resources
Beijing Language and Culture University

Beijing, China

cairui_blcu@163.com, wwei906@163.com, jinsong.zhang@blcu.edu.cn

Abstract—The scoring feedback of Computer Assisted Pronunciation Training (CAPT) systems facilitate learner's instant awareness of their problems, easily lead to more practices. But whether it is enough to instruct the learners to understand how to correct their errors is still unknown. To see in depth, the impacts from CAPT technology on language learning, and to investigate learner's correction strategy after receiving error warnings, this paper studies long term learning data of Chinese utterances by a number of CSL (Chinese as a Second Language) learners, with special efforts paid to the utterances of nasal Finals. The data resulted from a 3-week use of a CAPT APP, called “SAIT 汉语” for Chinese learning, by 10 learners with different mother tongues. Major findings include: 1) Improvements were seen with almost all kinds of phonemes, except nasal Finals; 2) Data analyses showed that the learners had tried to lengthen the nasal codas after they received error warnings, while Chinese native data shows a significant nasalization period before a short coda. These results suggest that the scoring feedback can be beneficial to pronunciation training in most cases, except for some special ones. For the sounds such as Chinese nasal Finals, more appropriate feedback method is desired.

Keywords-computer assisted pronunciation training; correction strategy; Chinese nasal Finals;

I. INTRODUCTION

Computer Assisted Pronunciation Training (CAPT) systems are known for their advantage of instant feedback for instruction and are becoming more and more popular. Nowadays most CAPT systems use GOP (goodness of pronunciation) score to evaluate pronunciation quality, in which the acoustic distances between the sounds of natives and second language (L2) learners are compared [1]. Such a feature can tell the learners whether their pronunciation is good or not based on a close or far distance measure. When reported with low scores learner will be instantly aware of their problems, and then can repetitively practice until they receive higher scores. But scores are usually not informative enough for learners to correct their mispronunciations. Therefore, many language teachers criticize corrective feedback from CAPT and doubt its efficiency in pedagogy practice [2].

Many researchers have tried to explore the effectiveness of CAPT systems from different perspectives. Mak [3] found that 77% of the learners believed that their pronunciation quality had been improved after using the CAPT system. Neri [4] and Hirata [5] used pre-test and post-test methods and found

that after CAPT training, learners' pronunciation improved significantly in the post-test. These previous studies made a general evaluation of the effectiveness of CAPT by using questionnaire or pre- and post- test methods. However, they did not answer whether all of the phonemes can be improved after CAPT training. And if there were some stubborn errors, what correction strategies adopted by learners resulted in the difficulty in correcting these special phonemes.

In order to make up for the shortcomings of previous methods, the present study attempts to use learning data to analyze learners' correction strategies. Learning data are the learner's behavior data collected automatically by CAPT system, including the results of the system's speech recognition, the system's scoring of each phoneme pronounced by learners, and audio recordings, etc. Through these learning data, we can evaluate the effect of the system more accurately according to the detailed correcting practice.

In this research, we focus on a Chinese learning APP which can provide scoring feedback on learners' pronunciation automatically. For the purpose to evaluate the performance of the APP we conduct two series of studies: the first one is to investigate the correction effectiveness of the APP among different phonemes. The second study aims at figuring out the reasons leading to the bad performance of Chinese nasal Finals in the first study. We attempt to answer what kind of corrective strategies are adopted by learners, which lead to the difficulty of CAPT training in the acquisition of nasal Finals.

II. STUDY I: CORRECTION EFFECTIVENESS ANALYSIS

In this study, we investigated the correction effectiveness of a Chinese learning APP named “SAIT 汉语”. Wei et al had introduced this system in detail [6]. We collected the learning data from a three-week APP teaching experiment, and then, found out the uncorrected mispronunciation.

A. Teaching Experiment

A three-week teaching experiment was conducted to test effectiveness of our system. 23 CSL (Chinese as a Second Language) students studying in Beijing Language and Culture University attended our experiment. All of the CSL learners were at the beginner level of Mandarin. Their average length of studying Mandarin was 0.71 years ($SD=0.29$).

During the teaching experiment, all participants were asked to complete pronunciation assignments in APP. With the CAPT system, learners could imitate the standard examples produced by CN (Chinese native) speakers and receive scoring feedback from the APP. Then students would practice the erroneous pronunciation repeatedly according to the feedback information of the APP.

B. Data Analysis

In this analysis, we first extracted the repetitive practices entries from the recording files, then, sorted out the repetitive practices of the Initials, Finals, and tones of each syllable from these entries. If the errors were corrected after several practices, the correction process would be regarded as a successful correction; if the errors were not corrected eventually, the correction process would be regarded as a failure correction. We used the proportion of successful correction in all corrections (P) and the number of repetitions required to correct pronunciation error, recorded as (N_T), as two indicators to evaluate the effect of pronunciation correction. Because the correction action was prompted by feedback information, these indicators could directly reflect the role of feedback information.

The speech recordings in our dataset contained 28,101 Chinese utterances which amounted to 21 hours of speech. To reduce the complexity, all of the utterances were monosyllable or disyllable words.

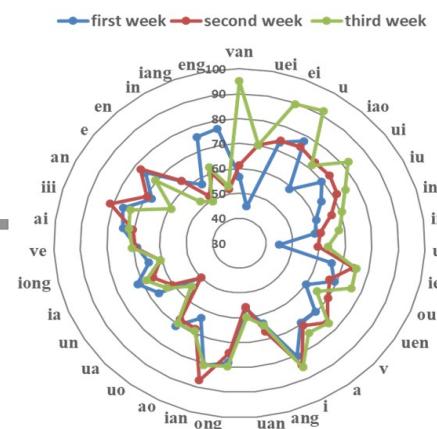
C. Results

TAB. 1 THE PROPORTION OF SUCCESSFUL CORRECTIONS IN ALL CORRECTIONS (P) AND THE NUMBER OF REPETITIONS REQUIRED TO CORRECT THE ERROR (N_T) OF INITIAL, FINALS, AND TONES.

	Initials	Finals	Tones	Mean
P (%)	85.9	84.6	79.2	83.2
N_T (times)	1.48	1.54	1.55	1.5

As shown in Tab.1, the average P is about 83.2%. This revealed that with the help of the APP most mispronunciations could be accurately corrected by learners. The average N_T is about 1.5 times, that is, only through one or two practice(s), the pronunciation of learners will significantly improve. The tones presented the worst performance in both P and N_T . The reason was that the tone training was little in the first three weeks. In addition, the performance of Finals was worse than that of Initials.

To see in depth, we took Finals as examples and calculated system's scorings of each Final to evaluate the learning trend during the three weeks.



The results of the trend of the systems' scoring can be seen from Fig. 1. Various level of improvements could be observed in most Finals phonemes which also suggested that for most cases the scoring feedback of CAPT system were effective. However, it was worth noting that some cases made a substantial decline after CAPT training especially for some Chinese nasal Finals like "en", "eng", and "ang".

We speculated that when there were inadequate instructions from scoring feedback, learners were likely to adopt wrong strategies in self-correction especially for some stubborn errors. Therefore, as the number of exercises increases, learners would deviate from the standard pronunciation instead.

III. STUDY II: CORRECTION STRATEGIES ANALYSIS

Based on the previous findings, this study aims at achieving a clear image about the correction strategies of learners. As the first step of the investigation, we mainly focused on two basic contrast pairs in Mandarin Nasal Finals: /an/-/aj/, /en/-/ej/ which showed a bad performance in the first study.

Finals ending with nasal codas are defined as nasal Finals. There are two contrastive nasal codas in standard Mandarin which are alveolar nasal /n/ and velar nasal /ŋ/ [7]. It is generally agreed that nasal Finals in Mandarin can be divided into three part: vowel nucleus, nasalized vowel (transition) and nasal coda [8].

A number of studies have revealed that for CSL learners there has always been a difficulty in learning Chinese nasal Finals [9][10][11]. Most previous works reported that CSL learners with different native backgrounds confused the nasal codas when they were producing or perceiving the nasal Finals. And they usually explained the reasons from the perspective of phonological contrast. The correction strategies from learning data of CAPT may provide a new view to study the traditional L2 acquisition questions.

A. Method

1) Data

The learning data used in this study were speech recordings of CSL learners while they were practicing the pronunciation. We filtered the data with the frequency of the learners' use and the quality of the recordings. 1,008 utterances read by 10 learners were eventually selected in our dataset which contained about 70% of the total data of nasal Finals. 10 CSL learners have five different mother tongue backgrounds (3 Urdu speakers, 3 Russian speakers, 2 Italian speakers, 1 Spanish speaker, and 1 Sinhalese speaker). None of these languages have phoneme contrast between alveolar nasal /n/ and the velar nasal /ŋ/.

2) Native Statistical Distribution

A native statistical distribution was established to compare with the pronunciation of CSL learners. The speech materials of CN (Chinese native) speakers (5 males and 5 females) were selected from BLCU-SAIT corpus. The entries read by CN speakers were same with CSL learners.

3) Annotation

The annotation was adopted to distinguish the vowel nucleus, transition (or nasalized vowel) and nasal coda in Finals separately. The annotation procedure could be observed in Fig. 2.

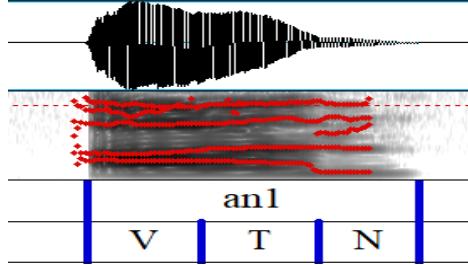


Fig. 2 An annotation example of vowel nucleus part(T), transition part (T) and nasal coda part (N).

In order to ensure the accuracy of the annotation, waveform and spectrum were taken into consideration. For instance, the waveform in nasal coda part is specific for the reduction in amplitude [12]. The formant in transition part will show a wider bandwidth [13] and there normally is a nasal formant between first vowel formant and second vowel formant [14]. When there was an inconsistency between the three factors, we would also annotate it with perception.

4) Parameters Measurement

Previous works have shown that the nasal coda in Mandarin can be considered as a nasalized part of the vowel [8] and it is obviously shorter than the vowel period [17]. The acoustic characteristics of nasal Finals are mainly determined by the second formant transition [15][16] and duration [17] in vowel part. Thus, we measured the following parameters:

- F2: The second formant in both vowel nucleus part and transition part (V+T), extracting 10 points on average for each syllable.
- D_V: The duration of the vowel nucleus part.
- D_T: The duration of the transition part.
- D_N: The duration of the nasal coda part

In order to eliminate the differences between the speakers, the absolute formant frequencies were converted to z-scores according to the equation of Lobanov's procedure [18].

B. Results

From the measurements, we calculated the three focused variables: F2_slope (the slope of ten F2 points), D%_T(D_T/D_V+D_T+D_N), D%_N(D_N/D_V+D_T+D_N).

1) Duration

The D%_N and D%_T are shown in Fig. 3 and 4.

It can be seen from Fig. 3 that D%_T are higher in CN speakers than in CSL learners for four nasal Finals. D%_T of CN speakers and CSL learners are compared by using an independent T-test, respectively. Results show that D%_T of CN speakers are significantly higher than CSL ($t=-5.596$, $p<0.001$, for /an/; $t=-8.947$, $p<0.001$, for /an/; $t=-3.107$, $p=0.006$, for /en/; $t=-2.416$, $p=0.027$, for /en/).

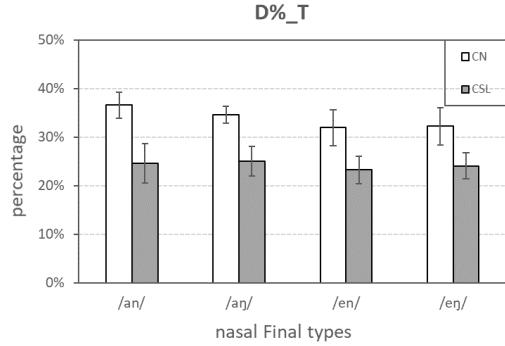


Fig. 3 D%_T of CN and CSL in four nasal Finals.

Manifestations are shown in Fig. 4 that D%_N are lower in CN speakers than in CSL learners. Results of independent T-test show that D%_N of CN speakers are significantly lower than CSL ($t=3.776$, $p=0.002$, for /an/; $t=4.114$, $p=0.001$, for /an/; $t=2.867$, $p=0.011$, for /en/; $t=2.66$, $p=0.016$, for /en/).

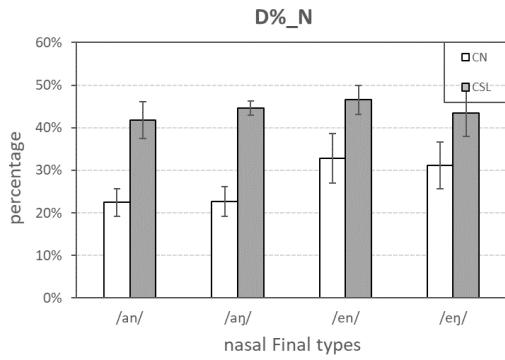


Fig. 4 D%_N of CN and CSL in four nasal Finals.

The results of D%_N and D%_T suggest that the CSL learners over-exaggerate the durations of the nasal coda but not notice to lengthen the durations in transition part which play an important role in natives' production.

2) Formant

Fig. 5 shows that the second formant (F2) trends of the /an/-/an/ pair. For CN speakers, the coarticulation effect of

the nasal coda makes the F2 curve of /an/ show an upward trend, and the F2 curve of /aj/ shows a downward trend. However, as to CSL learners, the F2 curves are almost two parallel straight lines. The F2 trends of /en/-/eŋ/ pair shown in Fig. 6 are identical to /an/-/aŋ/ pair.

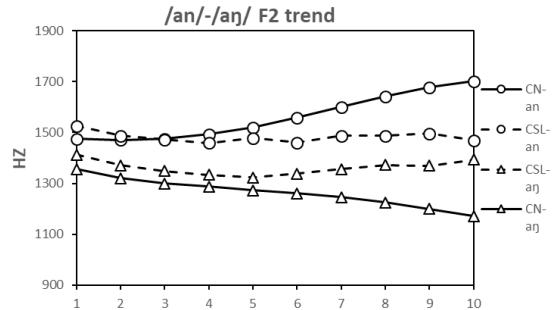


Fig. 5 F2 trend of CN and CSL in /an/-/aŋ/ pair (solid line for CN; dotted line for CSL; circle for /an/; triangle for /aŋ/)

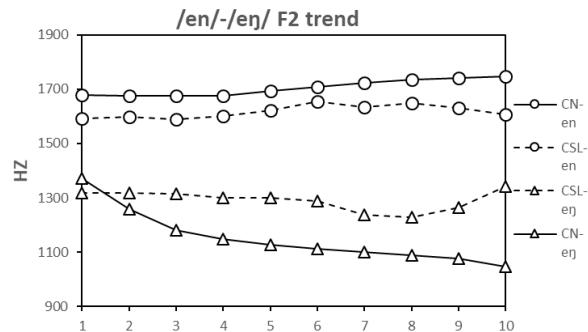


Fig. 6 F2 trend of CN and CSL in /en/-/eŋ/ pair (solid line for CN; dotted line for CSL; circle for /en/; triangle for /eŋ/)

In order to further understand the difference of formant tendency between the second language speaker and the native language speaker, that is, whether the difference of vowel formant curve under different finals is significant, we calculated the slope of formant curve, as shown in Table 2.

TAB. 2 F2 SLOPE OF CN AND CSL

	CN	CSL
/an/	28.41	-1.97
/aŋ/	-18.26***	0.11
/en/	9.31	4.79
/eŋ/	-29.07***	-4.77

Results of independent T-test showed that there was significant difference between the two slopes of F2 formant in CN speakers ($t=12.221$, $p<0.001$, for /an/-/aŋ/ pair) and ($t=10.125$, $p<0.001$, for /en/-/eŋ/ pair). Whereas no significant differences were found between the slopes in CSL ($t=-0.219$, $p=0.829$, for /an/-/aŋ/ pair) and ($t=1.804$, $p=0.092$, for /en/-/eŋ/ pair), which suggests that the coarticulation pattern of L1 and CSL is different. In other words, CSL learners did not attempt to approach CN speakers in coarticulation pattern which is the key acoustic cue for native speakers to identify the coda types.

IV. DISCUSSIONS

The present studies were designed to investigate research questions concerning the effectiveness of the scoring feedback from CAPT system, as well as the correction strategies of the CSL learners in Mandarin nasal Finals by using the learning data.

The results of the effectiveness analysis showed that the scoring feedback could be beneficial to pronunciation training in most cases, but ineffective in nasal Final case. The results of correction strategies analysis showed that firstly, compared to native speakers, CSL learners tended to extend the duration of nasal coda deliberately but bypassed transition part when they are correcting pronunciation errors. Secondly, contrary to native speakers, the slope of F2 between different nasal coda types showed no significant difference in CSL learners. A plausible explanation for these characteristics is that CSL learners are not aware of the true mistakes in pronunciation from the scoring feedback. Therefore, they concentrate more on the nasal coda and subsequently refrain from realizing coarticulation pattern more native-like. The wrong correction strategies lead to a growing deviation between the CSL learners and the native speakers which may explain the decline in assessment scores of CAPT. On the whole, the scoring feedback from CAPT appears to be acceptable currently. For the sounds such as Chinese nasal Finals, more appropriate feedback method is desired in the future.

As the first step of a comprehensive study, we merely analyzed the correction strategies of four basic Chinese nasal Finals. For other learning difficulties, similar methods could be used for reference.

V. CONCLUSIONS

We presented a practicing data analysis on the effectiveness of the CAPT system. The results suggested that the scoring feedback can be beneficial to pronunciation training but invalid for some special cases. Therefore, there is a possibility that the corrective strategies of learners could be blind after receiving fractional feedback. In the future, we will explore a more suitable feedback pattern for CAPT system.

ACKNOWLEDGMENT

This study was supported by Advanced Innovation Center for Language Resource and Intelligence (KYR17005), the Fundamental Research Funds for the Central Universities (16ZDJ03, 18YJ030006, 19YXC111), and the project of "Intelligent Speech technology International Exchange". Jinsong Zhang is the corresponding author.

REFERENCES

- [1] S. M. Witt, "Automatic error detection in pronunciation training: Where we are and where we need to go," Proc. IS ADEPT, 2012.
- [2] O. Engwall and O. Bälter, "Pronunciation feedback from real and virtual language teachers," Journal of Computer Assisted Language Learning, vol. 20(3), pp. 235–262, 2007.
- [3] B. Mak, M. H. Siu, M. Ng, Y. C. Tam, Y. C. Chan, K. W. Chan, K. Y. Leung, S. Ho, F. H. Chong, J. Wong, J. Lo "PLASER:

- Pronunciation Learning via Automatic Speech Recognition” , Proceedings of HLT-NAACL, 2003.
- [4] A. Neri, C. Cucchiari, and H. Strik, “ASR-based correction feedback on pronunciation: does it really work?” in INTERSPEECH, Pittsburg, USA, 2006, pp. 1982–1985.
 - [5] Y. Hirata. “Computer Assisted Pronunciation Training for Native English Speakers Learning Japanese Pitch and Durational Contrasts.” Computer Assisted Language Learning, Vol. 17, No. 3-4, pp. 357-376, 2004.
 - [6] W. Wei, and J. S. Zhang, “An Intelligent Chinese Pronunciation Teaching App and the Preliminary Result of a Teaching Experiment.” Journal of Technology and Chinese Language Teaching, 2019, vol. 9, no. 2, pp: 83-97.
 - [7] H.Q. Bao and M. C. Lin, Essentials of experimental phonetics. Beijing: Beijing University Press, 1994.
 - [8] M. C. Lin and J. Z. Yan, “Coarticulation in the zero-initial syllable with nasal ending in Standard Chinese,” Applied Linguistics, vol. 13, no. 1, pp. 12-20, 1994.
 - [9] Y. J. Wang, “An experimental study on the perception and production of nasal codas by Japanese Learners of Chinese Putonghua,” Chinese Teaching in the World, no. 2, pp. 47-60, 2002.
 - [10] Y. H. Lai, “Production of Mandarin Chinese Nasal Coda by CN and L2 Speakers of Mandarin Chinese,” Journal of Chinese Language Teaching, vol. 5, no. 1, pp. 155-180, 2008.
 - [11] H. Wang, The perception of Japanese learners on Standard Chinese nasal finals. Beijing: Beijing Language and Culture University, 2012.
 - [12] P. Delattre, “Les attributs acoustiques de la nasalité vocalique et consonantique,” *Studia Linguistica*, vol.8, no.2, pp. 103-109, 1954.
 - [13] R. X. Sun, “An Acoustic Study of Nasalized Vowel in Nasal Coda Syllables,” *Journal of Chinese Information Processing*, vol. 29, no. 1, pp. 49-56, 2015.
 - [14] P. Ladefoged, Phonetic data analysis: An introduction to fieldwork and instrumental techniques. Malden, MA: Blackwell, 2003.
 - [15] J. S. Zhang and Z. Y. Wang, “The influence of vowel segments on Japanese learners' perception of Chinese nasal codas,” *Journal of Tsinghua University (Science and Technology)*, vol. 57, no. 2, pp.164-169, 2017.
 - [16] Y. Li, Y. Xie, L. Feng, et al., “The perceptual cues for nasal finals in standard Chinese,” in ISCSLP 2016-10th International Symposium on Chinese Spoken Language Processing, October 17-20, Tianjin, China, Proceedings, 2016, pp.1-5.
 - [17] X. J. Luo, et al. “Coda's duration on perception of mandarin syllables with alveolar/velar nasal endings by Japanese CSL learners.” 2015 International Conference Oriental COCOSDA held jointly with 2015 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE), IEEE, 2015.
 - [18] B. M. Lobanov, “Classification of Russian vowels spoken by different speakers,” *The Journal of the Acoustical Society of America*, vol. 24, no. 2, pp. 175-184, 1952.

Examination-Style Reading Comprehension with Neural augmented Retrieval

Yiqing Zhang^{1,2,3}, HaiZhao^{1,2,3,*}, ZhuoshengZhang^{1,2,3}

¹Department of Computer Science and Engineering, Shanghai Jiao Tong University

²Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, China

³MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
zhangyiqing, zhangzs@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn

Abstract—In this paper, we focus on an examination-style reading comprehension task which requires a multiple choice question solving but without a pre-given document that is supposed to contain direct evidences for answering the question. Unlike the common machine reading comprehension tasks, the concerned task requires a deep understanding into the detail-rich and semantically complex question. Such a reading comprehension task can be considered as a variant of early deep question-answering. We propose a hybrid solution to solve the problem. First, an attentive neural network to obtain the keywords in question. Then a retrieval based model is used to retrieve relative evidence in knowledge sources with the importance score of each word. The final choice is made by considering both question and evidence. Our experimental results show that our system gives state-of-the-art performance on Chinese benchmarks and shows its effectiveness on English dataset only using unstructured knowledge source.

Keywords-MRC; retrieval; knowledge source;

I. INTRODUCTION

¹ Machine reading comprehension (MRC) which requires computers to answer questions based on acquired knowledge, is regarded as the milestone for deep natural language understanding. A great amount of datasets have been released, including cloze-style CNN/Daily Mail [1], multiply choice type QuaRel [2] and user-query type SQuAD [3], MS MARCO [4]. Most MRC datasets provide one or several corresponding documents which are commonly retrieved by search engines. However, MRC systems may not always find available pre-given document as the hypothesis like all previous cloze-style or user-query types tasks, they have to be both capable of discovering the relationship between question and answer and retrieve critical evidence even without the pre-given documents.

In this work, we thus consider such an examination-style MRC task, which releases the inconvenient requirement about the pre-given document commonly in other types of MRC tasks. Undoubtedly, training machines to take human examinations is more challenging in which answers have to be given on a basis without a standard relevant document that is supposed to contain cues for answering the question.

Why the concerned examination-style MRC task is more challenging than those common MRC types is that

^{*} Corresponding author. This paper was partially supported by National Key Research and Development Program of China (No. 2017YFB0304100) and Key Projects of National Natural Science Foundation of China (No. U1836222 and No. 61733011).

Table I
AN EXAMPLE OF QUESTION TYPE WE FOCUS ON. THERE IS NO CORRESPONDING DOCUMENT PRE-GIVEN.

Question: Some Western scholars think ‘Half of all the inventions and discoveries that the modern world depends on may be from China’. The science and technology achievement of Song Dynasty that was introduced to Europe and had a profound impact on the modern world is ?

Candidate answers:

- | | |
|------------------------|----------------------|
| A. Chang’s seismograph | B. Papermaking skill |
| C. Typography | D. Compass |

Correct answer: D

evidence knowledge for answering the questions should be found by question-answering (QA) system itself, which makes the solution model much more complicated. The concerned MRC task can be also regarded as a type of deep QA [5], [6] one considering that no standard pre-given document is given for question extraction.

However, in detail, this paper considers solving a subtype of the concerned MRC task, *multiple choice questions*. It slightly differs from common deep QA task that answer candidates have been given in a selectable list [7]. Table I shows a typical multiple choice question in history exam which consists of a question with some background information and four candidate answers. Our system follows the way how human solve this problem. First, we train an attentive neural network as the basic MRC model. During this step, the neural network can learn key words for searching evidence knowledge by attention mechanism, because the important words for choosing the right answer is always the important words for searching the relevant evidence knowledge. Then the importance of the key words and the question are taken into consideration when the retrieval model searches evidence knowledge from knowledge source. Finally, both question and evidence are used to determine the right answer. Experimental results show that, with the neural model indicating word importance, the retrieval model can find more accurate evidence to help judge which candidate is the correct answer. In Chinese dataset the relevant evidence is not tagged, the effectiveness of word importance can only be reflected by the final accuracy of choosing the right answer, so we use an English multiple choice question-answering dataset [8]. In this dataset, paragraph which contains information to answer the certain question is

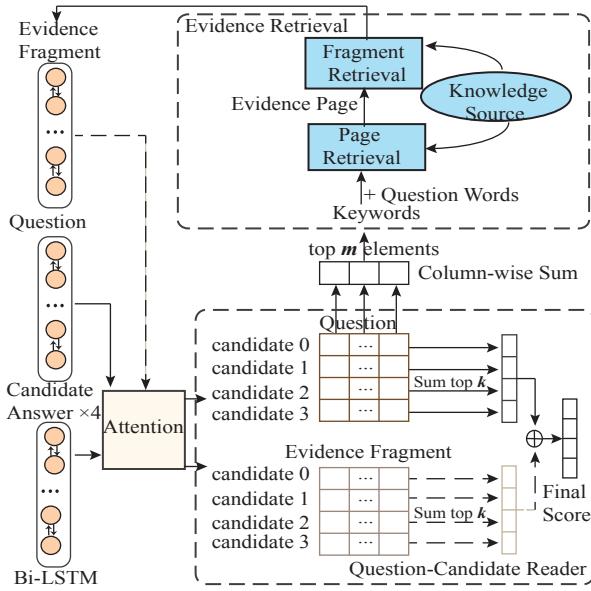


Figure 1. An overview of our system. The data flow denoted by dotted arrow appears after Evidence Retrieval module generates output.

indeed given, we tag the given paragraph as the best paragraph and all the paragraphs are collected to be the external knowledge source. After that we can directly know the effectiveness of the neural model indicating word importance to the retrieval model. The code of our system is available here².

Our contribution in this paper is three-fold:

- We formally focus on examination-style MRC task that the evidence document is not given and needed to be found by QA system itself.
- We use unstructured knowledge source which makes our system can be easily improved by using high-quality knowledge source in specific areas.
- We design an MRC system that combines the retrieval based and neural based method in a novel way for multiple choice question-answering.

II. RELATED WORK

In recent years, an ambitious AI task that develops QA system to pass examinations in different levels and subjects has been paid a lot of attention with the development of deep learning [9]. The Project Aristo [10] devoted to having the computer pass Elementary School Science and Math exams. Another more ambitious project is the Todai Robot Project [11] which aims to let the computer take the university admission examination in Japan and reach the acceptance score of University of Tokyo.

During the development of MRC, different types of datasets have been released. The SQuAD [3] poses a question-answering mode over the pre-given document, in which the question is related to the document and the answer is a span in the document. MS-MARCO [4] provides several documents for each question and the question

should be answered based on these documents. RACE [8] is from real-word examination with the same QA mode as SQuAD except for questions in multiple choice form. Anyway, it is generally accepted that providing a pre-given document as the above MRC task surely alleviates the difficulty of the task.

For a more challenging task without such a pre-given document, external knowledge source has to be taken into account. Knowledge base (KB) can be processed easily by computer, however, it has the inherent limitation of being hard to build. Researchers thus consider KB extension or enhancement. Berant [12] scaled up Freebase by learning from question-answer pairs. However, even the largest KB still suffers from the insufficiency, so unstructured resources like Wikipedia have been widely used in QA systems [13].

Recently, a few works considered using more diverse sources other than Wikipedia. For example, Sachan [14] used instructional materials to tackle science question. Zhang [15] studied answering questions in clinical medicine using knowledge in a large-scale document collection. Guo [16] solved multiple choice questions in history exam by using textbooks resources and relevant information collected from Baidu Encyclopedia. Zhang [17] recently adopted strong machine learning model like oneshot learning to overcome the data deficiency difficulty in examination-style MRC.

Different from all the mentioned work that either seeks specific support data sources or strengthens the model itself, this work adopts a hybrid mechanism by effectively integrating retrieval model and neural models instead for the concerned examination-style MRC task. We regard the DrQA system [18] is the most similar one to this work for both using a retrieval module inside. However, this work significantly differs from [18] from the following factors. First, this work is about strict examination-style MRC task, while DrQA actually still works for a SQuAD-like span MRC task but simply assuming that the pre-given document is missing. Second, this work uses neural module to feed retrieval module for the query and the final solution to the question is determined jointly by both retrieval and neural modules, while DrQA uses a simple retrieval module to return a Wikipedia page as the pre-given document, and then span MRC system turns on for all the rest processing.

III. SYSTEM

Each multiple choice question can be regarded as a triple (Q, C_s, A) where Q is the question, C_s is a set of four candidate answers and A is the correct answer. Figure 1 is an overview of our system.

A. Question-Candidate Reader

Our Question-Candidate Reader follows the backbone of the AOA model [19] which has proved its good performance. We keep the attention mechanism of it and change its embedding layer and answer prediction layer in order to make the model adapt to our concerned task. The following gives more model details.

²<https://github.com/Yiqingss/Gaokao-history-QA>

Contextual Embedding For English dataset, we use GloVe [20] word embedding. For Chinese datasets, we use word2vec [21] to pretrain word embeddings on Chinese Wikipedia. Then we use 2 Bi-LSTM [22] to get contextual representations $h(x)$ of the question and candidate answers.

Attention The attention score follows the AOA model [19] which is given by

$$\begin{aligned} M(i, j) &= h_{question}(i)^T \cdot h_{candidate}(j) \\ \alpha &= softmax_{column}(M), \beta = softmax_{row}(M) \\ \beta_{avg} &= \frac{1}{|Q|} \sum_{t=1}^{|Q|} \beta(t :) \\ S &= \alpha \cdot \beta_{avg} \end{aligned} \quad (1)$$

where $M(i, j) \in \mathbb{R}^{|Q|*|C|}$ is the matching matrix of each question-candidate pair, as each question has four candidate answers, we get four matrices for each question. The value of i -th row and j -th column of M is filled by the dot product of i -th word in question and j -th word in candidate answer. Then a column-wise and a row-wise softmax function is applied to the matching matrix M to get its question-level attention ($\alpha \in \mathbb{R}^{|Q|*|C|}$) and candidate-level attention ($\beta \in \mathbb{R}^{|Q|*|C|}$) respectively. β_{avg} is the result of column-wise average of β , $\beta(t :)$ is t -th row vector of β . $S \in \mathbb{R}^{|Q|}$ is the attention-over-attention score for each question-candidate pair.

Answer Prediction As each question has four candidate answers, each question will get four S . We denote the attention score S calculated by Eq. (1) with the i -th candidate answer as $S_i, i \in \{0, 1, 2, 3\}$. Then we compute a final score for each question-candidate pair by summing the value of elements in S_i with high score.

$$P(C_i|Q) = \sum top(k; S_i), i \in \{0, 1, 2, 3\} \quad (2)$$

where $top(k; S_i)$ means the k elements with the highest scores in S_i , $P(C_i|Q)$ is the sum of the value of these elements. The predicted answer is candidate answer with the highest $P(C_i|Q)$ score. K is set to 4 through empirical tuning. The words corresponding to the high-score elements make up the set of importance words which will be used in Evidence Retriever.

Training Giving a training corpus of multiple choice question: $U = \{(Q^{(i)}, C_s^{(i)}, A^{(i)})\}_{i=1}^{|U|}$, the training objective of our model is to minimize the cross entropy of the training data.

$$\begin{aligned} A_j^{(i)} &= \begin{cases} 1, & C_s_j^{(i)} = A^{(i)} \\ 0, & C_s_j^{(i)} \neq A^{(i)} \end{cases} \\ \mathcal{L} &= - \sum_{i=1}^{|U|} \sum_{j=0}^3 [A_j^{(i)} \cdot \log(\tilde{A}_j^{(i)}) \\ &\quad + (1 - A_j^{(i)}) \cdot \log(1 - \tilde{A}_j^{(i)})] \end{aligned}$$

where the \tilde{A}_j is the predict likelihood of the j -th candidate answer computed by applying softmax on four candidates' prediction scores $P(C_j|Q)$.

B. Evidence Retriever

Given a question Q consisting of l tokens $\{q_1, q_2, \dots, q_l\}$ and a knowledge source page set P with n pages $\{p_1, p_2, \dots, p_n\}$. The Evidence Retriever returns a fragment of a knowledge source, because we found that in most of the unstructured knowledge sources, even in one page there are still redundant information.

Page Retrieval We first calculate each word's TF-IDF score $S_{tf-idf}(w)$ in P :

$$S_{tf-idf}(w) = \log[tf(w, p_i) + 1] * \log \frac{vocab - idf(w) + 0.5}{idf(w) + 0.5}$$

where $tf(w, p_i)$ is the number of times that word w occurs in p_i , $idf(w)$ is the number of knowledge source pages containing word w and $vocab$ is the number of unique words in all knowledge source pages.

We then compute each word's importance score and choose m words with highest scores as keywords. We use a softmax function to get probability distributions of the keywords. The set of keywords is denoted as K consisting of m tokens $\{k_1, k_2, \dots, k_m\}$.

$$S_{imp}(j) = \sum_{i=0}^3 S_{ij}, w_j \in Q \quad (3)$$

$$P(k_i) = \frac{e^{S_{key}(k_i)}}{\sum_{j=1}^m e^{S_{key}(k_j)}}, k_i \in K$$

$$weight(w) = \begin{cases} \log_{0.9}(1 - P(w)), w \in K \\ 1, w \notin K \end{cases} \quad (4)$$

where S_{ij} is attention score of the j -th word in question computed with the i -th candidate. $S_{imp}(j)$ is the importance score of the j -th word in question which is the sum of j -th word's attention scores. $S_{key}(k_i)$ is the importance score of keyword k_i and its value is computed by Eq. (3). $weight(w)$ is the importance degree.

Finally, we compute the relevant score of every paragraph by

$$S_{para}(P, Q) = \sum_{w \in \{P \cap Q\}} [a * S_{tf-idf}(w) * weight(w)] + |P \cap Q|$$

$|Q \cap P|$ is the number of words both in Q and P , and a is a parameter to be tuned. We choose the page which has the highest $S_{para}(P, Q)$. Then this page will be processed by the following module.

Fragment Retrieval: In this module, we use a sliding window to find the most relevant fragment. Relevant score of each fragment selected by the sliding window is computed by

$$S_{frag}(F, Q) = \sum_{w \in \{F \cap Q\}} weight(w)$$

where F is a continuous sequence with a fixed length, $weight$ is computed by Eq. (4). The final result is the fragment with the highest score.

Table II
THE STATISTICS OF THE GAOKAO DATASETS

	Gaokao-577 (Test)	Gaokao-744 (Test)	TIKU (Train)
Count	577	744	53709

Table III
THE STATISTICS OF THE MODIFIED RACE

	Dev middle	Dev high	Test middle	Test high
#passage set	1,436	3,451	1,436	3,498

C. Final Prediction

The final prediction combines the results of Question-Candidate Reader and Evidence Retriever. To use the fragment returned by Evidence Retriever, we replace the Q in (Q, Cs, A) mentioned before with F , which is the fragment of Q found by Evidence Retriever. Then we get the triple (F, Cs, A) . The final prediction score is computed by

$$P(C_i|Q, F) = P(C_i|Q) + b \cdot P(C_i|F)$$

where $P(C_i|Q)$ and $P(C_i|F)$ is calculated by Eq. (2) with triple (Q, Cs, A) and (F, Cs, A) respectively and b is a hyperparameter to be tuned.

IV. EXPERIMENTS

A. Dataset

We evaluate the proposed model in both Chinese and English datasets.

Chinese: Gaokao dataset We adopt two published datasets whose question-candidates pairs are collected from Gaokao (Chinese College Entrance Examinations). The first one³ is published by Cheng [23], containing 577 multiple choice questions. This dataset is referred to Gaokao-577. The second one is published by Guo [16], containing 2 question sets. One is 744 questions from Gaokao, and the other is 53,709 practice questions⁴ from TIKU. We refer these two question sets to Gaokao-744 and TIKU, respectively. All the questions in TIKU are used as the training data. Each question in Gaokao-744 has been manually divided into entity questions (EQs) and sentence questions (SQs). Entity questions are those whose candidates are all entities, and sentence questions are those whose candidates are all sentences. The statistics of these datasets are listed in Table II. During our experiments, we found that most of the question-candidates pairs in Gaokao-744 appears in TIKU and some question-candidates pairs are repeated several times. So we removed TIKU's questions of which 90% words are the same as a certain question in Gaokao-744, as a result, about 1,000 question-candidates pairs are removed and 10% of the rest questions are used as development data.

³<http://ws.nju.edu.cn/gaokao/ijcai-16/GaokaoHistory577.xml>

⁴<https://github.com/IACASNLPIR/GKHMC/tree/master/data>

The external knowledge resource we used as knowledge source for Gaokao-577 and Gaokao-744 is the contents of Chinese Wikipedia pages (dump on 2017/11/14) with a total of 299,383 pages. The latest version for Chinese Wikipedia pages is available here⁵.

English: RACE RACE [8] (Reading Comprehension Dataset From Examinations) consists of 27,933 passages and 97,867 questions collected from English exams for middle and high school Chinese students. RACE is divided into two subsets RACE-M (from middle school exams) and RACE-H (from high school exams). Each question in RACE is followed with 4 candidate answers and only one of them is correct.

We choose RACE because it is a multiple choice style question answering dataset which is similar to the Gaokao datasets. The main difference between RACE and Gaokao multiple choice questions is that each question-candidates pair in RACE has a passage which contains the evidence for choosing the correct answer. This difference happens to cover the shortage of the Gaokao datasets that they do not have tagged evidence to directly evaluate the retrieval module's performance. We change the dataset to fit it to our system. We collect all the passages to build a knowledge source and the pre-given passage for a certain question is tagged as the best passage for the question, so that we can directly evaluate the retrieval module's performance. Table III lists the statistics of RACE dataset with our modification. The *#passage set* means how many passages a knowledge source contains in a sub-dataset

B. Reader Evaluation

Settings We use 1-layer bidirectional LSTM with 400 hidden units for both question and candidate answers encoding. For Gaokao-577 and Gaokao-744, We use pre-trained word embeddings by *word2vec* [24] from 903M Wikipedia data in Chinese, and for RACE, we use *GloVe* [20]. The dimensions of embedding for all the datasets is 300. Our model is implemented using Tensorflow. ADAM optimizer is adopted for weight updating with a declining learning rate from 0.001 to 0.00005 during training and dropout rate of 0.9. A gradient clipping threshold [25] is applied to avoid gradient explosions. All the training samples are divided into batches of 32 samples each. For Chinese datasets, sentences need to be segmented into words firstly [26].

Effectiveness of top k As the answer prediction of Question-Candidate Reader is up to the sum of the k largest element in Eq. (1), the value of k must have a great impact to the final result. We test the accuracy of the Question-Candidate Reader with k varying from 1 to 6 on both Gaokao-577, Gaokao-744, and the result is showed in Figure 2, from which, Question-Candidate reader achieves best performance on Gaokao-577 when $k = 4$, on Gaokao-744 when $k = 5$.

Figure 2. Test accuracy against the value of K.

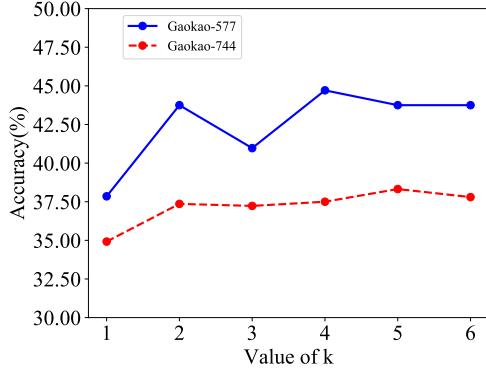


Table IV
ACCURACIES ON GAOKAO-577 (%)

Gaokao-577	
Random	25.00
[23]	33.79
Our system (Q)	44.71
Our system (F)	30.03
Our system (Q+F)	45.41

C. System Evaluation

We examine the performance of our system with question only, evidence fragment only and combination of question and evidence fragment on Gaokao-577. The evaluation on Gaokao-744 is absent because the model in [16] used additional textbooks and resources other than Wikipedia as the external knowledge source. As the Wikipedia provides a great deal of informative clues, it makes incomparable results.

As we can see in Table IV, our system outperforms those systems that use Wikipedia as unique external knowledge on Gaokao datasets by a large margin, where 11.62% absolute improvements over [23] in Gaokao-577. DrQA does not perform well here, which is due to the relatively much more poor quality of Chinese Wikipedia pages. It can be also found that the system can achieve a relatively high accuracy only taking the question into consideration. The accuracy (30.03%) is low when we only use the evidence fragment on Gaokao datasets.

We also examine our best-performing system on Gaokao-744, in order to figure out our system performance on EQs and SQs, and the result of it is showed in Table V. We can see that our system performs better on SQs than EQs which proves that our system has a better ability to detect the relationships between sentences.

D. Evidence Retriever Evaluation

Effectiveness of key words and Fragment

To have a thorough investigation in the effectiveness of key words and their weighted scores given by Question-Candidate Reader, we first get the evidence fragment with n key words, $n \in \{0, 1, 2, 3, 4, 5, 6\}$, and then compute the accuracy of Question-Candidate reader only with the

⁵<https://dumps.wikimedia.org/zhwiki/latest/>

Table V
ACCURACIES ON GAOKAO-744 (%)

Count	entity question	sentence question
	160	584
[16]	45.63	45.72
Our system	31.25	39.04

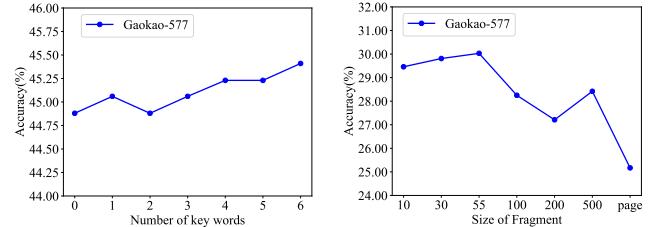


Figure 4. Test accuracy against the number of key words on Gaokao-577

Figure 5. Test accuracy of different sizes of fragment on Gaokao-577. Fragment size is measured by the number of Chinese characters inside and ‘page’ indicates the entire page used as the evidence fragment.

fragment. $n = 0$ means that we do not use the attention result of Question-Candidate Reader, and every words’ weight is 1. $n = 1$ means only one key word is taken and its weight will be set to 15, as Eq. (4) can not calculate the weight at this situation. The results on Gaokao-577 are showed in Figures 4.

We see that the highest accuracy appears when $n = 6$ and the accuracy is low when $n = 0$, which proves that searching evidence without considering the importance of words performs poorly.

Different from the previous work [23], we do not use the whole page as evidence information to judge the possibility of a candidate answer. Instead, we use a slide window to pick a continuous sequence with a fixed length in the found page. To investigate whether this method works, we examine the best-performing system with the different fragment sizes, and the result is showed in Figures 5. The performance is poor when the fragment size is page size on Gaokao-577, this shows that using the entire page cannot bring about a better performance because of the redundant information in the entire page. Our system actually achieves the best performance when the fragment size is set to 55 on Gaokao-577.

Accuracy of retrieval results on modified RACE

After modifying the RACE dataset, we can directly figure out the accuracy of retrieval module. We build the TF-IDF index with the passages in the training data of RACE and the best result is achieved when the number of keywords is set to 5. As is showed in Table VI, the accuracy increases after adding the keywords’ weight. During our experiment we found that the number of keywords do not make much influence on the accuracy, it is mainly because the questions of RACE are short and usually do not contain information to choose the right answer, which makes the neural model hard to learn the attention weight for each question.

Table VI
RETRIEVAL ACCURACY ON RACE (%)

	dev middle	dev high	test middle	test high
TF-IDF	0.533	0.390	0.487	0.406
TF-IDF + keywords	0.544	0.422	0.501	0.429

V. CONCLUSION

This work focuses on a challenging question answering type in real-word examination which does not have a pre-given document for answering the question. We proposed a system that uses external knowledge sources to help choose the right answer, and our system integrates neural model with attention and retrieval based model in a novel way. The effectiveness of the proposed approach has been verified on benchmark datasets. Only using Wikipedia as the unique knowledge source, our system outperforms previous state-of-the-art systems.

REFERENCES

- [1] K. M. Hermann, T. Kociský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, “Teaching machines to read and comprehend,” in *NIPS 2015*, 2015, pp. 1693–1701.
- [2] O. Tafjord, P. Clark, M. Gardner, W. Yih, and A. Sabharwal, “Quarel: A dataset and models for answering questions about qualitative relationships,” *CoRR*, vol. abs/1811.08048, 2018.
- [3] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “SQuAD: 100,000+ questions for machine comprehension of text,” in *EMNLP 2016*, 2016, pp. 2383–2392.
- [4] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng, “MS MARCO: A human generated machine reading comprehension dataset,” in *NIPS 2016*, vol. 1773, 2016.
- [5] D. A. Ferrucci, E. W. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. M. Prager, N. Schlaefler, and C. A. Welty, “Building watson: An overview of the deepqa project,” *AI Magazine*, vol. 31, no. 3, pp. 59–79, 2010.
- [6] A. Lally, S. Bagchi, M. Barborak, D. W. Buchanan, J. Chu-Carroll, D. A. Ferrucci, M. R. Glass, A. Kalyanpur, E. T. Mueller, J. W. Murdock, S. Patwardhan, and J. M. Prager, “Watsonpaths: Scenario-based question answering and inference over unstructured information,” *AI Magazine*, vol. 38, no. 2, pp. 59–76, 2017.
- [7] S. Zhang, H. Zhao, Y. Wu, Z. Zhang, X. Zhou, and X. Zhou, “Dual co-matching network for multi-choice reading comprehension,” *CoRR*, vol. abs/1901.09381, 2019. [Online]. Available: <http://arxiv.org/abs/1901.09381>
- [8] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. H. Hovy, “RACE: large-scale reading comprehension dataset from examinations,” in *EMNLP 2017*, 2017, pp. 785–794.
- [9] S. He, Z. Li, H. Zhao, and H. Bai, “Syntax for semantic role labeling, to be, or not to be,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, 2018, pp. 2061–2071.
- [10] P. Clark, “Elementary school science and math tests as a driver for AI: Take the Aristo challenge!” in *IAAI 2015*, 2015, pp. 4019–4021.
- [11] A. Fujita, A. Kameda, K. Ai, and Y. Miyao, “Overview of Todai Robot Project and evaluation framework of its NLP-based problem solving,” in *IREC 2014*, 2014, pp. 2590–2597.
- [12] J. Berant, A. Chou, R. Frostig, and P. Liang, “Semantic parsing on Freebase from question-answer pairs,” in *EMNLP 2013*, 01 2013, pp. 1533–1544.
- [13] D. Ahn, V. Jijkoun, G. Mishne, K. Müller, M. de Rijke, and S. Schlobach, “Using wikipedia at the TREC QA track,” in *TREC 2004, November 16-19*, 2004.
- [14] M. Sachan, A. Dubey, and E. P. Xing, “Science question answering using instructional materials,” in *ACL 2016*, 2016, pp. 467–473.
- [15] X. Zhang, J. Wu, Z. He, X. Liu, and Y. Su, “Medical exam question answering with large-scale reading comprehension,” in *AAAI 2008*, 2018, pp. 5706–5713.
- [16] S. Guo, X. Zeng, S. He, K. Liu, and J. Zhao, “Which is the effective way for gaokao: Information retrieval or neural networks?” in *EACL 2017*, 2017, pp. 111–120.
- [17] Z. Zhang and H. Zhao, “One-shot learning for question-answering in gaokao history challenge,” *CoRR*, vol. abs/1806.09105, 2018.
- [18] D. Chen, A. Fisch, J. Weston, and A. Bordes, “Reading Wikipedia to answer open-domain questions,” in *ACL 2017*, 2017, pp. 1870–1879.
- [19] Y. Cui, Z. Chen, S. Wei, S. Wang, T. Liu, and G. Hu, “Attention-over-attention neural networks for reading comprehension,” in *ACL 2017*, 2017, pp. 593–602.
- [20] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *EMNLP 2014*, October 25-29, 2014, pp. 1532–1543.
- [21] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *CoRR*, vol. abs/1301.3781, 2013.
- [22] M. Sundermeyer, R. Schlter, and H. Ney, “LSTM neural networks for language modeling,” in *ICA 2012*, 2012, pp. 194–197.
- [23] G. Cheng, W. Zhu, Z. Wang, J. Chen, and Y. Qu, “Taking up the gaokao challenge: an information retrieval approach,” in *IJCAI 2016*, 2016, pp. 2479–2485.
- [24] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *NIPS 2013*, vol. 26, 2013, pp. 3111–3119.
- [25] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in *ICML 2013*, 2013, pp. III-1310.
- [26] H. Zhao and C. Kit, “Integrating unsupervised and supervised word segmentation: The role of goodness measures,” *Inf. Sci.*, vol. 181, no. 1, pp. 163–183, 2011.

Automatic Extraction and Quantitative Evaluation of the Character Relationship Networks from Children's Literature works

Kun Ma*

Institute of Chinese information processing
Beijing Normal University
Beijing, China
*e-mail: 201821090021@mail.bnu.edu.cn

Abstract—To automate the graded reading task, we urgently need to extract and calculate the important index of the complexity of the relationship between the characters affecting the plot complexity of narrative literature. In order to realize this purpose, this paper describes a computational method for automatic analysis of the virtual social network from children's literature works. We selected the required bibliography for primary school students recommended by the Ministry of Education, then automatically extract the characters of the novel by CRF, and constructs the character network based on the co-occurrence relationship. The statistical analysis method of complex network provides a quantitative basis for distinguishing the complexity of characters' relationships in different texts. The results show that the structural characteristics of character interaction networks are similar to those of small world networks, and the selected network measurement indexes are significantly related to the complexity of text characters. Finally, we achieved effectively evaluating and predicting the complexity of the social networks from more extensive literature works some classical regression model based on machine learning.

Keywords-Graded reading; Automatic analysis of text; Children's Literature works; machine learning; complex network

I. INTRODUCTION

The traditional vocabulary reading grading method and the evaluation of text difficulty rely too much on the evaluator's intensive reading of literary texts, and the subjective experience of literary works characters, so the efficiency is relatively low and has a strong subjectivity. In recent years data-driven analysis has emerged as a growing methodology within literary studies.

Twenty years ago, an article on the Small World Network was published in Nature and an article on scale-free networks was published in Science. Most network science researchers regard these two articles as a sign of the rise of network science. With the deepening of the theory of complex network, networks are currently being studied across many fields of science, such as Internet, social networks and even literary. The cognitive science researcher, R.Andeson shows that although the reader finally obtains the content schema by means of language schema and formal schema, once the reading is over, the deepest trace in memory is the content schema. This conclusion coincides with the graphical representation of complex networks. This paper hopes to go further on the basis of predecessors and try to use the

Lijiao Yang[§]

Institute of Chinese information processing
Beijing Normal University
Beijing, China
§ e-mail: yanglijiao@bnu.edu.cn

combination of various theoretical methods to realize the purpose of automatic extraction and complexity evaluation of character interaction network of children's literature works.

This paper defines text difficulty as the subjective and objective evaluation of the difficulty of text, which combines all the quantifiable factors that affect reading difficulty. It is usually used to evaluate and sort the difficulty of reading materials. In addition to the lexical level and sentence level which are most concerned with by existing graded reading system, the textual dimension of the entire book is undoubtedly an important factor. The character relationship of narrative texts is one of the important indicators which has a significant impact on readers, especially children with poor reading ability. But even the maturest graded reading systems in foreign countries, such as: Lexile Measure, GEL (Guided Reading Level), Accelerated Reader, only achieve automatic scoring of text difficulty at the level of words and segments. The evaluation of textual level is still scored by experts. So the automatic extraction and calculation of the important indicators of the relationship complexity of characters is the one of the innovations of this paper.

Therefore, basing on the theory of complex networks, this paper is oriented to the reality of graded reading tasks and needs with the help of more mature tools and partial self-programming of machine learning. We constructed networks of characters based on co-occurrence relations for 100 narrative children's literature texts. Based on the statistical analysis method of complex network, the texts are calculated and analyzed, trying to find the network measurement index that can automatically predict the complexity of the relationships between the characters. Finally, the extraction and investigation of the relationship networks in this paper confirms the high correlation between the selected networks measurement indicators and the complexity of the textual characters through the correlation test. Finally, the classic regression algorithm of machine learning was adopted. It effectively realized the effective prediction of the complexity of narrative text characters.

Now, this article will discuss the four aspects of related work, experimental process, experimental results and analysis, conclusions and follow-up work.

II. RELATED WORKS

As early as 2002, Moretti plotted the character relationship networks of "Hamlet", "Our Mutual Friend" and "The Story

[§] Corresponding author: yanglijiao@bnu.edu.cn

of the Stone" and performed a deep analysis of the plot structures (2005). Then, Sparavigna applied the method of Moretti's to plot the character relationship networks of "Harry Potter". David K. Elson and others focused on 19th-century English novels and serials, extracting networks from dialogue relationships, and drew preliminary conclusions on the simultaneous growth of social network cohesiveness, interconnectivity. Rydberg-Cox created an application to visualize and explore social networks of Greek tragedies(2011). Sparavigna and Marazzato applied the Graph Visualization Software to visual the character networks of two Shakespeare's play (2014).

With the introduction of this method, domestic scholars have gradually started to carry out relevant experiments and research. Some traditional literary researchers draw on the research methods and experiences of foreign scholars to conduct similar human relationship extraction and visual representation for a few Chinese novels, and to explore the relationship between the characters, the centrality of the characters, and so on. For example, Tang Yi studied the 108 main characters in The Water Margin. Zhu Haijun and others built the character relationship network in the TV series "Bailu Village". Liu Haiyan conducted a comparative study of Fitzgerald's four novels. The topological measurement of the four networks confirmed that they have the characteristics of "small world". Chen Bikun and others used the "Journey to the West Prequel" as a corpus to calculate and visualize the character network. Chen Lei, Hu Yiqi and others explored the social hierarchical relationship between the pairs of people who frequently co-occurred in "The Story of the Stone". In addition, some information technology researchers focused on the implementation path of automatic extraction of character relationships. The computational models used mainly include word vectors and two-way GRU nerves Network, convolutional neural network, etc.

In general, the realization of the above research and the conclusion of the preliminary conclusions have proved the similarity between the virtual world of the text and the real world to a certain degree. At the same time, it also verifies that using complex network measures to statistically analyze relationship between the characters is a completely feasible and reliable research method. But we can also see that there are obvious deficiencies in the existing research. First, the research on existing literary texts mostly focused on the network construction and text content mining of individual works, individual series of texts or individual writers. The number of texts is very limited, lacking horizontal comparison of different texts makes it difficult to get more general conclusions. Second, the use of topological measure indicators for the relationship network is more based on the qualitative research and does not have more effective application and mining of the network measurement. In addition, the relevant research in China is mostly empirical research, trying to provide a quantitative basis for the literary criticism of the novel, and thus lacks specific scenarios and fields for future applications.

III. EXPERIMENTAL PROCESS

A. Data And Pre-Processing

This article selects the narrative texts recommended by the Ministry of Education in 2018 and uses the children's literature works recommended by Beijing's outstanding teachers as supplements. A total of 100 narrative children's literature texts are collected and cleaned, such as "Little Pig Lili Lulu", "Alice in Wonderland", "Peter Pan", "Xiao Bing Zhang Ga", "Niels riding a goose travel", etc., adding up to more than 160,000 words. Since the purpose of this paper is to evaluate the complexity of the relationship between the characters, the texts selected should have obvious differences in the complexity of the relationship of the characters. There should also be more obvious differences in indicators. The selected texts are graded by three linguistic graduate students and two associate professors based on the complexity of the relationship between the characters, with a value of 1-15. The average score of the five scores obtained from the text is used as the difficulty marker. In this paper, each literary text is divided into words by jieba tokenizer, and the total number of words in each paper is counted.

B. Extracting Conversational Networks from Literature

1) Character Identification

Character recognition and marking tasks are one of the research hotspots and emerging development directions. The name is part of the named entity. Most of the current mainstream methods are based on machine learning models. This method learns the annotation corpus and implements unregistered words and person name recognition in the form of sequence annotation. This paper uses conditional random field CRF (Conditional Random Fields) to solve the problem. By manually defining the feature template, the contextual annotation information is used to explore the named entity. The article uses the open source tool CRF++, which combines the 50,000-word "People's Daily" corpus with 50 children's literary texts collected in this paper to complete the word segmentation, part-of-speech tagging and name tagging, forming a training corpus. The non-named entity N, and the required character word entity was labeled Per, as shown:

TABLE I. THE LABELS OF ONE SENTENCE

THE LABELS OF ONE SENTENCE											
孔子	称	子产	是	对	人	有	惠爱	的	人	.	
Nr	v	N	v	p	n	v	n	uj	N	x	
Per	N	Per	N	N	N	N	N	N	N	N	

Using this as a training set, the model is trained using the characteristics of the word itself, the part of speech, the position of the word. Then the name recognition was performed for the remaining half of the text to extract all the names of the remaining literary texts and save them as a character dictionary.

In addition, the name of the person should be reviewed. First, the characters with the word frequency below 2 should be excluded to avoid excessive non-important characters

from interfering with the network construction. Based on these work, this article carried out the list of names. In a continuous narrative text, the person entities and the referents do not necessarily correspond one-to-one. This requires us to identify different references of the same entity and to make them consistent. Therefore, this article asks five Chinese students to participate in the proofreading work on the same name of the text person, and assign different referential methods to the same person. For example, "Confucius" is replaced by "Confucius" and "Zhongni" is replaced by "Confucius". To ensure that the name of the person is not redundant or repeated, this is a prerequisite for ensuring accurate identification of the relationship between the characters.

2) Constructing social networks

In the predecessors' research, the character relationship network of literary texts is mostly based on the co-occurrence of characters and the dialogue of characters. This paper combines two methods to place independent paragraphs of consecutive dialogues in the same paragraph, and then adopt the method of judging the co-occurrence of characters. That is, the paragraph newline is used as the criterion, and the characters appearing in the same paragraph are recorded as co-occurrence.

The co-occurrence relationship of two characters is regarded as one side of the relationship network. We extracted the relationship of the characters based on the name dictionary, constructed the adjacency matrix representing the co-occurrence relationship between the characters, and transformed it into a network model for visualization by the self-editing program. The construction tool used in this paper is networkx. The existing person name dictionary is used as point data, and the character relationship information is passed as side data to construct and output the network model.

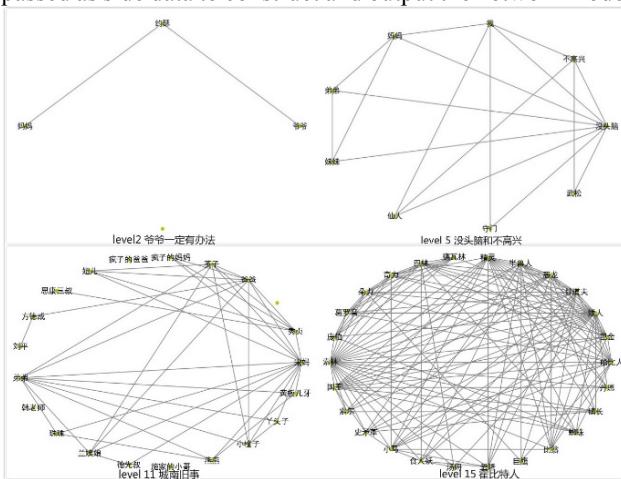


Figure 1. Examples of the networks in different levels.

In this paper, data analysis and simulation modeling of complex networks are performed on the generated 100 networks, using its built-in graph and complex network

analysis algorithm. Then, the indicators are extracted and statistically analyzed.

C. Topological Measurement Calculation of Virtual Social Network

After obtaining the character relationship network in the novel text, the complex network measure can be used to analyze its structural characteristics. Here, the paper firstly defines and explains the network measurement and the calculation meaning of the corresponding person network.

1) *Network*: The network in this paper is actually a set of points and edges, represented by the graph $G=(V, E)$. V is the set of network nodes; E is the relationship between the nodes on the network, which called the edge set.

2) *Points and Edges*: The points in this article are the abstractions of the characters, and the labels are the specific names of the characters. The side is the co-occurrence relationship of a pair of characters such as "Confucius-Mencius". In addition, the number of points of the relationship network is abbreviated as N , and the number of sides is abbreviated as M .

3) *Average Path Length*: In a network, the distance d_{ij} between two nodes, labeled i and j respectively, is defined as the number of edges along the shortest path connecting them. The average path length L of the network, then, is defined as the mean distance between two nodes, averaged over all pairs of nodes.

4) *Diameter*: The diameter D of a network, therefore, is defined to be the maximal distance among all distances between any pair of nodes in the network.

5) *Network Density*: Referring to the closeness of the connection between nodes in a network, the mathematical meaning is the ratio of the number of edges actually existing in the network to the upper limit of the number of edges that can be accommodated. And it is denoted by ' δ '.

6) *Average Degree k*: The degree k_i of a node is usually defined to be the total number of its connections. Thus, the larger the degree, the "more important" the node is in a network. The average of k over all points is called the average degree of the network, and is denoted by ' $\langle k \rangle$ '.

7) *Character Density*: Due to the particularity of literary text, this paper also examines the character density, denoted as P :

$$P = \frac{N}{\text{Total number of words}} \quad (1)$$

In the process of selecting calculation indicators, this paper focuses on the two dimensions of the network as a whole and the internal structure of the network. On the one hand, the number of characters and relationships, the network diameter and the density of text characters are the overall portrayal of the character network of the text. On the other hand, the character network path length, the character network density, and the character network average node degree are more capable of portraying the association of the characters, and the narrative structure of the text.

TABLE II. CHARACTER NETWORK CHARACTERISTIC INDEX

Name	Number of points N	Number of edges M	average path length L	diameter D
Logogram				
	Network Density D	average degree k <k>	character density P	
Logogram				

Based on the above concepts and calculation methods, this paper uses self-programming program to call the networkx built-in standard graph theory algorithm to empirically analyze the structural characteristics of the character relationship network in 100 children's literary novels.

First, the data results are compared with the adult texts and the topological features of complex networks in the real world.

TABLE III. INDEXES OF THE THREE TEXTS

Book name	N	M	L	D	d	<k>	P	RA/NK
Grandpa have a way	3	2	0.67	2	0.67	1.32	1%	2
South of the city	17	88	1.84	4	0.65	10.44	0.29%	13
Hobbit	26	285	1.51	3	0.88	21.91	0.08%	15

IV. RESULTS AND ANALYSIS

A. The Characteristics and Analysis of the Relationship network

In network theory, "Small world theory" is a special kind of complex network structure. Most of the nodes in this network are not connected to each other, but most of the nodes pass a few cloths. The famous "Six degrees of separation" is one of the manifestations. Moreover, the diameter D and the characteristic path length L are both small, and the feature path length is proportional to the logarithm of the network scale.

In this paper, the L and D indicators of the sample are taken out and verified. More than 95% of the text L is between 1.12 and 2.00, and D is located at 2-3. Both of them are much smaller than the size of the relationship network. In addition, this paper linearly fits the feature path L with the logarithm of the number of nodes. The fitted regression equation is $y=1.6093x+0.0848$ and R^2 is 0.6997.

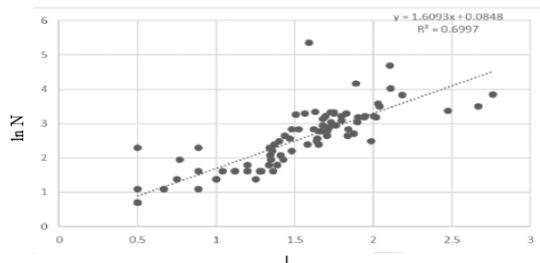


Figure 2. Linear relationship between L and ln(N).

The fitting effect is good, so it is confirmed from two aspects. The virtual social network of characters in narrative

children's literature is in line with the characteristics of the small world network.

The network diameter and average distance of children's literary figures are significantly small. Compared to the regular network with the longest average distance at the same scale, the network average distance is shortened by adding a new edge to the original rule network with a small probability. That is to increase the uncertainty and randomness of the network. The connection between any two characters is uncertain before the text is formed. It is gradually developed and realized in the process of literary narration. This is not only the embodiment of language, but also the characteristics of literature. In contrast, unlike random networks, the construction of literary texts is not completely random. It has the constraints of narrative, cognitive psychology, literature, and the writer's "deliberate action". The connection is not completely based on a certain probability, but a self-organizing network system.

The small average distance reflects the good "connectivity" of the network. One or several main characters are used as clues to connect the whole narrative process in series, so that other characters can be connected through a few parts. It reflects the common linear narrative structure of narrative literature. The frequent appearance of the protagonist is easy for children to remember and understand, and the narrative coherence is strong, which reduces the cognitive load of the reader to a certain extent, and thus is more susceptible to the preference of children's literature writers. With the length of the text, the number of characters, the number of people's relationships, the complexity of the relationship between the characters, and the network diameter of the text, it also shows an increase, from 2 to 6, gradually approaching some adult texts, such as "Beautiful Friends." It is also gradually approaching the network diameter of the real social world. This shows to some extent the development of children's cognitive ability.

In addition, this paper finds that within the same text, the connection ability and importance of different characters in the network are obviously different. Therefore, another measurement concept of complex network are defined and selected in this paper - node degree $d(v)$. The node degree of each node in the regular network is the same. The node degrees of most nodes in the random network are the same too, but the node degrees of the other nodes are very evenly distributed. However, the degree of node distribution in small world networks is not balanced. In this paper, a story of Confucius is used to calculate the node traversal of each character, and the number of nodes with the same node degree is counted as frequency. Fig. 3 shows the relationship between the node degree ($d(v)$) and the frequency of nodes.

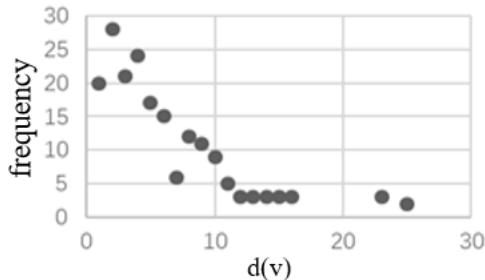


Figure 3. the relation between $d(v)$ and frequency in story of Confucius

Fig. 3 shows the phenomenon that a small number of nodes occupy most of the edges of the entire network, and the number of connected edges of most nodes is small. Due to the narrative needs of literary works, the text focuses on the description of the protagonist. Many of the supporting roles are played around the protagonist, and more marginal characters are distributed at the outermost part of the network of character relationships. Zipf's law states that in natural language, only a very small number of words are used frequently, and most words are rarely used. From this point of view, there is a similar relationship between the character degree of the character and the number of characters of the same node degree in the narrative text. Therefore, the small world characteristics of the text reflect to some extent the detailed differences in the character characterization of children's literary works. This feature is also applicable in other narrative literary works. A large number of experimental studies have shown that real networks in the world have almost a small world effect, and scientists have also found that a large number of real network nodes obey the power rate distribution. After the mining and analysis of the relationship network, we should see that although the content of children's narrative literature is quite different from the real world, there are more virtual components, but the deep structure of the relationship between characters and objects is very similar to the real world. It shows that the beauty of literature is to reflect reality.

B. Method and indicator validity test

This paper plans to use the topological measure indicators of the extracted text person network to evaluate and predict the complexity index of the characters in the children's literature. In order to determine the validity of this quantitative method and the accuracy of the final prediction, this paper examines the differences in the ability of different topological measures to reflect the complexity of the relationship in the children's literature and the impact of each indicator on the prediction of the complexity of the relationship. This paper evaluates the degree of correlation between individual network eigenvalues and tag values, sorting from high to low, leaving a high degree of correlation. In terms of specific methods, this paper draws the correlation heat map of each index and measures the correlation between the characteristic

index and the complexity of the relationship based on the Pearson correlation coefficient:

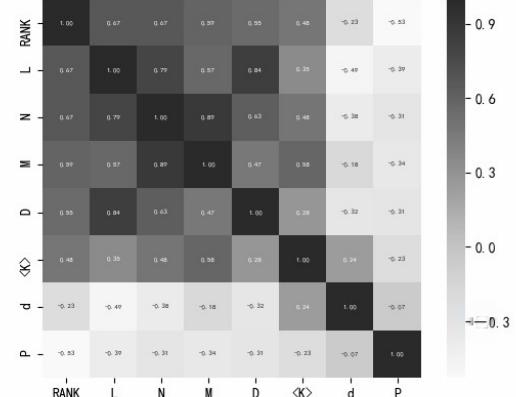


Figure 4. Heatmap of the eight indicators

It can be seen from the Fig.4 that among the selected six characteristic indicators, the absolute value of the correlation degree with the tagged classification information RANK is: the character network path length L(0.67), the number of characters N(0.67), the number of relationship pairs M(0.59), network diameter D (0.55), average degree $\langle K \rangle$ (0.53), text person density P (0.45), character network density d (0.2). It can be seen that there is a strong correlation between the L, N and text person relationship complexity, and the characteristics M, D, $\langle K \rangle$, P and the text person relationship complexity are moderately related, and d is weakly related to the text person relationship complexity.

Therefore, this paper confirms the feasibility of using the topological measure index of the extracted text person network to predict the character relationship complexity index in children's literature. After the correlation test, this paper excludes the character network density d index, and uses the residual measure parameter as the feature to predict the complexity.

C. Regression model prediction and evaluation

Because the corpus size is not large enough, in order to ensure that the sample is used as much as possible, this paper uses the 5-fold cross-validation method to record the mean square error as the evaluation index, and selects the MAE and R2 results from the regression models of 3 commonly used machine learning.

TABLE IV. EFFECT OF REGRESSION MODEL EVALUATION

Model	MAE(Mean Absolute Error)	R^2
Linear Regression	1.487	0.595
XGBoost	1.095	0.524
Lasso	1.743	0.60

The average absolute error of the three models on the test set is about 1. Due to the ambiguity and subjectivity of the complexity of the relationship between the characters, this paper calculated the **average difference** (AD) of the scores given by the five scorers, it is 1.34. The experimental result error is within an acceptable range, thus confirming the validity of the experimental results.

The resulting regression equation is:

$$Y(\text{grade of difficulty})=0.071N+3.849L+0.018< k>+0.156D-0.13P+0.0159M +1.267. \quad (2)$$

In order to further confirm the applicability of the method, we randomly selected another 20 children's literature texts for non-experimental data to calculate the complexity of the relationship between the characters, and used the trained regression model to predict .The input is the characters list for each text ,the network average path length L, the number of people N, the number of relationship pairs M, the network diameter D, the network average degree $< K >$, the text person density P six index values, and the output is the text relationship complexity prediction of each text value. The test results are shown below:

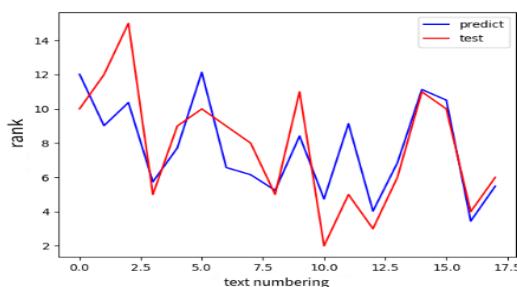


Figure 5. Result of prediction

The Fig.5 show that the average error remains at 1.42, which is within acceptable limits. But it also reflects the greater uncertainty of narrative in children's literary texts . The results of the small sample size experiment are more to prove the validity of this evaluation method. In order to improve the predictive validity, the quantity and quality of the experimental data still need higher requirements.

V. CONCLUSION AND FUTURE WORK

Based on the methods of predecessors, this paper applied the CRF++ training model to 100 children's literature works. We automatically extracted the characters and the social network between the characters based on the co-occurrence relationship to characterize and analyze the character relationship of the literary novels. The result is consistent with the content of the novel.

In this paper, we presented a method for characterizing a text of literary fiction by extracting the network of social conversations that occur between its characters. This allowed us to take a systematic and wide look at a large corpus of texts, an approach which complements the narrower and deeper analysis performed by literary scholars and can provide evidence for or against some of their claims. After that, this study successfully applied the network attribute of some measure indicators of complex networks to the automatic analysis, mining and comparison of the character network and plot complexity of different literature texts.

On the basis of this work, we can expand the text readability measurement dimension of children's literature text from character, word and sentence to the whole book. Therefore, the score of this index will be used in the

evaluation of text readability together with other indexes such as character frequency, word frequency, semantic transparency, sentence complexity, sentence length and so on.

In order to make the evaluation and prediction effect better and enhance the fitting effect of the model, this paper plans to collect more experimental data, and further consider different text characters in the data collection. The complexity of the text is as balanced as possible, facilitating the construction of subsequent machine learning models. In the construction of the personal relationship network, it is expected to try to improve the recognition accuracy based on the more characteristic BI-LSTM-CRF name recognition method. In addition, the paper finds that the diversity, ambiguity and uncertainty of the reference affect recognition of character to identify relationship. For example, "君主(prince)" as a collective noun, representing a number of prince entities, such as "Jin Wen Gong", "Chu Zhuang gong", etc. Besides, in different contexts, specific reference objects, computer automatic identification and generational digestion requires further exploration and research. In addition, in the difficulty of children's literature texts, the upper limit of the difficulty of our defense is difficult to control. This paper calculated the weight of the character and the weight of the edge in the calculation of the relationship between the characters, but it is not used in the network construction for the time being. In the later research, we need to solve how to add the weight of the edge and the character to the construction and evaluation of the network.

REFERENCES

- [1] Moretti, F., Graphs, Maps, Trees: Abstract Models for a Literary History. London: Verso Press. 2005.
- [2] Alberich.R ,Miro-Julia.J,Rosello F.Marvel Universe Looks almost Like a Real Social Network,New York :Cornell University Library,2002.
- [3] He, H., Barbosa, D. & Kondrak, G., Identification of Speakers in Novels. Meeting of the Association for Computational Linguistics (pp. 1312-1320). Sofia., 2013.
- [4] Rydberg- Cox J. Social networks and the language of greek tragedy. Journal of the Chicago Colloquium on Digital Humanities and Computer Science, 2011,1 (3).
- [5] Sparavigna A.C., Marazzato R.. Graph visualization software for networks of characters in plays. International Journal of Sciences, 2014,3 (2): 69– 79.
- [6] Watts D J , Strogatz S H . Collective Dynamics of Small World Networks[J]. Nature, 1998, 393(6684):440-442.
- [7] Y.Tang, S.Wang,J.Hu, "Small World" on BailuVillage: An Empirical Analysis of the Network of Character Relations in the TV BailuVillage, China Radio & TV Academic Journal,2018,pp117-12
- [8] H.Y. Liu , X.H. Yang . "Quantifying the Vicissitude of Fitzgerald's Creativity a Statistical Analysis Based on Lexical Measures." Information Technology & Artificial Intelligence Conference 0.
- [9] Chen, Bikun , and Y. Wang . "Character interaction network analysis of chinese literary work- A preliminary study." Proceedings of the Association for Information Science & Technology 53.1(2016):1-4.
- [10] H.P. Zhu, Z.Y.Luan, Text Mining of Character Relation Network in Water Margin, Social Sciences Review,
- [11] Stiller J., Hudson M. ,Weak links and scene cliques within the small world of Shakespeare. Journal of Evolutionary Psychology, 3(1), 2005 pp.57-73

A Chinese word segment model for energy literature based on Neural Networks with Electricity User Dictionary

Bochuan Song

Artificial Intelligence on Electric Power System

State Grid Corporation Joint Laboratory

Global Energy Interconnection Research Institute co.Ltd

Beijing, China

songbochuan@geiri.sgcc.com.cn

Qiang Zhang

Artificial Intelligence on Electric Power System

State Grid Corporation Joint Laboratory

Global Energy Interconnection Research Institute co.Ltd

Beijing, China

zhangqiang1@geiri.sgcc.com.cn

Bo Chai

Artificial Intelligence on Electric Power System

State Grid Corporation Joint Laboratory

Global Energy Interconnection Research Institute co.Ltd

Beijing, China

chaibo@geiri.sgcc.com.cn

Quanye Jia

Artificial Intelligence on Electric Power System

State Grid Corporation Joint Laboratory

Global Energy Interconnection Research Institute co.Ltd

Beijing, China

jiaquanye@geiri.sgcc.com.cn

Abstract—Traditional Chinese word segmentation (CWS) methods are based on supervised machine learning such as Conditional Random Fields(CRFs), Maximum Entropy(ME), whose features are mostly manual features. These manual features are often derived from local contexts. Currently, most state-of-art methods for Chinese word segmentation are based on neural networks. However these neural networks rarely introduce the user dictionary. We propose a LSTM-based Chinese word segmentation which can take advantage of the user dictionary. The experiments show that our model performs better than a popular segment tool in electricity domain. It is noticed that it achieves a better performance when transferred to a new domain using the user dictionary.

Keywords-Chinese word segmentation, electricity user dictionary, neural networks

I. INTRODUCTION

Words are the basic units to process for most Natural Language Processing (NLP) tasks. However, most east Asian languages are written without explicit word delimiters, including Chinese. As a result, word segmentation is usually the first step of NLP in east Asian languages process.

There are two obstacles for Chinese word segmentation. One is overlapping ambiguity. For example, the sentence “南京市长江大桥顺利通车”(Nanjing Yangtze River Bridge was successfully opened to traffic). The segmentation of the utterance can be “南京市(Nanjing City) / 长江大桥(Changjiang Bridge) / 顺利(successfully) / 通车(opened to traffic)”. It can also be segmented as “南京(Nanjing City) / 市长(mayor) / 江大桥(Jiang Daqiao) / 顺利(successfully) / 通车(opened to traffic)”. The middle character of an overlapping ambiguous string can constitute words with the characters to both their left and their right [1].

The other one is the out of vocabulary problem (OOV). For example, while “国家电网”(State Grid) appears as one word at test time, a model trained on a dataset where the character subsequence is segmented as two words “国

家”(Country) and “电网”(Grid) will split the word “国家电网”(State Grid) into two words.. Some of the errors will be almost impossible to solve [2].

There have been a wide range of studies on Chinese word segmentation. Most methods formalize this task as a sequence labeling problem [2], [3]. In a supervised machine learning fashion, this problem may adopt various models e.g., Maximum Entropy [4] and Conditional Random Fields [5]. These methods usually heavily depend on manual features. Tseng et al. [6] designed a CRF segmenter using n-gram character features. There are more complicated features, e.g., accessor varitey criteria [7], conditional entropy features [8]. As neural networks can extract features on their own, neural models have been widely used for NLP tasks. For the task of CWS, Zheng et al. [3] adapted the general neural networks architecture for sequence labeling and used character embeddings as input to a two-layer network. Cai et al. [9] proposed a methods based on both character features and word embeddings. The beam search method is used to trade off the search complexity and search accuracy.

By introducing the user dictionary with the aim of solving the OOV problem, the traditional sequence labeling methods (e.g., CRFs) are limited by the context window to reach the long term history information. In addition, the neural network models cannot deal with the OOV problem, due to the lack of utilization of the user dictionary.

It is our goal to develop NLP technologies in the field of electricity. The Chinese word segmentation is the fundation of most NLP tasks, e.g., name entity recognition (NER) and information extraction (IE). A common model usually obtains a lower performance in a novel specific domain. It is necessary to study the CWS problem in the electricity field. Some prior work has been done to tackle the challenge brought by CWS in the area, e.g., collection of training data, and construction of an electricity user dictionary.

In this paper, we proposed a method based on neural networks which can model all the previous history information rather than a fixed context window size as machine learning models do. In order to get better performance on out-of-domain corpus, the proposed model is designed to adopt the user dictionary. To the best of our knowledge, it is the first time to introduce user dictionary in neural networks to solve the CWS problem.

The paper is organized as follows. Our methods are introduced in section II. The analysis and results are discussed in section III. Section IV presents the conclusion.

II. METHODS

Chinese word segmentation task is regarded as a character-based sequence labeling problem. Specifically, each character in a sentence is assigned with a label, i.e., $\mathcal{L} = \{B, M, E, S\}$, indicating the begin, middle, end of a word or a word with a single character. To segment a character sequence, we employ neural networks to extract the features and employ the CRFs to utilize the dependencies between tags. To improve the performance in a different domain, we introduce a electricity user dictionary which is implemented between the LSTM layer and the CRFs layer. Fig. 1 illustrates the proposed model.

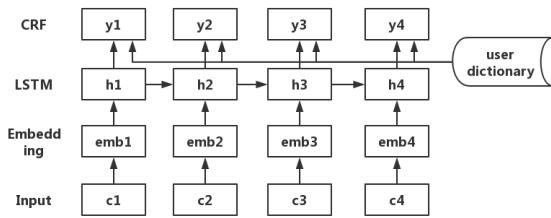


Figure 1. Our LSTM CRF User dict model.

A. Embedding

As neural network models are based on vector, firstly we generate the character embeddings randomly. Formally, we have a character dictionary \mathcal{D} of size $|\mathcal{D}|$. Each character \mathbf{c}_i is represented as a real-valued vector $\mathbf{c} \in R^{d_c}$, where d_c is the dimensionality of character vector. The character vectors are inserted into a matrix $\mathbf{M} \in R^{d_c \times |\mathcal{D}|}$. The embedding layer retrieves the character \mathbf{c}_i according to its index to get its embedding.

B. LSTM

The LSTM neural network [10] is an extension of Recurrent Neural Networks(RNNs), which has been widely adopted in NLP tasks. RNNs are a family of neural networks that operate on sequential data. In the RNN, a sequence of vector $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ is taken as input, and another sequence of vector $(\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n)$ is finally obtained. Although RNNs can learn long-dependencies in theory, but they usually failed in practical because of gradient exploding and vanishing problems [11]. The LSTM is designed to cope with the issues by introducing

a memory-cell and three gate functions, i.e., input gate, forget gate and output gate. LSTMs have been shown to capture long-dependencies better than RNNs [12]. There also are variant implementation of LSTMs. In our model, we use the following implementation:

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \quad (1)$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \quad (2)$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \quad (3)$$

$$\tilde{c}_t = \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \quad (4)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (5)$$

$$h_t = o_t \odot \sigma_h(c_t) \quad (6)$$

f_t, i_t, o_t are forget gate, input gate and output gate, respectively. \odot is elementwise multiplication. x_t is the input of LSTMs at time step t and h_t is the output of LSTMs at time step t .

C. CRF

A simple but effective model use the h_t as feature to make independent classification by adding a softmax layer following the LSTM layer, which models the distribution as follows:

$$P(y_i) = P(y_i | x_1, x_2, \dots, x_{i-1}) \quad (7)$$

However, it may fail when there are strong dependencies between tags. Chinese word segmentation is one such task, since the characterize interpretable sequences of tags imposes several hard constraints (e.g., tag *I* cannot follow tag *E* or tag *S*). So we use CRFs as labeling layer which can model these constraints. For an input sequence,

$$\mathbf{x} = (x_1, x_2, \dots, x_n), \quad (8)$$

we consider \mathbf{E} to be the matrix of scores derived from the output of the LSTM layer. \mathbf{E} is of size $(n \times k)$, where k is the number of distinct tags. $E_{i,j}$ represents the score of the j th tag of the i th word in a sentence. For a sequence of predictions

$$\mathbf{y} = (y_1, y_2, \dots, y_n), \quad (9)$$

we define its score to be

$$s(\mathbf{x}, \mathbf{y}) = \sum_i T_{y_{i-1}, y_i} + \sum_i E_{i, y_i} \quad (10)$$

in which \mathbf{T} is the transition matrix of CRF, T_{y_{i-1}, y_i} means the score of a transition from tag y_{i-1} to tag y_i . A softmax over all tag sequences yield a probability of \mathbf{y}

$$P(\mathbf{y} | \mathbf{x}) = \frac{e^{s(\mathbf{y}, \mathbf{x})}}{\sum_{\tilde{\mathbf{y}}} e^{\tilde{s}(\mathbf{y}, \mathbf{x})}} \quad (11)$$

For training, we maximum the $P(\mathbf{y} | \mathbf{x})$ where \mathbf{y} is the truth ground tag sequence. During decoding, we predict the tag sequence that obtains the maximum score given by:

$$\hat{\mathbf{y}} = \operatorname{argmin}_{\tilde{\mathbf{y}}} s(\tilde{\mathbf{y}}, \mathbf{x}) \quad (12)$$

In evaluation stage, the Viterbi algorithm is utilized for decoding. The Viterbi algorithm is a dynamic programming algorithm for finding the most likely sequence of hidden states [13]. Its algorithm is described in Algorithm 1.

Algorithm 1: Viterbi Algorithm

Data: emission matrix, output of LSTM, E , and transition matrix T

Result: best path P

```

1 /* initialize the parameters.      */
2 paths ← List;
3 scores ← List;
4 scores[0] ← E[start_tag,:];
5 for i ← 1 to seqlen do
6   potentials ← scores[i - 1] + T;
7   score, path ← max(potentials);
8   paths[i] ← path;
9   scores[i] ← score + E[i,:];
10 end
11 /* construct the most likely      */
   sequence backwards             */
12 viterbi_score, viterbi_path ←
  max(scores[seqlen,:]);
13 viterbi_paths[seqlen] ← [viterbi_path];
14 for i ← seqlen - 1 to 1 do
15   viterbi_path ← paths[i - 1];
16   viterbi_paths[i] ← viterbi_path;
17 end

```

D. User Dictionary

In traditional segmentation fashion, the user dictionary is a widely used method, e.g., n-gram, forward maximum matching. The mentioned models can achieve better performance with a well-organized user dictionary when applied in a novel-specific domain. As far as we know, there are not any neural-based models that adopt user dictionary. The user dictionary is used to combine the LSTM and the CRF to tackle the word segmentation problem. Firstly, we find all candidate words in a input sentence by looking up the user dictionary. Then, for each character in every candidate words, we add a weight to the corresponding E_{i,y_i} according to the charater position in the candidate word to get \hat{E}_{i,y_i} . Finally replace E_{i,y_i} by \hat{E}_{i,y_i} in (10)

III. EXPERIMENT

A. Datasets and Metrics

We use two datasets, the PKU [14] and the Energy. The overview of the two datasets is presented in Table I.

Table I
OVERVIEW OF DATASETS

Dataset	Trainset	Devset	Testset
PKU	31479	4526	9078
Energy	9575	1137	2240

PKU dataset is collected from People Daily. It has been widely used for CWS task [8], [10], [15]. Energy dataset contains around 12,000 sentences. We also make a user dictionary, which contains about 9,100 words and is collected from other corpora. For evaluation, we use F1-measure. The formula is given as following:

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (13)$$

$$Precision = \frac{N_Right}{N_Pred} \quad (14)$$

$$Recall = \frac{N_Right}{N_Gold} \quad (15)$$

The N_Right is the total number of correct predicted segment words. The N_Pred is the total number of predicted segment words. The N_Gold is the total number of ground truth words.

B. Hyper-parameters

Hyper-parameters of neural networks models has a strong impact on their performance. In our model, the size of character embeddings is 128. The hidden size of LSTM is 256. We use adam optimizer, and the learning rate is 0.01.

C. Results and Analysis

Several experiments have been conducted. Firstly, we compare our model in the general domain with the popular segment tool Jieba segmenter on PKU dataset. The results are showed in Table II.

Table II
RESULTS ON PKU

Model	Precision	Recall	F1
Jieba	0.8174	0.7829	0.7998
Our model	0.8932	0.9011	0.8971

It shows that the our model is obviously better than the widely used Jieba segmenter in the general domain.

Then we train our model on Energy trainset, and test both our model and Jieba segmenter on the testset of Energy. We also find that if we combine the trainsets of PKU and Energy our model will perform marginally better. The results are showed in Table III.

Table III
RESULTS ON ENERGY

Model	Trainset	Precision	Recall	F1
Jieba	-	0.7547	0.8157	0.7840
Jieba-user_dict	-	0.7720	0.8054	0.7884
Our model	Energy	0.7794	0.8387	0.8079
Our model	Energy+PKU	0.7875	0.8371	0.8116

As the Jieba segmenter gains a little improvement on F1-measure score after the user dictionary employed (the comparison between Jieba and Jieba-user_dict in Table III), it shows that the dictionary introduce some domain information. As a result, the comparison between

our model trained on Energy and Jieba-user_dict shows that our model also gains improvement with Jieba segmenter. It is noticed that our model perform better after we combine trainsets of PKU and Energy. The reason can be illustrated as that Energy dataset is not sufficient. The bigger datasets, PKU, can introduce some common knowledge into our model, which has positive effect on Chinese word segmentation. It is a simple data transfer mode of transfer learning.

Finally we conduct experiment to evaluate the effects of the user dictonay on our neural-based model. We have two sets of models. The first one is trained on Energy trainset, and is tested on Energy testset. The second one is trained on PKU trainset, and is tested on Energy testset. The results are showed in Table IV.

Table IV
RESULTS ON USER DICTIONARY NEURAL MODEL

Model	Trainset	Precision	Recall	F1
our model	PKU	0.5770	0.6812	0.6248
our model user dict	PKU	0.6001	0.6762	0.6359
Our model	Energy	0.7794	0.8387	0.8079
our model user dict	Energy	0.7834	0.8180	0.8003

In order to make the difference clear, we add our model without user dictionary which trained on Energy into the Table IV. The first two lines of Table IV shows that the user dictionary has positive effect on segmentation, since the model with the user dictionary gains 1.11% improvement on F1-measure score compared with model without the user dictionary. These two model are trained on PKU dataset. it is noteworthy that when the trainset changed from PKU to Energy, the results are different. In the last two line of Table IV, it shows that the introducing of the user dictionary degrades the performance of the proposed model when trained on Energy.

We investigate the overlapped words among the dictonary and testset of Energy. In the testset, there are 919 sentences where there are subsequences of characters, which can compose words in the dictionary. However among these sentences, only 450 sentences have the correct words after segment. When the our model is trained on Energy, it learns the sufficient knowledge of the electricity domain. It is the gap between the domain from which the dictionary collected and the Energy dataset that degrades the performance when our model is trained on Energy with the user dictionary. For a example, there is a word “能源基地”(Energy base) in the dictionary. For a sentence in the testset, “新形势下能源基地开发潜力...”(Energy base development potential under the new situation), the correct segmentation is “新(new) / 形势(situation) / 下(under) / 能源(energy) / 基地(base) / 开发(development) / 潜力(potential)”. But the word in the dictionary will force the character sequence “能源基地”(energy base) to be one word. When the model is trained on general domain dataset, PKU, it lack of the domain knowledge. Thus the user dictionary is supposed to improve the performance on Energy testset.

IV. CONCLUSION

We proposed a neural-based model with the user dictonary. Experiments show that the proposed model gains improvement on both general domain and energy domain compared with Jieba segmenter. One attractive feature of the proposed model is that it can adopt a user dictionary, which can transfer the proposed model to a new domain with little work. The experiments show that the proposed model with a user dictionary obtains a better performance when trained on a general domain and tested on a new domain compared to the model without a user dictionary.

In the future, we plan to introduce the user dictionary into the transition matrix of the CRF, which will make the introducing more seamlessly and more flexibly. In addition, our basic model, LSTM-CRF, still need to be improved. Some word-based segmentation mechanism can be introduced in the future work.

ACKNOWLEDGMENT

This research was supported by State Grid Company Research Project “Task-driven multi-round dialogue generation based on semantic understanding” under grant 5455HJ190008.

REFERENCES

- [1] Ma, Guojie, Xingshan Li, and Keith Rayner. “Word segmentation of overlapping ambiguous strings during Chinese reading.” Journal of Experimental Psychology: Human Perception and Performance 40.3 (2014): 1046.
- [2] Ma, Ji, Kuzman Ganchev, and David Weiss. “State-of-the-art Chinese word segmentation with bi-lstms.” arXiv preprint arXiv:1808.06511 (2018).
- [3] Zheng, Xiaoqing, Hanyang Chen, and Tianyu Xu. “Deep learning for Chinese word segmentation and POS tagging.” Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. 2013.
- [4] Low, Jin Kiat, Hwee Tou Ng, and Wenyuan Guo. “A maximum entropy approach to Chinese word segmentation.” Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing. 2005.
- [5] Peng, Fuchun, Fangfang Feng, and Andrew McCallum. “Chinese segmentation and new word detection using conditional random fields.” Proceedings of the 20th international conference on Computational Linguistics. Association for Computational Linguistics, 2004.
- [6] Tseng, Huihsin, et al. “A conditional random field word segmenter for sighan bakeoff 2005.” Proceedings of the fourth SIGHAN workshop on Chinese language Processing. 2005.
- [7] Feng, Haodi, et al. “Accessor variety criteria for Chinese word extraction.” Computational Linguistics 30.1 (2004): 75-93.
- [8] Gao, Qin, and Vogel Stephan. “A multi-layer Chinese word segmentation system optimized for out-of-domain tasks.” CIPS-SIGHAN Joint Conference on Chinese Language Processing. 2010.

- [9] Cai, Deng, and Hai Zhao. “Neural word segmentation learning for Chinese.” arXiv preprint arXiv:1606.04300 (2016).
- [10] Hochreiter, Sepp, and Jürgen Schmidhuber. “Long short-term memory.” Neural computation 9.8 (1997): 1735-1780.
- [11] Hochreiter, Sepp, et al. “Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.” (2001).
- [12] Hochreiter, Sepp, and Jürgen Schmidhuber. “Long short-term memory.” Neural computation 9.8 (1997): 1735-1780.
- [13] Lou, H-L. “Implementing the Viterbi algorithm.” IEEE Signal processing magazine 12.5 (1995): 42-52.
- [14] Emerson, Thomas. “The second international Chinese word segmentation bakeoff.” Proceedings of the fourth SIGHAN workshop on Chinese language Processing. 2005.
- [15] Huang, Weipeng, et al. “Toward Fast and Accurate Neural Chinese Word Segmentation with Multi-Criteria Learning.” arXiv preprint arXiv:1903.04190 (2019).

Fusion of Image-text attention for Transformer-based Multimodal Machine Translation

Junteng Ma, Shihao Qin, Lan Su, Xia Li[✉]

School of Information Science and Technology

Guangdong University of Foreign Studies
Guangzhou, China

{juntengma, shihao_qin, destinyofloveing}@126.com
xiali@mail.gdufs.edu.cn

Lixian Xiao

Faculty of Asian Languages and Cultures

Guangdong University of Foreign Studies
Guangzhou, China

200110732@oamail.gdufs.edu.cn

Abstract—In recent years, multimodal machine translation has become one of the hot research topics. In this paper, a machine translation model based on self-attention mechanism is extended for multimodal machine translation. In the model, an Image-text attention layer is added in the end of encoder layer to capture the relevant semantic information between image and text words. With this layer of attention, the model can capture the different weights between the words that is relevant to the image or appear in the image, and get a better text representation that fuses these weights, so that it can be better used for decoding of the model. Experiments are carried out on the original English-German sentence pairs of the multimodal machine translation dataset, Multi30k, and the Indonesian-Chinese sentence pairs which is manually annotated by human. The results show that our model performs better than the text-only transformer-based machine translation model and is comparable to most of the existing work, proves the effectiveness of our model.

Index Terms—Multimodal Machine Translation; Image-text attention; Transformer-based; Self-attention.

I. INTRODUCTION

Multimodal neural machine translation [1,2] (MNMT) aims to use several modalities, such as image modal information, to help solve the problem of semantic ambiguity in plain text, so as to improve the quality of machine translation.

Previous work of multimodal machine translation can be roughly divided into recurrent neural network based (RNN-based) framework and Transformer-based framework. In the work of RNN-based architecture, the main idea is to integrate different forms of image features into different parts of the model, such as the work of Huang et al. [3], Calixto et al. [4-5] and Caglayan et al. [6]. Elliott et al. [7] proposed a way of “imagination” which decompose the multimodal machine translation task into two subtasks, one is a regular translation task, and the other task is to predict the corresponding visual representation by the encoded sentence representation, that is making the distance between the text and the image representation closer.

With the advantages of self-attention mechanism for text-only machine translation, which is proposed by Vaswani et al. [8], some work began to extend Transformer-based framework for multimodal machine translation. For example, Helcl et al. [9] used Transformer to build MNMT model. They proposed two ideas, one is to modify the structure of decoder by adding

a visual cross-attention layer, the other is to use the imagination [7] method. Inspired by Caglayan et al. [6], Grönroos et al. [10] regarded the image features as pseudo words and used a gating procedure to process the image feature.

Different from the previous Transformer-based MNMT model, our model is mainly to change the internal structure of the encoder. We argue that if the visual information can be applied to the source words at the end of encoding layer, then the semantic information of those words that are more related to the image could be enhanced. Based on this motivation, we propose to add an image-text attention layer in the end of encoder layer, so that the model can receive two modalities in the end of encoder and capture the relationship between visual and text information, refining the sentence representation. In this way, the output of the encoder may contain both image and textual representations and the key information related to image is strengthened to the representation. The contribution of our work are as follows:

(1) In order to capture the different weights between the words that is relevant to the image or appear in the image, we propose to add an image-text attention layer in the end of encoder layer, so we can get a better text representation that fuses the semantic relationship between image and text words.

(2) We carried several experiments on the Multi30k dataset, not only on English to German sentence pairs but also on Indonesian to Chinese sentence pairs (The validation data and test data of Indonesian to Chinese sentence pairs are annotated by human.) We will show that our extension of multimodal machine translation model performs better than the text-only transformer-based machine translation model.

II. MODEL

A. The Architecture of Our Model

Followed by the encoder-decoder scheme, the architecture of our model is based on Transformer NMT [8]. Transformer contains self-attention layer and a feed-forward network layer. Both the end of encoder and decoder consists of several layers to get better text representation by deepening the network. Since Transformer does not have the ability to capture sequence information of source words like RNN, position embedding is added to the word embedding to give position information to each word.

[✉] corresponding author: xiali@mail.gdufs.edu.cn

In order to enhance semantic information of the text words and improve the translation quality of text-only NMT model, we extend the Transformer-based machine translation model for multimodal machine translation by fusing with the attention of image vision and text words. The architecture is shown in Figure 1.

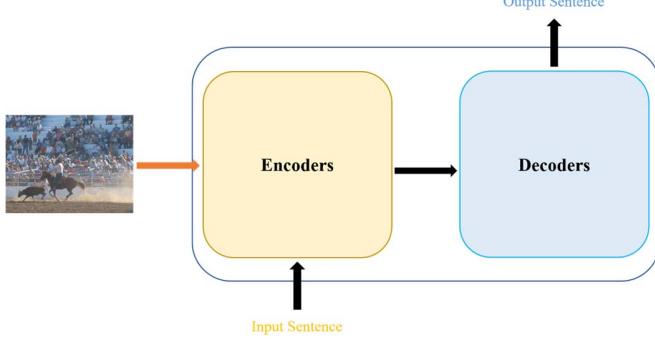


Figure 1. The architecture of our model.

B. Fusion of Image-text Attention

The main method of image fusion in previous work is to change the structure of the decoder layer [9] or to apply a gating procedure to the image feature in the output of encoder or decoder [10]. Different from previous work, we introduced an image-text attention layer in the model, which is between the self-attention layer and the feed-forward network layer. The model is shown as Figure 2.

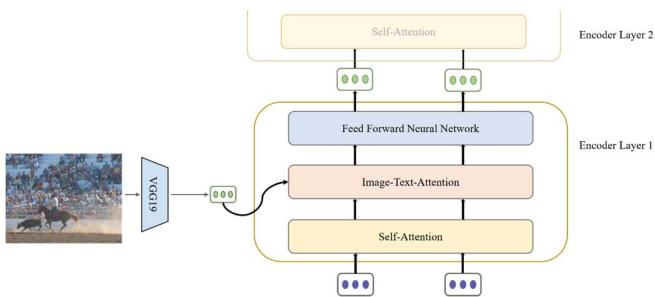


Figure 2. Encoder of our model.

In the original Transformer NMT, each encoder layer only contains a self-attention layer and a feed-forward network layer. In self-attention layer, each input vector is converted into three different vectors, which are the Query vector, the Key vector, and the Value vector. They are obtained by multiplying the word embedding and three different weight matrices W^Q , W^K , W^V , where the shapes of these three matrices are the same. For a given Q , the similarity function is first used to calculate and compare with each K . Then, the result are normalized to obtain weights to calculate the context vector, which is a weighted sum of weights and V .

The similarity function of self-attention in Transformer uses scaled dot-product, so the results of self-attention is as Equation (1), where d is the dimension of Q , K , V .

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (1).$$

In order to improve the performance of the self-attention layer, the Multi-Head attention mechanism is used, which not only expands the ability of the model to focus on different positions, but also provides multiple representative subspaces of the self-attention layer. The specific approach is to project the input Q , K , V into multiple subspaces, that is, Multi-Head, performs self-attention calculation, and concatenate the output of each head and then feed to a fully connection layer. The Equation is as (2), Where W_i^Q , W_i^K , $W_i^V \in \mathbb{R}^{d_{model} \times d}$, $W^O \in \mathbb{R}^{hd \times d_{model}}$, d_{model} is the model's dimension, h is the number of heads.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)W^O, \text{ where } \text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (2).$$

In our model, as shown in Figure 2, the Q in the image-text attention layer is regarded as the image feature corresponding to the sentence, which is extracted by a pre-trained VGG-19 [11]. K , V are derived from the self-attention layer's output. The context vector obtained by the Multi-Head attention mechanism thus contains the interaction information between the image and the source language sentence. Supposed that the extracted image features as I , and the output of image-text attention layer is expressed as Equation (3):

$$\begin{aligned} \text{Context}_{img} &= \text{MultiHead}(I, K, V) \\ &= \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)W'^O, \end{aligned} \text{ where } \text{head}_i = \text{ImageTextAttention}(IW_i^I, KW_i^K, VW_i^V) \quad (3).$$

All layers are interconnected with residual connections and their outputs are normalized by layer normalization [12].

C. Decoder Layer of Our Model

Followed the work of Vaswani et al. [8], the decoder layer of our model is shown in Figure 3. In the decoder layer, self-attention is masked to prevent the decoder to process the “future” states. Different from the encoder layer, there is an additional sub-layer called encoder-decoder-attention layer after self-attention layer, which attends to the final output of the encoder and the output of self-attention layer. It allows every position in the decoder to attend over all positions in the input sequence.

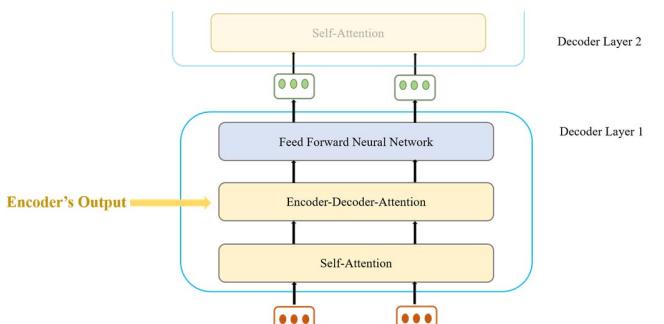


Figure 3. Decoder of our model.

D. Model Training

Followed previous work, we define multimodal translation as the task of producing target language translation y , given the source language x and an image v . Multimodal neural machine translation can be formed as minimizing the negative log-likelihood of a translation model that is additionally conditioned on the corresponding image. The loss function is shown in Equation (4).

$$J(\theta) = - \sum_j \log p(y_j | y_{<j}, x, v) \quad (4)$$

Where x is the source sentence description of the image v , y_j is the j -th token of the target sentence y .

III. EXPERIMENTS

A. Dataset

We use Multi30k [13] dataset in our experiments to test the performance of our model. Multi30k is a multilingual expansion of the original Flickr30k [14] dataset for image description generation task. Each image in Multi30k consists of an English sentence description and a German sentence translated by professional translators. The data of Multi30k is divided into training set, validation set and test set. The image size of training set, validation set and test set are 29,000, 1,014 and 1,000 respectively in Multi30k, each image has one sentence pair: the original English description and its German translation.

As for the preprocess of the dataset, we use Moses [15] scripts to tokenize, normalize-punctuation and true case for both English and German sentences. And we use byte pair encoding compression algorithm [16] to convert the words into sub-words. We use the training set of Multi30k to train the model, and validation set for model selection and test set for evaluation. We use BLEU4 [17] and METEOR [18] to measure the quality of the translation result.

B. Experiment Setup

As for the image features, we use a VGG-19 network which is pre-trained on ImageNet [19] to extract visual representations. We feed the images in Multi30k and use the 4096D activations of the penultimate fully connected layer FC7 as our global image features.

For the configuration of Transformer, we use the Transformer network with 6 layers and 8 heads, the model dimension with 512 and feed-forward network dimension with 2048. We use Adam optimizer [20] with initial learning rate 0.01, and Noam learning rate decay scheme [8] with $\beta_1 = 0.9$, $\beta_2 = 0.997$, $\epsilon = 10^{-9}$ and 16,000 warm-up steps. In the experiment, dropout rate is 0.1, beam search size is 10. We train the model on training set and select the best model according to BLEU4 scores performed on validation set. We report the results on test set with the best model.

C. Experimental Results

In this section, we will introduce the baselines we used in our experiments and present the results of our model on different datasets.

1) Baselines

In order to verify the performance of our model, we use several baseline models for comparison, they are: Transformer Text-only NMT, Huang et al., 2016[3], Calixto et al., 2017 [4], Calixto et al., 2017[5], Caglayan et al., 2017[6], and Helcl et al., 2018[9]. All these methods can be roughly classified as RNN-based methods and Transformer-based methods.

RNN-based. We use several RNN-based models as our baselines, they are Huang et al., 2016[3], Calixto et al., 2017 [4], Calixto et al., 2017[5] and Caglayan et al., 2017[6]. The work of Huang et al. [3] proposed a MNMT model that fuses regional images which is extracted by RCNN [21], together with the entire image as pseudo words. Calixto et al. [4] used two independent attention mechanisms for text and images. Calixto et al. [5] incorporated global image feature to initialize the decoder's hidden state. And the work of Caglayan et al. [6] modulated each target embedding with global image feature, which is extracted by ResNet[22], using element-wise multiplication.

Transformer-based. We also use Transformer-based model as our baselines. We use text-only Transformer-based model and multimodal as our two baselines. Text-only Transformer-based NMT model is trained on Multi30k's sentence pairs, and does not use the visual information. Our model is an extension of it. We also use the work of Helcl et al. [9] to be another baseline. Helcl et al. [9] proposed a Transformer network with Imagination method [7] which decompose the MNMT into two sub-tasks: translation task and use the encoded textual representation to predict the corresponding image feature, to bring their distance closer.

2) Results on English to German Sentence Pairs.

Table 1 shows the results on English-German sentence pairs of Multi30k, the bold number is the results of our model. As we can see, our proposed model performs better than the Text-only NMT and improves 0.25 BLEU scores and 0.5 METEOR scores. That is to say, the image features do have positive effect on the model. Compared with most of the baseline model, like Huang et al. [3] and Calixto et al. [4][5], Image-text attention still performs better in BLEU and METEOR metrics, and is comparable to the work of Caglayan ta al. [6]. But there still has a gap between our model and Helcl et al. [9]'s imagination system.

TABLE I. THE RESULTS ON ENGLISH TO GERMAN SENTENCE PAIRS.

	Models	BLEU4	METEOR
RNN-based	Huang et al. (parallel RCNNs) [3]	36.5	54.1
	Calixto et al. (SRC+IMG) [4]	36.5	55.0
	Calixto et al. (IMG _D) [5]	37.3	55.1
	Caglayan et al. (trg-mul) [6]	37.8	57.7
Transformer-based	Helcl et al.(Imagination) [9]	38.8	56.4
	Text-only	37.14	55.4
Our model	Image-Text-Attention	37.39	55.9

3) Results on Indonesia to Chinese Sentence Pairs.

In this paper, we also evaluated our model on low-resource language data. We manually translated the English and German

sentence pairs of the validation and test set into the Indonesian and Chinese respectively. As for the training set of Multi30k, we use Google Translation¹ to translate the English-German sentence pairs into the corresponding Indonesian and Chinese.

TABLE II. THE RESULTS ON INDONESIAN TO CHINESE SENTENCE PAIRS.

	Models	BLEU4
RNN-based	Text-only	27.48
Transformer-based	Text-only	28.69
Our method	Image-Text-Attention	29.02

Table 2 shows the results of our model and the baseline models text-only NMT base on RNN and Transformer. From Table 2, we can see that Transformer-based NMT model does outperform RNN-based NMT model, and our proposed model has the best result and 0.33 BLEU scores higher (METEOR metric does not support Chinese) than Transformer Text-only NMT. These results again demonstrate the effectiveness of our model.

4) Discussion.

According to the results shown above, we can see the effectiveness and advantages of our model. That means the Image-text attention layer does help the model to consider the relevance between visual and the text representation, to get better semantic information of encoder's output. And the image can provide supplemental information to the model.

But there is still a gap between our model and the model proposed by Helcl et al. [9]. We think it is possible that our MNMT's final sentence representation in the end of encoder layer is more or less affected by image features, while imagination model in the work of Helcl et al. [9] can learn a better textual representation by using visual information, which can help the model improve the translation quality. And the selection of image features may cause the difference.

IV. CASE STUDY

In order to better demonstrate our model, we conducted a case study of the results on the test set. Due to the better understanding of the Chinese, we select the Indonesian to Chinese translation for case study. Figure 4 and Figure 5 are the results of two cases from our model **Image-Text-Attention** and Transformer-based **Text-only NMT**.

From Figure 4, we can see that our Image-Text-Attention model performs much better than that of Text-only NMT model. The phrase “肮脏的(dirty)” is generated in the target sentence in our model while Text-only NMT does not. And the whole translated sentence is more accurately and smoothly. We also can see that the adjective to describe the old man in the image “秃顶的(bald)” is generated which even not appears in the reference sentence which seems that our model consider the visual information as well.

In Figure 5, **Text-only NMT** translates “积木(blocks)” into “灌木丛(bushes)” by mistake. Maybe it is because both phrases has the same character “木” and leads the model to deviate from

the correct result during prediction. But our proposed model translates it correctly, which means that visual information can help the model back on track.



Indonesian source sentence.	Seorang lelaki yang tua dan kurus mengenakan kemeja yang putih dan kotor sedang mengendarai sepeda di jalan.
Chinese reference.	穿着 肮脏的 白色衬衣的一个皮包骨头的老人在街道上骑着自行车。
English sentence	An old skinny man wearing the dirty white shirt riding on a bicycle on the street.
Model	Translation Result
Text-only NMT	一个年长的男人和一个瘦小的家伙穿着一件白色的衬衫，骑着自行车在大街上。(An old man and a skinny guy are wearing a white shirt and riding a bicycle on the street.)
Image-Text-Attention	一个年长， 秃顶的 男人穿着白色和 肮脏的 衬衫骑在街上。(An old, bald man was riding down the street in a white and dirty shirt)

Figure 4. Translations of different models on Case 1.



Indonesian source sentence	Seorang wanita yang lebih tua dan seorang anak kecil dengan kemeja merah muda bermain dengan balok warna-warni.
Chinese reference	一个老妇人和一个穿着粉红色的衬衫的小孩子，玩五颜六色的 积木块 。
English sentence	An older woman and a young child in a pink shirt playing with multicolored blocks.
Model	Translation Result
Text-only NMT	一个年长的女人和一个穿着粉红色衬衫的小孩玩五颜六色的 灌木丛 。(An older woman and a child in a pink shirt played with colorful bushes)
Image-Text-Attention	一个年长的女人和一个穿着粉红色衬衫的男孩正在玩五颜六色的 积木 。(An old woman and a little boy in a pink shirt were playing with colorful blocks)

Figure 5. Translations of different models on Case 2.

¹ <https://translate.google.com/> Translated on March 2, 2019.

V. RELATED WORK

Machine translation has made a great progress in recent years, from statistical methods [23-28] to neural-network based machine translation (NMT) methods. Kalchbrenner et al. [29] proposed a neural machine translation model based on distributed continuous representation. This neural network applies an end-to-end fashion and is an early research in the field of machine translation that propose the concept of NMT. Cho et al. [30] and Sutskever et al. [31] improved it in 2014, which better promoted neural-network based machine translation model. In subsequent studies, attention mechanism is used to NMT and achieve better results. Bahdanau et al. [32] introduced attention mechanism based on the work of Cho et al. [30]. Luong et al. [33] improved Bahdanau et al. [32]'s work and proposed a new local and global attention mechanism for NMT. Besides using RNN to implement seq2seq model, researchers began to use CNN architecture for NMT model [34-35]. Gehring et al. [36] proposed an encoder-decoder architecture totally based on CNN. Vaswani et al. [8] first proposed to use Transformer for seq2seq machine translation. Because Transformer abandoned the traditional RNN structure and only use self-attention for feature extraction, the model achieved good results in text-only NMT.

In recent years, multimodal machine translation has become a hot research topic in machine translation. Vinyals et al. [37] proposed an IDG model, which use pre-trained CNN as encoder of the seq2seq model for image caption task. Huang et al. [3] proposed to use regional and global image features from VGG-19 to be fused into the model by regarding them as pseudo text words. Calixto et al. [5] used the global image features and incorporated them in different ways into the NMT model.

Elliott et al. [7] proposed a new solution called Imagination. It decomposes the MNMT task into two subtasks, one is to train a “imagine decoder” to predict the corresponding visual representation with a margin-based objective, the other task is the regular translation task. Caglayan et al. [6] tried to fuse image features in different ways: (1) compute a new context vector through the regional image features and the target word and concatenate to the original context vector. (2) Modulate the encoder or decoder’s output with global image feature using element-wise multiplication.

Besides RNN-based seq2seq architecture, Helcl et al. [9] used Transformer to build MNMT model. They proposed two ideas, one is to modify the structure of decoder by adding a visual cross-attention layer, the other is that use the imagination [7] method. Grönroos et al. [10] regarded image feature as a pseudo words and use a gating procedure to process the image feature.

Different from the previous Transformer MNMT model, our model is mainly to change the internal structure of the encoder layer. In each encoder layer, an Image-text attention layer is added between the self-attention layer and the feed-forward network layer, capturing the relationship between source words and image. So that the semantic information of important words that is related to image can be enhanced, thus improve the translation quality.

VI. CONCLUSION

This paper extends a Transformer network for multimodal machine translation. We introduced an Image-text attention layer in the end of encoder layer to capture the relationships between source sentence words and the corresponding image by receiving image features as input and constructing Q, K, V to calculate image and text attention like Self-Attention. So that the semantic information of those words that are more related to image could be enhance.

Several experiments are carried on original English to German sentence pairs of Multi30k dataset and Indonesian to Chinese sentence pairs which is manually annotated by human. The results show that compared to the Text-only NMT, our model has a better performance and is comparable to most of the existing work.

In the future, we will try to visualize the attention weights and see how visual information affects the performance of the model. And we will explore some of new fusion ways to incorporate different forms of image features, like global or local image features into our model.

ACKNOWLEDGMENT

This work is supported by National Nature Science Foundation of China (61976062) and the Science and Technology Program of Guangzhou, China (201904010303).

REFERENCES

- [1] T. Baltrusaitis, C. Ahuja, L. Morency, “Multimodal Machine Learning: A Survey and Taxonomy”, in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, pp. 423-443, 2019.
- [2] L. Specia, S. Frank, K. Sima'an, D. Elliott, “A Shared Task on Multimodal Machine Translation and Crosslingual Image Description”, in Proceedings of the First Conference on Machine Translation (WMT), 2016, pp. 543-553.
- [3] P. Huang, F. Liu, S. Shiang, J. Oh, C. Dyer, “Attention-based Multimodal Neural Machine Translation”, in Proceedings of the First Conference on Machine Translation, Berlin, Germany, 2016, pp. 639–645.
- [4] I. Calixto, Q. Liu, N. Campbell, “Doubly-Attentive Decoder for Multimodal Neural Machine Translation”, in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017, pp. 1913-1924.
- [5] I. Calixto, Q. Liu, N. Campbell, “Incorporating Global Visual Features into Attention-Based Neural Machine Translation”, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2017, pp. 992-1003.
- [6] O. Caglayan, W. Aransa, A. Bardet, M. Garcia-Martinez, F. Bougares, L. Barrault, “LIUM-CVC Submissions for WMT17 Multimodal Translation Task”, in Proceedings of the Conference on Machine Translation(WMT), 2017, pp. 432-439.
- [7] D. Elliott, A. Kadar, “Imagination Improves Multimodal Translation”, in Proceedings of the 8th International Joint Conference on Natural Language Processing, 2017, pp. 130-141.
- [8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, I. Polosukhin, “Attention Is All You Need”, in 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 2017, pp. 71-81.
- [9] J. Helcl, J. Libovicky, D. Varis, “CUNI System for the WMT18 Multimodal Translation Task”, in Proceedings ofthe Third Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, 2018, pp. 622-629.
- [10] S. Grönroos, B. Huet, M. Kurimo, J. Laaksonen, B. Merialdo, P. Pham, M. Sjöberg, U. Sulubacak, J. Tiedemann, R. Troncy, R. Vazquez, “The MeMAD Submission to the WMT18 Multimodal”, in Proceedings of the Third Conference on Machine Translation (WMT) Volume 2: Shared Task Papers, 2018, pp. 609-617.

- [11] K. Simonyan, A. Zisserman, "Very Deep Convolutional Networks For Large-Scale Image Recognition", in International Conference on Learning Representations (ICLR), 2015, pp. 1-14.
- [12] L. Ba, R. Kiros, G. Hinton, "Layer Normalization", arXiv, vol. abs/1607.06450, 2016.
- [13] D. Elliott, S. Frank, K. Sima'an, L. Specia, "Multi30K: Multilingual English-German Image Descriptions", in Proceedings of the 5th Workshop on Vision and Language, 2016, pp. 70-74.
- [14] P. Young, A. Lai, M. Hodosh, J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions", in Transactions of the Association for Computational Linguistics, 2014, pp. 67-78.
- [15] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation", in Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, 2007, pp. 177-180.
- [16] R. Sennrich, B. Haddow, A. Birch, "Neural Machine Translation of Rare Words with Subword Units", in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 2016, pp. 1715-1725.
- [17] K. Papineni, S. Roukos, T. Ward, W. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation", in Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 2002, pp. 311-318.
- [18] M. Denkowski, A. Lavie, "Meteor Universal: Language Specific Translation Evaluation for Any Target Language", Proceedings of the Ninth Workshop on Statistical Machine Translation, 2014, pp. 376-380.
- [19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg · L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge", in International Journal of Computer Vision, 2015, pp. 211-252.
- [20] D. P. Kingma, J. L. Ba, "Adam: A Method for Stochastic Optimization", in International Conference on Learning Representations (ICLR), 2015.
- [21] S. Ren, K. He, R. Girshick, J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks", in Advances in Neural Information Processing Systems (NIPS), 2015, pp. 91-99.
- [22] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition", in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
- [23] P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, P. S. Roossin, "A Statistical Approach to Machine Translation", Computational Linguistics, vol. 16, pp. 79-85, 1990.
- [24] P. F. Brown, V. J. Della Pietra, S. A. Della Pietra, R. L. Mercer, "The Mathematics of Statistical Machine Translation : Parameter Estimation", Computational Linguistics - Special issue on using large corpora: II, vol. 19, pp. 263-311, 1993.
- [25] F. Josef Och, H. Ney "Discriminative Training and Maximum Entropy Models for Statistical Machine Translation", in Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), 2002, pp. 295-302.
- [26] Y. Liu, K. Wang, C. Zong, K. Su, "A unified framework and models for integrating translation memory into phrase-based statistical machine translation", Computer Speech & Language, vol. 55, pp. 176-206, 2019.
- [27] J. Zhang, C. Zong, "Learning a Phrase-based Translation Model from Monolingual Data with Application to Domain Adaptation", in Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, 2013, pp. 1425-1434.
- [28] Z. Tu, Y. Liu, Y. Hwang, Q. Liu, S. Lin, "Dependency Forest for Statistical Machine Translation", in Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), 2010, pp. 1092-1100.
- [29] N. Kalchbrenner, P. Blunsom, "Recurrent Continuous Translation Models", in Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2013, pp. 1700-1709.
- [30] K. Cho, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation", in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1724-1734.
- [31] I. Sutskever, O. Vinyals, Q. V. Le, "Sequence to Sequence Learning with Neural Networks", in Advances in Neural Information Processing Systems, 2014, pp. 3104-3112.
- [32] D. Bahdanau, K. Cho, Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate", in International Conference on Learning Representations (ICLR), 2015.
- [33] M. Luong, H. Pham, C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation", in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2015, pp. 1412-1421
- [34] J. Bradbury, S. Merity, C. Xiong, R. Socher, "Quasi-Recurrent Neural Networks", in International Conference on Learning Representations (ICLR), 2017.
- [35] N. Kalchbrenner, L. Espeholt, K. Simonyan, A. van den Oord, A. Graves, K. Kavukcuoglu, "Neural Machine Translation in Linear Time", arXiv, vol. 1610.10099, 2017.
- [36] J. Gehring, M. Auli, D. Grangier, D. Yarats, Y. N. Dauphin, "Convolutional Sequence to Sequence Learning", in Proceeding ICML'17 Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 1243-1252.
- [37] O. Vinyals, A. Toshev, S. Bengio, D. Erhan, "Show and Tell: A Neural Image Caption Generator", in 2015 IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, pp. 3156-3164.

Speech Command Classification System for Sinhala Language based on Automatic Speech Recognition

Thilini Dinushika, Lakshika Kavmini, Pamoda Abeyawardhana, Uthayasanker Thayavivam and Sanath Jayasena

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

Email: dinushiranagalage.15@cse.mrt.ac.lk

Abstract—Conversational Artificial Intelligence is revolutionizing the world with its power of converting the conventional computer to a human-like-computer. Exploiting the speaker's intention is one of the major aspects in the field of conversational Artificial Intelligence. A significant challenge that hinders the effectiveness of identifying the speaker's intention is the lack of language resources. To address this issue, we present a domain-specific speech command classification system for Sinhala, a low-resourced language. It accomplishes intent detection for the spoken Sinhala language using Automatic Speech Recognition and Natural Language Understanding. The proposed system can be effectively utilized in value-added applications such as Sinhala speech dialog systems. The system consists of an Automatic Speech Recognition engine to convert continuous natural human voice in Sinhala language to its textual representation and a text classifier to accurately understand the user intention. We also present a novel dataset for this task, 4.15 hours of Sinhala speech corpus in the banking domain. Our new Sinhala speech command classification system provides an accuracy of 89.7% in predicting the intent of an utterance. It outperforms the state-of-the-art direct speech-to-intent classification systems developed for the Sinhala language. Moreover, the Automatic Speech Recognition engine shows the Word Error Rate as 12.04% and the Sentence Error Rate as 21.56%. In addition, our experiments provide useful insights on speech-to-intent classification to researchers in low resource spoken language understanding.

Keywords: Sinhala Speech Command Classification, Automatic Speech Recognition, Intent Classification

I. INTRODUCTION

Recent advances in conversational Artificial Intelligence (AI) have resulted in conversation-based applications with a wide range of supported platforms. Google Assistant [1] and Amazon Alexa [2] are two such prominent commercial conversational agents that assist in voice-based control ranging from smartphones to home automation. Speech command classification, also known as *intent classification*, is a key research area in the field of conversational AI. Yet obtaining state-of-the-art results in classifying free-form speech commands of low-resourced languages is challenging [3].

Sinhala, an Indo-Aryan language and the official language of Sri Lanka is recognized as a low-resourced language [4]. Developing an accurate Sinhala speech command classification system can be considered as an initiative to promote Sinhala community to reach the digitized world through their native spoken language. But the lack of resources for Sinhala language hinders the development of accurate models for intent classification.

The combination of Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU) is known for its high level of accuracy in speech command classification of resource full languages. In contrast, Buddhika et al. [b3] and Karunanayake et al. [5] claim that the resource limitation in Sinhala causes suboptimal performance in both the ASR and NLU. Hence, they have proposed a direct classification approach for Sinhala speech-to-intent. To the best of our knowledge, these were the only prior efforts of Sinhala speech to intent mapping. Other Sinhala speech related efforts have focused only on speech-to-text conversion, speech classification or speech clustering.

In this paper, we present the first domain-specific speech command classification system for Sinhala language using ASR and text classification. It classifies banking domain related free-form Sinhala speech commands to their respective intents. Our novel approach aims at exploiting a fine-tuned ASR to gain better accuracy in classification and outperforms the state-of-the-art direct speech-to-intent mapping. Further, we analyze the relevant gain in using a fine-tuned ASR as opposed to the direct classification approach and associated efforts. Though we focus the Sinhala language in this research, the insights presented through this analysis are applicable to other low-resourced language research for developing speech-to-intent systems efficiently and effectively.

The contributions of this paper are: 1. We present a new Sinhala speech corpus in the banking domain 2. We present a novel continuous speech recognition system for Sinhala using Gaussian Mixture Model (GMM) and Hidden Markov Model (HMM) 3. We demonstrate our Sinhala speech command classification system based on ASR and evaluate its performance.

The rest of the paper is organized as follows. Section two presents the related work, and section three includes the background. Section four and five present the dataset and the methodology, respectively. Section six presents our experiments and section seven contains a comparative discussion about the results. Finally, section eight carries our conclusion and future work.

II. RELATED WORK

Buddhika et al. [3] have presented a Sinhala speech command classification system which uses a direct speech-to-intent mapping without an intermediary (e.g., text) representation.

They have implemented a classification algorithm based on Feed Forward Neural Networks to classify Sinhala speech commands. Mel-frequency Cepstral Coefficients (MFCC) have been extracted from speech signals as feature vectors. Using only 10 hours of speech data, they have achieved a classification accuracy of 74%. In addition, Karunananayake et al. [5] have extended the same direct classification approach together with transfer learning to classify domain specific free-form Sinhala and Tamil speech commands. The system has been developed by utilizing a character probability map from an ASR model trained on the English language and has achieved a reasonable accuracy for both Sinhala and Tamil data sets. In contrast to this, our effort employs an intermediary text representation to outperform this direct approach of speech command classification.

Most previous research on speech command classification of both popular and low-resourced spoken languages have been dominated by the combination of ASR and NLU components. An ASR module converts the speech into a text representation and this text is fed to a text classifier in order to understand the intention of the speaker [6] [7]. As shown by Rao et al. [6], this cascading approach can be effectively utilized for intent classification by developing high accurate, fine-tuned ASR systems. Hence, our research focuses on evaluating the applicability of ASR and NLU techniques for Sinhala speech command classification.

The advancements in ASR techniques have shown proven performance in recognizing the speech of several languages. A review on applying HMM for speech recognition [8] has suggested that HMM is flexible and easily implemented for speech recognition of any language due to its inherent mathematical framework. Thus, several speech recognition toolkits that support HMM are widely available [9] [10]. Zissman et al. [11] have presented the combination of GMM together with HMM as an approach to further enhance the accuracy of speech recognition.

There are few prior researches on Sinhala ASR systems implemented using HMMs. The Interactive Voice Response (IVR) system presented by Manamperi et al. [12] shows a Word Error Rate (WER) of 11.2% for digit recognition and a Sentence Error Rate (SER) of 5.7% for song recognition. Also, Nadungodage et al. [13] have achieved an accuracy of 75.74% in recognizing continuous Sinhala speech commands using a data set collected from the voice of a single speaker. In addition, [14] [15] can be identified as successful attempts to recognize isolated Sinhala words with reasonable accuracy. We refer these approaches to build the language model and phoneme dictionary of our ASR system. In addition, ASR systems [16] and [17], for Bengali and Oriya languages, respectively, similar languages to the spoken Sinhala language, can be taken as reference models to develop a fine-tuned ASR system for Sinhala.

III. BACKGROUND

This section carries a brief overview of the major techniques we employed to develop the ASR engine and the text classi-

fier of the proposed Sinhala speech command classification system.

A. MFCCs for feature extraction

This is a method of extracting frequency information in speech signals and converting them into coefficients. Since MFCCs simulate the properties of human auditory system, they are widely applied in speech processing [18].

B. GMM-HMM for acoustic modeling

HMM is a statistical model to compute the probability of a sequence of feature vector observations based on some sequence of hidden state transitions. GMM is combined with HMM to model the distribution of real-valued feature vectors corresponding to each HMM state. This combination of GMM-HMM is widely adopted for acoustic modeling in ASR systems [9].

C. N-gram model

The N-gram model implementation predicts the posterior probability of observing a word (W_n) in a sequence of words (i.e a sentence), given that the words W_1, W_2, \dots, W_{n-1} are preceding in the sequence i.e. $P(W_n|W_1\dots W_{n-1})$. (1) calculates this posterior probability using the word counts in a given language corpus.

$$P(W_n|W_1\dots W_{n-1}) = \frac{\text{count}(W_1, \dots, W_{n-1}, W_n)}{\text{count}(W_1, \dots, W_{n-1})} \quad (1)$$

D. Support Vector Machines

Support Vector Machines (SVM) is a binary classification algorithm that determines the decision boundary between feature vectors of two classes. It can be scaled well for multi-class classification and can be generalized well in high dimensional feature spaces. SVM is recognized as an easy-to-use and robust technique for text classification [19].

IV. SINHALA SPEECH CORPUS

The scope of existing Sinhala speech corpora are not extensible to model a conversation between a human and an agent. Thus, a novel Sinhala speech corpus is built with the intention of developing a Sinhala speech dialog system. We selected the banking domain, analyzed few conversations between a customer and a customer service assistant of a bank during the process of opening a new bank account. We identified 14 basic intentions a customer would express during this common conversation.

First, we used a crowdsourcing approach to identify different inflections on how each of the intents would be uttered in spoken Sinhala language. A Google form including the 14 predefined intents was distributed among 130 participants covering different age groups. We requested them to provide alternative ways in which people express each of these intents in spoken Sinhala language. In addition, participants were selected to capture the different dialects of the Sinhala language spoken in different regions of the country.

TABLE I: Few inflections under a sample intent

Intent		Different inflections (In English transliteration)
Sinhala intent in English transliteration	Meaning	
nawa giNumak wiwurta kiriema	Request to open a new bank ac- count	maTa nawa giNumak wiwurta karanna ona
		maTa alut giNumak wiwurta karanna ona
		giNumak arinna ona

The data was analyzed and a finalized set of inflections corresponding to each intent was created with the help of few language experts. Table 1 includes an example of few inflections identified under the intent - “Request to open a new bank account” in English transliteration.

Voicer, a web/smartphone based crowdsourcing tool presented in [20]: was used to collect speech samples. The tool was re-configured to capture balanced amounts of speech clips for each intent. Multiple users can simultaneously access the tool and record their voice by uttering inflection commands prompted by the tool. The data collection process was conducted under uncontrolled environmental conditions.

The data was collected from 120 speakers representing 60% males and 40% females. 30% of total speakers were university students and the rest from the general community within the age group of 25 to 60 years. The average length of an individual recording ranges from one to three seconds.

Using Voicer, we collected a total of 9650 speech clips for all infections under the 14 intents. These speech clips were validated manually and subjected to noise removal. After removing all flawed clips with over recordings, halfway-stopped, and high noise profiles, 8977 speech clips were shortlisted to build the corpus. The final Sinhala speech corpus was 4.15 hours long. The corpus was divided into training and testing set with 80%, 20% ratio respectively.

V. METHODOLOGY

The proposed Sinhala speech command classification system is designed with two main subcomponents: a domain specific ASR engine and a text classifier. The ASR engine

is developed using a combination of GMM-HMM to convert continuous Sinhala speech into text. The text classifier is modeled using SVM and it predicts the intent of the text-output generated by the ASR. The high-level architecture of the system is depicted in Fig.1. The rest of the section describes the implementation and training process of each component.

A. ASR engine

The ASR engine is developed using a statistical approach as a combination of three basic models: acoustic model, language model and the dictionary model. A 3.32 hours worth speech corpus is used to train the acoustic model. During the training, MFCCs are extracted as the acoustic feature vectors in order to characterize speech signals. The following subsections describe the process of building the aforementioned models during the training phase.

1) *Acoustic model*: The acoustic model of the ASR is designed to capture important attributes from the extracted feature vectors of speech signals. To build the acoustic model, the system is fed with audio clips and their respective text transcriptions. The GMM-HMM combination is used to capture the sequence of phones for acoustic modeling. The procedure followed to build the acoustic model is illustrated in Fig. 2.

During the acoustic model training, separate HMM models are trained for each phoneme (smallest sub-word unit in speech) identified within the speech corpus. The topology of each HMM model is assigned with three hidden states, which the first state for the transition into the phoneme, second for the middle part and the third state for the transition out of the phoneme. This phoneme-based HMM modeling is known as monophone training. The monophone models do not capture any contextual information variants based on preceding or following phonemes.

Next, these monophone models are chained together based on the phoneme constitution of words in order to form new models to recognize words. These context-dependent models which are referred to as triphones, reflect the contextual variations in phoneme occurrences within a speech wave. Finally, a network of HMM models is trained by concatenating these matching triphone models in order to recognize words in continuous speech.

For every state in an HMM, a GMM model is trained. This probabilistic model is built using the Expectation-

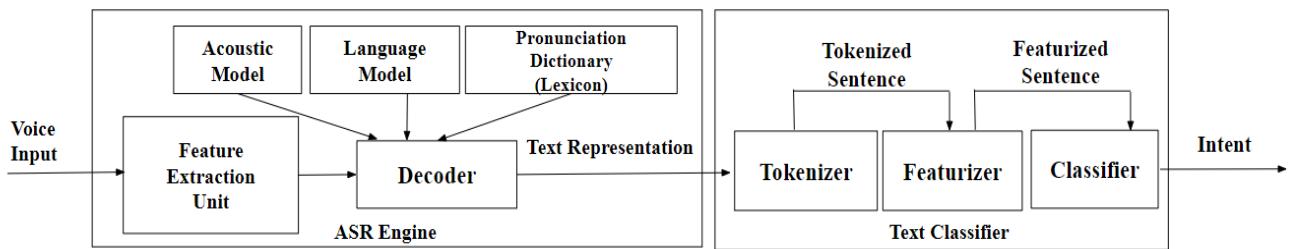


Fig. 1: Architecture of the speech command classification system

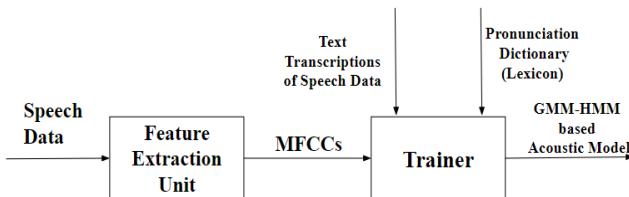


Fig. 2: The process of training the acoustic model

Maximization algorithm. These GMM models are responsible for modeling the emission probability of each state, the probability of each HMM state's association to a speech feature distribution.

2) *Language model*: The language model essentially captures the grammar of the language. Since our ASR system is based on a limited domain of vocabulary, the N-gram model is chosen instead of hardcoded grammar rules. To train the language model, we have implemented 3-Gram model with the help of SRI Language Modeling Toolkit (SRILM).

3) *Dictionary model*: A phoneme is the smallest unit of speech which differentiates a word from another. The phonetic dictionary model is created manually by including the phoneme representation of all words of the selected vocabulary. ISO 15919 - Transliteration of Devanagari and related Indic scripts into Latin characters [21] was referred to construct the phonetic representation of words.

The Sinhala language has 47 phonemes comprised of 14 vowels and 33 consonants. Our domain specific system based on the banking domain has 96 Sinhala words in its dictionary.

B. Text classifier

The textual representation of user utterance generated by the ASR engine is fed into the text classifier in order to predict the most relevant intent out of the 14 pre-defined intent classes. The Sinhala text corpus comprised of infections under 14 intents which was collected through the crowdsourcing, is used to train the classifier models.

During the training, the sentences in the text data corpus is directly fed to the tokenizer in raw format without subjecting to any text normalization. The tokenizer breaks the sentences into tokens (i.e. words) and passes an array of tokens to the featurizer. The featurization component builds a dictionary of words including all the unique words within the text corpus (in our case 96 different words). It assigns a unique integer to each of the words and computes a score for each word using Term Frequency and Inverse Document Frequency (TF-IDF) weighting scheme. Next, each sentence in the corpus is given a vector representation based on the created dictionary. These labeled feature vectors are utilized to train a multi-class text classifier based on SVM while incorporating 'one vs all' approach [22].

VI. EXPERIMENTS

The decoder of the ASR utilizes the trained acoustic model and the language model to convert a given speech sample

to its text representation. This text-output is fed into the text classifier to predict the intent and evaluate the intent classification accuracy.

The performance of the ASR is evaluated using both the WER and SER. The WER is the number of incorrectly identified words out of the total number of words in the test speech corpus. SER is the number of incorrectly identified sentences out of the total number of sentences in the same test speech corpus.

The domain-specific intent classification system presented by Buddhika et al. [3] is selected as our benchmark. It has shown a 74% of test accuracy for intent detection using its direct approach with 10 hours of speech data. As the final outcome, our Sinhala speech command classification system achieves an overall intent classification accuracy of 89.7% only using a 4.15 hours speech data corpus with its GMM-HMM based acoustic model in the ASR and SVM as the text classifier.

Our ASR engine based on GMM-HMM shows a WER of 12.04% and an SER of 21.56%. In addition, the variation of WER and SER with respect to the duration of training speech data corpus is presented in Table 3.

A comparative analysis is conducted to understand the impact of text classification technique on the overall intent classification accuracy. Accordingly, different text classification models are trained using, 1. Neural Network Embeddings 2. Naive Bayesian 3. Logistic Regression and used in place of the SVM classifier [23] [24]. Table 4 presents the accuracy of the speech command classification system against each of the classifier models.

VII. DISCUSSION

The results show that our proposed methodology of utilizing a fine-tuned ASR module outperforms the state-of-the-art direct speech-to-intent mapping. Hence, we prove that utilizing an ASR and following through an intermediary stage of text representation is a viable solution for the challenge of recognizing low-resourced spoken languages. Moreover, our ASR based speech command classification system shows a significant accuracy even with half-sized data corpus compared to [3].

TABLE II: Variation of WER and SER of the ASR based on the speech corpus size

Percentage from the original data set	No of speech clips	No of hours	WER%	SER%
100%	7232	4.15	12.04	21.56
80%	5743	3.30	13.84	23.96
60%	4331	2.42	14.88	25.33
50%	3612	2.02	15.05	27.50
30%	2165	1.20	17.79	32.08
20%	1435	0.86	18.34	32.30

TABLE III: Speech command classification accuracy with respect to text classification model

Text classification model	SVM	Logistic Regression	Naive Bayesian	Neural Network Embeddings
Accuracy (%)	89.7	88.9	87.2	84.5

Fig. 3 depicts the word and sentence recognition accuracy of the ASR and the accuracy of the intent classification based on the size of the training speech corpus. It is notable that even with 50% of the total corpus, a speech recognition accuracy closer to 80% could be obtained. Further, it shows that the overall accuracy of the speech command classification is following the same accuracy variation pattern compared to the ASR. After exceeding 85% word recognition and 75% sentence recognition accuracy, the intent classification reports a stable accuracy. Hence, we can suggest that the accuracy of the ASR is the major determining factor of the overall performance of the speech command classification system.

The speech command classification accuracy variation with respect to different approaches of text classification shows that SVM outperforms the other three models. It is due to the SVM's ability to perform well even with a small amount of training corpus. The Neural Network Embedding approach shows the least accuracy due to the inherent performance degradation of Neural Networks when a small training corpus is used. It is notable that the speech command classification accuracy values show only a slight variation from each other despite the text classification model used. Thus, we claim that the accuracy of the ASR is the determining factor of the overall performance of the speech command classification system.

In Fig 4, we have plotted the speech command classification accuracy against the word recognition accuracy per utterance; i.e., the fraction of correctly identified words out of the total number of words in an utterance. Once the word recognition accuracy per utterance exceeds 60%, the speech command classification accuracy increases significantly. Typically, an utterance consists of few words that have a higher weight on determining the intent. The rest of the words are commonly found in most utterances. Failing to correctly identify those significant words in an utterance has resulted to drop the speech command classification accuracy even though more than 70% of words are correctly identified. The same word is pronounced in different styles based on the context. This also can be presented as a reason for the variation of classification accuracy based on the word recognition rate per utterance.

Since our approach involves intermediary speech-to-text conversion, the WER and SER of the ASR are essential to analyze. Table 5 contains a comparison of the obtained performance results with few prior continuous Sinhala speech recognition systems. In addition, the performance comparison with some ASR systems developed for few other low-resourced Indo-Aryan languages is included in Table 6. Both cases show that our approach of using GMM-HMM based

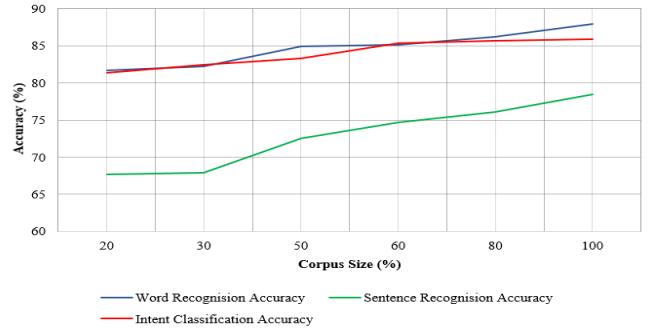


Fig. 3: Accuracy variations based on the speech corpus size

TABLE IV: Comparison with prior continuous Sinhala speech recognition systems

Speech recognition system	ASR technique and data set	WER %	SER %
Manamperi et al. [12]	HMM, 50 Sinhala songs, 2.63 hours, 85 speakers	-	5.7
Nadungodage et al. [13]	HMM, 983 distinct continuous Sinhala sentences, single speaker	3.86	24.26
Our system	HMM-GMM, Free-form Sinhala commands, 4.15 hours, 120 speakers	12.04	21.56

acoustic modeling in the ASR contributes to obtain a comparatively good performance in continuous speech recognition for Sinhala spoken language.

The IVR system presented by Manamperi et al. [12] shows a higher sentence recognition accuracy compared to our ASR system. The reason behind this variation can be explained as our speech corpus based on banking domain has longer utterances compared to the song recognition data set used by them. The continuous Sinhala speech recognizer by Nadungodage et al. [13] has achieved a very small WER as a result of its speech corpus collected only from a single speaker. The higher accuracy rates of the Bengali language ASR system by Zinnat et al. [16] is due to the utilization of a new feature named Local Features together with MFCCs. The ASR system presented by Kumar et al. [26] has used a training speech corpus collected only from 12 speakers. It can be presented as a reason for its lesser SER compared to our system.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we presented our domain-specific speech command classification system for Sinhala language. It detects the intent of a Sinhala utterance using ASR followed by text classification. The performance comparison of our ASR with prior ASRs developed for Sinhala language revealed that our GMM-HMM based ASR has promising results with a higher rate of accuracy. Our speech command classification system showed an accuracy of 89.7% in detecting the intent of a Sinhala utterance while outperforming the previous direct

TABLE V: Comparison with ASRs developed for other Indo-Aryan languages

Speech recognition system	Language	ASR technique	WER%	SER%
Zinnat et al. [16]	Bengali	HMM	7.50	8.50
Chowdhury et al. [25]	Bengali	HMM	28.62	-
Kumar et al. [26]	Hindi	HMM	12.99	9.07
Mohanty et al. [27]	Oriya	GMM-HMM	-	21.77
Our system	Sinhala	GMM-HMM	12.04	21.56

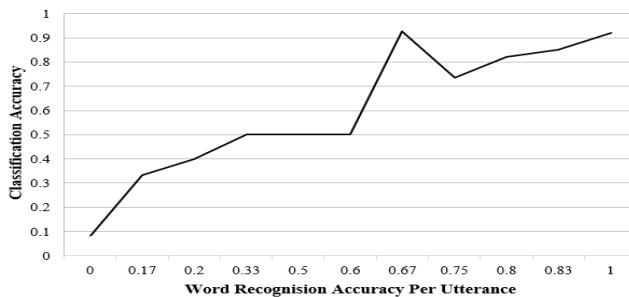


Fig. 4: Variation of speech command classification accuracy based on the word recognition accuracy per utterance

classification approach. Being the first attempt on both ASR and speech command classification for Sinhala; we suggest that our work is a positive approach and a viable solution to gain a significant improvement in intent detection of free-form Sinhala speech commands. As future work, we will extend this speech command classification system and develop a speech dialog system for the Sinhala language.

ACKNOWLEDGEMENT

The authors of this paper would acknowledge the reviewers for their valuable comments, University of Moratuwa senate research committee grant for supporting this research and all the people who participated in the data collection.

REFERENCES

- [1] "Google Assistant, your own personal Google", Assistant.google.com, 2019. [Online]. Available: <https://assistant.google.com>. [Accessed: 19- Sep- 2019].
- [2] "What is Alexa? Amazon Alexa Official Site", Developer.amazon.com, 2019. [Online]. Available: <https://developer.amazon.com/alexa>. [Accessed: 19- Sep- 2019].
- [3] D. Buddhika, R. Liyadipita, S. Nadeeshan, H. Witharana, S. Jayasena, and U. Thayavaram, "Domain specific intent classification of Sinhala speech data," 2018 International Conference on Asian Language Processing (IALP), Bandung, Indonesia, 2018, pp. 197–202.
- [4] W. S. N. Dilshan, S. Yashothara, R. T. Uthayasan and S. Jayasena, "Linguistic Divergence of Sinhala and Tamil Languages in Machine Translation," 2018 International Conference on Asian Language Processing (IALP), Bandung, Indonesia, 2018, pp. 13-18.
- [5] Y. Karunanayake, U. Thayavaram, and S. Ranathunga, "Transfer learning based free-form speech command classification for low-resource languages," in Proc. of ACL 2019, Student Research Workshop, 2019.
- [6] J. Rao, F. Ture, and J. Lin, "Multi-task learning with Neural Networks for voice query understanding on an entertainment platform," 24th ACM SIGKDD International Conference on Knowledge Discovery & Data, 2018, pp. 636–645.
- [7] S. Yaman, L. Deng, D. Yu, Y. Wang and A. Acero, "An integrative and discriminative technique for spoken utterance classification," IEEE Transactions on Audio, Speech, and Language Processing, vol. 16, no. 6, pp. 1207–1214, Aug. 2008.
- [8] L.R. Rabiner and B. H. Juang, Fundamentals of Speech Recognition, Prentice Hall, 1993.
- [9] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," IEEE 2011 Workshop on Automatic Speech Recognition and Understanding, Dec. 2011, IEEE Signal Processing Society.
- [10] S. Young, G. Evermann, M.J.F. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev and P. Woodland, The HTK book, 2002
- [11] M. A. Zissman, "Automatic language identification using gaussian mixture and hidden Markov models," 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing, Minneapolis, MN, USA, 1993, vol.2, pp. 399–402.
- [12] W. Manamperi, D. Karunathilake, T. Madhushani, N. Galagedara, and D. Dias, "Sinhala speech recognition for interactive voice response systems accessed through mobile phones," 2018 Moratuwa Engineering Research Conference (MERCon). IEEE, 2018, pp. 241–246.
- [13] T. Nadungodage and R. Weerasinghe, "Continuous sinhala speech recognizer," Conference on Human Language Technology for Development, Alexandria, Egypt, 2011, pp. 2–5.
- [14] W. Amarasingha and D. Gamin, "Speaker independent sinhala speech recognition for voice dialling," Advances in ICT for Emerging Regions (ICTer), 2012 International Conference on. IEEE, 2012, pp. 3–6.
- [15] J. Nallathambi, K. Kariyawasam, H. Pullaperuma, D. Vithana, and S. Jayasena, "deBas: a Sinhala Interactive Voice Response (IVR) System," [Online]. Available: <http://dl.lib.mrt.ac.lk/handle/123/8061>. [Accessed: 1st July 2019].
- [16] S. B. Zinnat, R. M. A. Siddique, M. I. Hossain, D. M. Abdullah and M. N. Huda, "Automatic word recognition for Bangla spoken language," 2014 International Conference on Signal Propagation and Computer Technology (ICSPCT 2014), Ajmer, 2014, pp. 470–475.
- [17] B. Karan, J. Sahoo and P. K. Sahu, "Automatic speech recognition based Odia system," 2015 International Conference on Microwave, Optical and Communication Engineering (ICMOCE), Bhubaneswar, 2015, pp. 353–356.
- [18] K. Gupta and D. Gupta, "An analysis on LPC, RASTA and MFCC techniques in Automatic Speech Recognition system," 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence), Noida, 2016, pp. 493–497.
- [19] T. Joachims, "Text categorization with Support Vector Machines: learning with many relevant features," 10th European Conference on Machine Learning (ECML'98), Chemnitz, Germany, 1998, pp. 137–142.
- [20] D. Buddhika, R. Liyadipita, S. Nadeeshan, H. Witharana, S. Jayasena and U. Thayavaram, "Voicer: A crowd sourcing tool for speech data collection," 18th International Conference on Advances in ICT for Emerging Regions (ICTer), 2018, pp. 174–181.
- [21] "ISO 15919", En.wikipedia.org, 2019. [Online]. Available: https://en.wikipedia.org/wiki/ISO_15919. [Accessed: 19- Sep- 2019].
- [22] Y. Ahuja and S. K. Yadav, "Multiclass classification and Support Vector Machine," Global Journal of Computer Science and Technology Interdisciplinary, vol. 12, pp. 14 – 20, 2012.
- [23] M. Y. H. Setyawan, R. M. Awangga and S. R. Efendi, "Comparison Of Multinomial Naive Bayes Algorithm And Logistic Regression For Intent Classification In Chatbot," International Conference on Applied Engineering (ICAЕ), 2018, pp. 1–5.
- [24] L. Wu, A. Fisch, S. Chopra, K. Adams, A. Bordes, and J. Weston, "Starspace: Embed all the things!," arXiv preprint arXiv:1709.03856, 2017.
- [25] S. A. Chowdhury, "Implementation of speech recognition system for Bangla," Ph.D. dissertation, BRAC University, 2010.
- [26] K. Kumar, R. Aggarwal, and A. Jain, "A Hindi speech recognition system for connected words using HTK," Int. Journal of Computational Systems Engineering, vol. 1, pp. 25 – 32, 2012.
- [27] S. Mohanty and B. K. Swain, "Continuous Oriya digit recognition using Bakis Model of HMM," International Journal of Computer Information Systems, vol. 2, no. 1, 2011.

An Enhancement of Malay Social Media Text Normalization for Lexicon-Based Sentiment Analysis

Muhammad Fakhrur Razi Abu Bakar, Norisma Idris, Liyana Shuib

Faculty of Computer Science and IT

University of Malaya

Kuala Lumpur, Malaysia

fakhrurrazi0202@gmail.com, norisma@um.edu.my, liyanashuib@um.edu.my

Abstract—Nowadays, most Malaysians use social media such as Twitter to express their opinions toward any latest issues publicly. However, user individuality and creativity of language create huge volumes of noisy words which become unsuitable as dataset for any Natural Language Processing applications such as sentiment analysis due to the irregularity of the language featured. Thus, it is important to convert these noisy words into their standard forms. Currently, there are limited studies to normalize the noisy words for Malay language. Hence, the aim of this study is to propose an enhancement of Malay social media text normalization for lexicon-based sentiment analysis. This normalizer comprises six main modules: (1) advanced tokenization, (2) Malay/English token detection, (3) lexical rules, (4) noisy token replacement, (5) n-gram, and (6) detokenization. The evaluation has been conducted and the findings show that 83.55% achieved in Precision and 84.61% in Recall.

Keywords—Malay social media text, Twitter, Noisy text, Lexicon-Based, Text Normalization

I. INTRODUCTION

Social Networking Service (SNS) like Twitter is increasingly popular as a medium for Malaysians to communicate or express their opinions overtly. The growing popularity of SNS attracts attention of researchers, especially in Sentiment Analysis (SA). Hybrid, lexicon-based, and machine learning were the three cardinal approaches for SA [1]. Lexicon-based approach is preferred when the English language is not being used because of its flexibility [1]. Based on our analysis, the noisy Malay text is not being handled completely at the pre-processing phase before being processed at the SA phase.

Twitter is considered as one of the top SNS in Malaysia [2]. According to [3], Malaysians write their respond on the SNS without following any abbreviations rules. Based on our analysis on 20k of Twitter messages from Malaysians, most of them use local dialects which give the same meaning but the spelling are totally different from the standard words (e.g. *awat* → *kenapa*), abbreviations where the patterns are not fixed or user generated text (e.g. *dgn*, *dgan* → *dengan*), local trend language where two similar words have totally different meanings compare to the meaning found in the dictionary (e.g. *payung(umbrella)* → *payung(belanja)*), and mix languages which mostly consist of English and Malay (e.g. *i am so lapar*). According to [4], most of the Natural Language Processing (NLP) tools were predominantly trained on formal text. Hence, the existence of the noisy

Malay texts become one of the major challenges in applying SA applications to public responds on SNS [5].

This paper proposed an enhancement of Malay text normalizer for lexicon-based SA. The remainder of this paper is organized as follows: The problem is identified from literature analysis and preliminary studies which is discussed in Section II. In Section III, the analysis on the 20k of Twitter data by Malaysian users is explained. Then the proposed architecture of the Malay Text Normalizer for lexicon-based SA in Section IV is presented. The evaluation of the normalizer is discussed in Section V. Finally, in Section VI we conclude the article and highlight the future works.

II. RELATED WORKS

Based on our analysis, there are 16 pre-processing techniques existed namely case folding, stop word removal, spelling correction, tokenization, stemming, lemmatization, intrinsic words removal, spam removal, characters removal, punctuation marks removal, non-words removal, social media tags removal, repeated characters removal, diacritics removal, symbols removal, and others. Case folding was used by several of the previous works such as [6], [7], [8], [1], and [9]. Case folding technique converts all the characters in a document into the same case, either all upper case or lower case. The second technique is stop words removal where it has been used by [6], [10], [11], [12], [13], [14], [1], [3], and [15] in their works. This technique removes words that carry unimportant meaning with respect to SA. Spelling correction is a technique to correct spelling errors due to abbreviations or typo. This technique was used by [10], [16], [14], and [1] in their works where it can be done by implementing spelling correction algorithm or creating a dictionary for converting the noisy word into its meaningful word. Another technique is tokenization which it helps to simplify the SA process. This technique was used by [10], [11], [13], [14], [1], [17], and [15] in their works. Stemming and lemmatization techniques both have been used by [1] in their study. For the basic removal techniques, [10] has removed intrinsic words in their work. Spam which largely be composed of insignificant words and emoticons were removed by [7] in their work. Other than that, [18] removed characters in their work. Another technique is punctuation mark removal which has been used by [10], [14], and [15] in their works. Non-words have been removed by [10] and [14] in their works. Besides than that, social media tags were removed by [12] and [14] in their works. Reference [12] has also removed repeated characters and diacritics in their works. The last basic removal technique which is

symbols removal has been used by [7], [18], [1], and [3] in their works. Besides the above process, there are other pre-processing techniques used by the previous studies. Reference [19] cleaned their dataset which consisted of unwanted tags to obtain the words including abbreviations. For [9], their dataset was filtered by choosing tweets that were written in English, Malay or Indonesian language only. Other than that, they only chose the tweets which contained only one of the subjects being monitored in their project. To avoid huge numbers of neutral tweets, they filtered out any tweets that did not include any sentiment words which included in their sentiment lexicon. To avoid unreadable symbols and characters, the selected tweets must able to be encoded and decoded in UTF-8. In [6], they merged the word “tidak” with the next word to cater for negative words in the Malay language. Next, for [15], the duplicates that may modify the sentiment analysis’s result were removed. In a study by [8], any words in a tweet that contained “www” or “https://”, “#hashtag”, and “@username” were converted to “URL”, “hashtag”, and “AT_USER” respectively. They also used another technique called trim.

To the best of authors’ knowledge, there are only a few studies on normalizer for noisy Malay text. A study by [20] developed a dictionary-based system which also known as NoisyTerm to normalize Malaysian micro-texts. NoisyTerm has an ambiguity problem as highlighted by [5] in their study. Other than that, [20] removed directly the correct spelling in English or Malay language at the beginning of the process which will affect the latest Malay trend or noisy language actual meaning (e.g. *kite/saya/kita*) → *kite/layang-layang*, *payung/belanja* → *payung/umbrella*). In another study by [21] they proposed an approach for correcting the noisy Malay words without any interaction from the user. In addition, there is an ambiguity problem occur in their misspelled word dictionary which is the same as [20]. Finally, they removed symbols at the beginning of their approach which several of them will have a value in the latest informal patterns of the noisy Malay text. A corpus-driven analysis approach for normalizing Malay Twitter messages has been proposed by [5]. Reference [5] tagged any character excluding digit 2 and alphabetic characters as a proper noun which cannot be changed anymore at the upcoming steps. This will affect some of the latest informal patterns of the noisy text actual meaning and spelling (e.g. \$ → *duit/money*). Other than that, the in-vocabulary words detection caused a loss of the latest Malay trend & noisy language actual meaning.

III. ANALYSIS OF TWITTER DATA

The Malay text normalizer is designed based on the Twitter analysis results. 20k Twitter messages by Malaysian users were analysed with guidance from three linguists from Faculty of Academic Studies of Malay, University of Malaya, Malaysia. The Twitter messages were collected based on location set to Malaysia using web scripting. The analysis process was done by reading all the Twitter messages one by one and classified the words into categories manually. Table I shows the summary of the Twitter analysis results where these datasets can be grouped into six categories. Based on the results, we found

that most Malaysians used mix languages which consist of English and Malay in their tweets. Malay category refers to the whole tweet using a slang, standard and noisy Malay text. English category refers to the whole tweet using a slang, standard and noisy English text. Mix category refers to the mix of Malay and English languages, and some tweets consist of dialect in between. Dialect category refers to a minimum use of any Malay dialect in a standard Malay language tweet. Others refer to an identification name (@malaysiatravel) or any word starts with ‘#’ symbol. The final category, decline, refers to other than the above mentioned.

The general analysis for the mix category is shown in table II where most of the messages used more Malay terms than English. Table III shows the analysis of formal and noisy words for the tweets that used mix languages. Table IV shows the analysis of formal and noisy words for the English category. The analysis of the formal and noisy words for the Malay category is presented in table V. Based on this analysis, we conclude that there is a high volume of noisy text occurred in Malay Twitter messages. The existence of the noisy text has become the biggest obstacle to applying text-mining. Thus, we propose an architecture of Malay text normalizer to clean the noisy text, before SA can be applied onto it.

TABLE I. OVERALL ANALYSIS RESULTS

Category	Frequency	Percentage
Malay	5479	27.40%
English	4246	21.23%
Mix	6068	30.34%
Dialect	479	2.40%
Others	33	0.17%
Decline	3695	18.48%
Total	20000	100%

TABLE II. FREQUENCY AND PERCENTAGE OF DOMINANT LANGUAGE

Category	Frequency	Percentage
Malay (Dominant)	5110	84.21%
English (Dominant)	958	15.79%
Total	6068	100%

TABLE III. MIX ANALYSIS RESULTS

Category	Frequency	Percentage
Formal	463	7.63%
Noisy	5605	92.37%
Total	6068	100%

TABLE IV. ENGLISH ANALYSIS RESULTS

Category	Frequency	Percentage
Formal	3925	92.44%
Noisy	321	7.56%
Total	4246	100%

TABLE V. MALAY ANALYSIS RESULTS

Category	Frequency	Percentage
Formal	1292	23.58%
Noisy	4187	76.42%
Total	5479	100%

IV. PROPOSED TEXT NORMALIZATION

The overall process flow for SA is presented in Fig. 1 where in this work, we only focus on the Normalizer module which will be used to normalize the noisy Malay text found in the Twitter messages before proceeding to Polar Word Identification module. The overall process flow is from [3] where the data tokenization and data pre-processing module is replaced with our Normalizer module. Fig. 2 shows the architecture of the Normalizer module. In this overall process flow, tweets by Malaysian users will be used as the dataset. After going through the normalizer module, the polar words exist from the dataset will be identified by mapping them against the Malay sentiment lexicon. Lastly, the valence shifter will be handled, and the sentiment value will be identified.

Several modules and rules from this architecture were edited from [5], [20], and [21]. During the tokenization module process, all capital letters will be converted to small letters since capital letter is not having any orthographic value in noisy text [5]. Next, “\n”, “#”, and “http or link” will be removed after every extra blank have been converted to single blank. Any special character and mix special characters which do not have any meaning will also be removed. Any word which start with “@” or ended with “” or “s” will be tagged as Proper Noun Token (PNT). After that, any word that exist inside a Trend dictionary and Dialect dictionary will be tagged as Trend Dictionary Token (TDT) and Dialect Dictionary Token (DDT) respectively. There are another 19 Punctuation List (PL) tokens exist in this study for reducing the complexity and fasten the process of the proposed architecture. Finally, all the white spaces will be converted to new lines and every word will be tokenized.

After the tokenization process is completed, any token that exist in Global Malay/English dictionary will be tagged with In-Vocabulary Token (IVT). To fasten the process, only token that has been tagged with PL tokens or has not been tagged yet are allowed to go through this module. There is a condition where some of the noisy Malay tokens which have the correct spelling will have different meaning compare to the meaning found in the dictionary (e.g. *payung/noisy* → *belanja*, *payung/dictionary* → *umbrella*). Thus, this issue has been solved by tagging them with TDT before entering this module. There is also another condition for a tweet that used mix languages (English and noisy Malay text) where

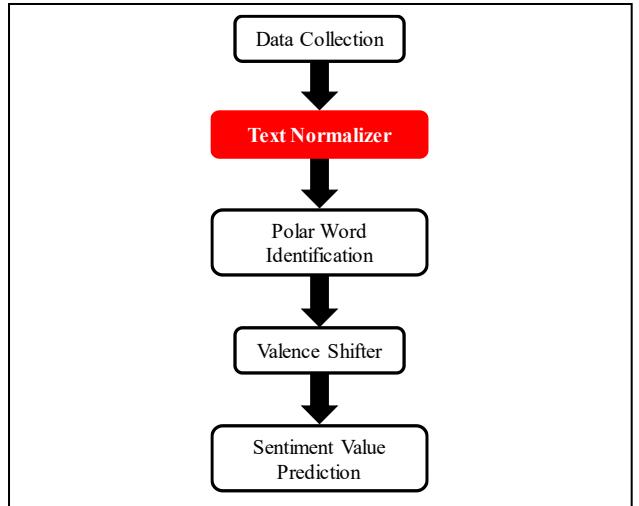


Figure 1. Overall Process Flow

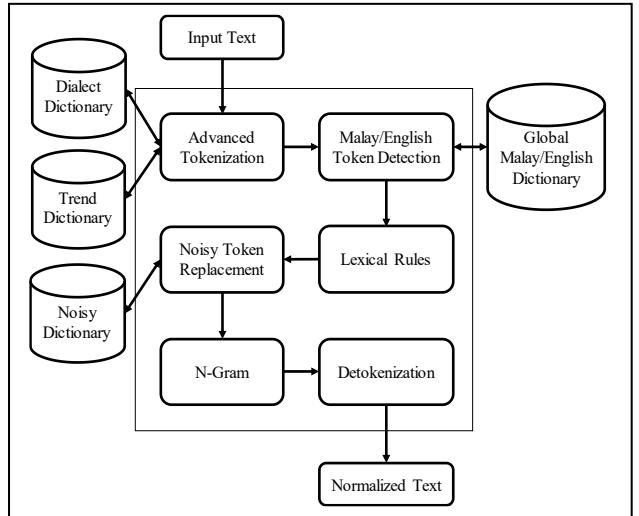


Figure 2. Enhancement of Malay Text Normalization

the noisy token “kite” can be referred to “saya/kita” or “layang-layang”. Based on the our Twitter analysis results, 84.21% of Malaysians used more noisy or standard Malay language in a single tweet compare to noisy or standard English language when they tweeted using mix languages. Hence, the token “kite” only will be tagged with IVT if the dominant language inside the whole tweet is English.

Lexical rules are the most complex and important module in this normalizer where it handles automatically most of the latest informal patterns of noisy Malay text as shown in Fig. 3. This module consists of 12 sub-modules namely Repeated Letter Elimination, Repetitive Words, Rules of X, Vowel Rules, Consonant Rules, Prefix Rules, CCV Rules, CC (1) Rules, CC (2) Rules, DPG Rules, RUYN Rules, and White List Rules. The lexical rules have two main functions namely tagging and process. During the tagging function, every token will go through the sub-modules one by one by following the order where any token that matches with one of the sub-modules will be tagged with related token. During the process function, the tagged token from the tagging function will be solved using related rules. During this process in most of the sub-

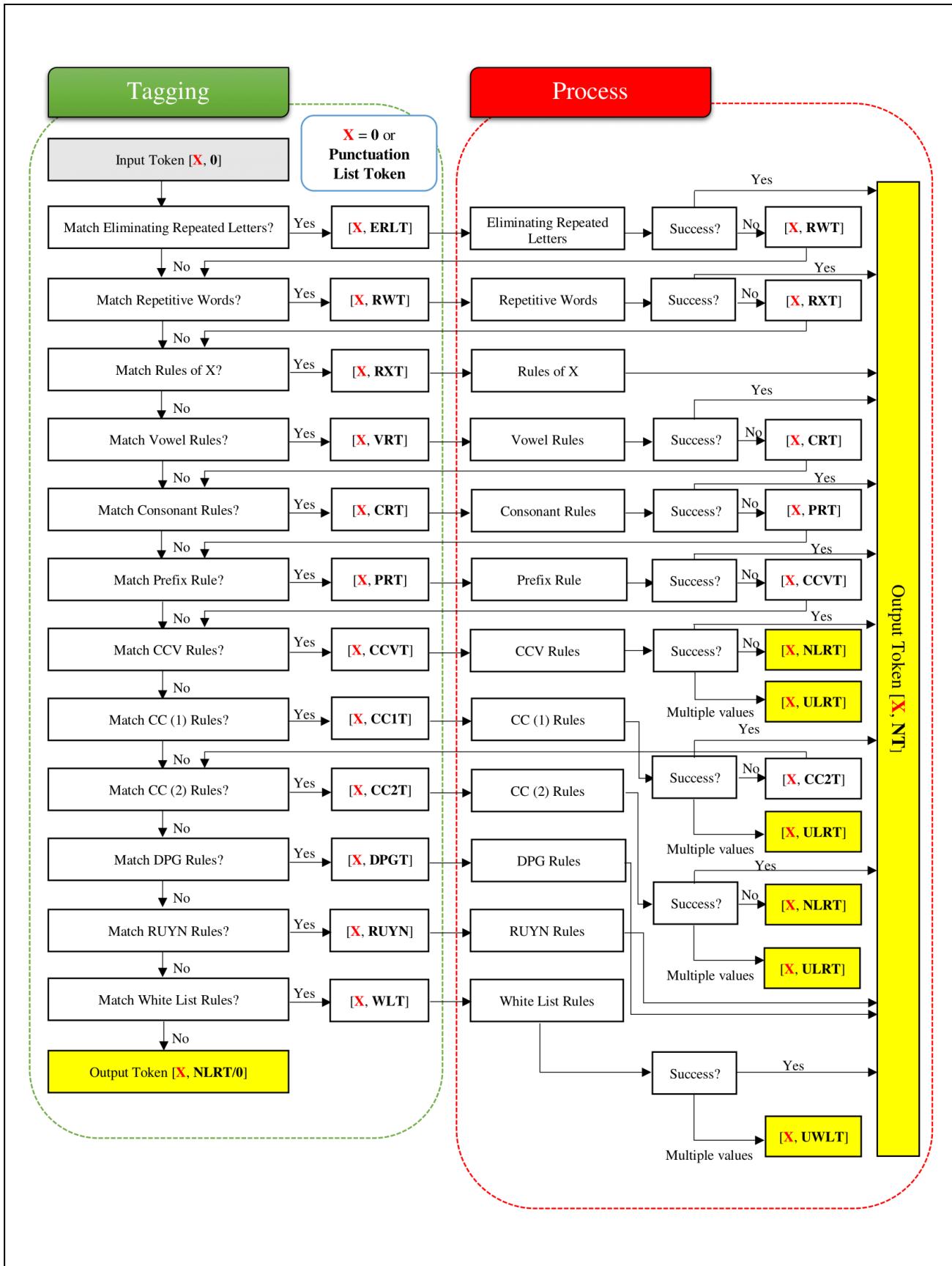


Figure 3. Lexical Rules Flow

modules, the normalized token will be checked automatically by using a dictionary which contains Malay words namely Global Malay dictionary. If the normalized token does not exist inside the Global Malay dictionary, it will be tagged with the next sub-module's token, otherwise the normalization of that token is considered as success. Other than that, stemming function also has been used to check the results in both Repeated Letter Elimination and Prefix Rules.

After the lexical rules process is completed, the token will next go to the noisy token replacement module. This module only allowed token which has been tag with Noisy Lexical Rules Token (NLRT) or has not been tag yet to fasten the process. If the token exists inside the noisy dictionary with more than one meaning, the token will be tagged with Noisy Dictionary Token (NDT), otherwise the token will be tagged with Normalized Token (NT). The N-gram module is provided to solve any of the ambiguity issues. Finally, all the tokens will be detokenized, and all the punctuation marks will be placed back by following the tagged PL tokens.

V. EVALUATION

In evaluation stage, 100 Malaysians tweets have been used as a preliminary result in this study. Since 20k Twitter messages have been used to design the proposed architecture, another randomly 100 Malaysians Twitter messages have been collected for evaluation. The evaluation is conducted by comparing the normalized word produced by the proposed architecture with the normalized word produced by three linguists from Faculty of Academic Studies of Malay, University of Malaya, Malaysia. The result shows the proposed architecture produced a promising result with 83.55% Precision and 84.61% Recall.

VI. CONCLUSION

This research paper explained briefly the conceptual theory of the proposed architecture since it is still an ongoing research. The aim of this research work is to produce a Malay text normalizer for lexicon-based SA. To develop the architecture of the normalizer, the dataset needs to be analysed thoroughly. Thus, we collected 20K Twitter data from Malaysian users and analysed the data to classify the words into different categories. From the results of the analysis, we proposed an architecture of the Malay Text Normalizer which comprises 4 dictionaries which are dialect dictionary, trend dictionary, global Malay/English dictionary and noisy dictionary. Besides than the 4 dictionaries, there are also 6 main processes proposed in the architecture which are advanced tokenization, Malay/English token detection, lexical rules, noisy token replacement, n-gram, and detokenization. To gauge the performance of the Malay Text Normalizer, the proposed architecture has been developed and evaluated using new Twitter dataset. The findings show that the proposed architecture achieved 83.55% in Precision and 84.61% in recall.

ACKNOWLEDGMENT

This research work is supported by University of Malaya (Faculty Research Grant: GPF007D-2018) and Cloud Connect Sdn. Bhd.

REFERENCES

- [1] M. H. A. Hijazi, L. Libin, R. Alfred, and F. Coenen, "Bias aware lexicon-based Sentiment Analysis of Malay dialect on social media data: A study on the Sabah Language," *Proceeding - 2016 2nd Int. Conf. Sci. Inf. Technol. ICSITech 2016 Inf. Sci. Green Soc. Environ.*, pp. 356–361, 2017.
- [2] N. A. Muhamad, M. A. Saloot, and N. Idris, "Proposal: A Hybrid Dictionary Modelling Approach for Malay Tweet Normalization," in *Journal of Physics: Conference Series*, 2017, vol. 806, no. 1.
- [3] K. Chekima and R. Alfred, "Sentiment Analysis of Malay Social Media Text", *Computational Science and Technology*, vol. 488, pp. 205–219, 2018.
- [4] T. Baldwin and L. Yunyao, "An In-depth Analysis of the Effect of Text Normalization in Social Media," *NAACL HLT 2015 - 2015 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. Proc. Conf.*, pp. 420–429, 2015.
- [5] M. A. Saloot, N. Idris, and R. Mahmud, "An architecture for Malay Tweet normalization," *Inf. Process. Manag.*, vol. 50, no. 5, pp. 621–633, 2014.
- [6] N. Samsudin, M. Puteh, A. Razak, and M. Zakree, "Immune Based Feature Selection for Opinion Mining," *Proc. World Congr. Eng.*, vol. III, pp. 1520–1525, 2013.
- [7] N. F. Shamsudin, H. Basiron, Z. Saaya, A. F. N. Abdul Rahman, M. H. Zakaria, and N. Hassim, "Sentiment classification of unstructured data using lexical based techniques," *J. Teknol.*, vol. 77, no. 18, pp. 113–120, 2015.
- [8] M. Naim, M. Ibrahim, M. Zaliman, and M. Yusoff, "Twitter Sentiment Classification Using Naïve Bayes Based on Trainer Perception," *IEEE Conf. e-Learning, e-Management e-Services Twitter*, pp. 187–189, 2015.
- [9] Y. F. Tan, H. S. Lam, A. Azlan, and W. K. Soo, "Sentiment analysis for telco popularity on twitter big data using a novel Malaysian dictionary," *Front. Artif. Intell. Appl.*, vol. 282, pp. 112–125, 2016.
- [10] A. Alsaffar and N. Omar, "Study on feature selection and machine learning algorithms for Malay sentiment classification," *Conf. Proc. - 6th Int. Conf. Inf. Technol. Multimed. UNITEN Cultiv. Creat. Enabling Technol. Through Internet Things, ICIMU 2014*, pp. 270–275, 2015.
- [11] A. Alsaffar and N. Omar, "Integrating a Lexicon based approach and K nearest neighbour for Malay sentiment analysis," *J. Comput. Sci.*, vol. 11, no. 4, pp. 639–644, 2015.
- [12] T. Al-Moslmi, S. Gaber, A. Al-Shabi, M. Albared, and N. Omar, "Feature Selection Methods Effects on Machine Learning Approaches in Malay Sentiment Analysis," no. October, pp. 2–5, 2015.
- [13] A. A. Sadanandan *et al.*, "Improving Accuracy in Sentiment Analysis for Malay Language," *Proceeding 4th Int. Conf. Artif. Intell. Comput. Sci.*, no. November, pp. 28–29, 2016.
- [14] M. I. Eshak, R. Ahmad, and A. Sarlan, "A preliminary study on hybrid sentiment model for customer purchase intention analysis in socialcommerce," *2017 IEEE Conf. Big Data Anal. ICBDA 2017*, vol. 2018-Janua, pp. 61–66, 2018.
- [15] A. Al-Saffar, S. Awang, H. Tao, N. Omar, W. Al-Saiagh, and M. Albared, "Malay sentiment analysis based on combined classification approaches and Senti-lexicon algorithm," *PLoS One*, vol. 13, no. 4, pp. 1–18, 2018.
- [16] N. F. Shamsudin, H. Basiron, and Z. Sa'aya, "Lexical based sentiment analysis - Verb, adverb & negation," *J. Telecommun. Electron. Comput. Eng.*, vol. 8, no. 2, pp. 161–166, 2016.
- [17] T. Al-Moslmi, N. Omar, M. Albared, and A. Alshabi, "Enhanced Malay sentiment analysis with an ensemble classification machine learning approach," *Journal of Engineering and Applied Sciences*, vol. 12, no. 20, pp. 5226–5232, 2017.
- [18] S. Tiun, "Experiments on Malay short text classification," *Proc. 2017 6th Int. Conf. Electr. Eng. Informatics Sustain. Soc. Through Digit. Innov. ICEEI 2017*, vol. 2017-Novem, no. i, pp. 1–4, 2018.
- [19] N. A. M. Zamani, S. Z. Z. Abidin, N. Omar, and M. Z. Z. Abiden, "Sentiment Analysis : Determining People's Emotions in Facebook," *Appl. Comput. Sci.*, vol. ISBN: 978-, pp. 111–116, 2014.
- [20] N. Samsudin, M. Puteh, A. Razak, and M. Zakree, "Normalization of Common NoisyTerms in Malaysian Online Media," *Proc. Knowl. Manag. Int. Conf.*, no. July, pp. 515–520, 2012.
- [21] S. B. Basri, R. Alfred, and C. K. On, "Automatic spell checker for Malay blog," *Proc. - 2012 IEEE Int. Conf. Control Syst. Comput. Eng. ICCSCE 2012*, pp. 506–510, 2013.

Improving text simplification by corpus expansion with unsupervised learning

Akihiro Katsuta and Kazuhide Yamamoto

Nagaoka University of Technology

Nagaoka, Japan

{katsuta, yamamoto}@jnlp.org

Abstract—Automatic sentence simplification aims to reduce the complexity of vocabulary and expressions in a sentence while retaining its original meaning. We constructed a simplification model that does not require a parallel corpus using an unsupervised translation model. In order to learn simplification by unsupervised manner, we show that pseudo-corpus is constructed from the web corpus and that the corpus expansion contributes to output more simplified sentences. In addition, we confirm that it is possible to learn the operation of simplification by preparing large-scale pseudo data even if there is non-parallel corpus for simplification.

Keywords-unsupervised machine translation; Japanese simplification; corpus expansion;

I. INTRODUCTION

The number of foreigners in Japan has reached approximately 2.64 million and has been increasing¹. Guide plates and official documents are often written in both Japanese and English to accommodate foreigners. However, foreigners in Japan are not always able to understand English. As such, they often experience inconveniences or are otherwise disadvantaged in their daily life. There are many more non-English speaking foreigners (56%) than foreigners who do not speak Japanese (37%), in Japan[6]. Simplification is one way of addressing the problem whereby foreigners are unable to effectively access the required information. In this investigation, a practical system is desired.

Automatic sentence simplification aims to reduce the complexity of vocabulary and expressions in a sentence while retaining its original meaning. Current approaches often consider the simplification process as a monolingual text generation task such as machine translation[18; 19; 15; 23; 21; 9]. The translation model learns simple rewriting operations from a parallel corpus consisting of complex sentences and simplified sentences. Neural Machine Translation (NMT) can more effectively exploit large parallel corpora, although Statistical Machine Translation (SMT) is still superior when the training corpus is not big enough. The only corpus available in Japanese is a Japanese Simplified Corpus with Core Vocabulary (hereinafter referred to as SNOW T15+T23)². In this corpus, there are 85,000 sentences including a corpus of 50,000 sentences given by students (SNOW T15)[14] and a corpus of 35,000 sentences given to an anonymous person via crowdsourcing (SNOW T23)[8].

We built a Japanese text simplification model trained by large corpus automatically expanded from web text using an unsupervised translation model. We first construct a large-scale non-parallel corpus of complex and simplified sentences collected automatically. Next, we learn simplification operation by unsupervised translation. In fact, in order to automatically collect simplified sentences, it is necessary to estimate the readability of sentences and classify them into simplified sentences or complex sentences. One of the most commonly used methods for estimating readability in English is the Flesch Reading Ease Formula[4]. However, in this corpus, simplification task is focused on the compression of the vocabulary, and the operation of manual rewriting to the predetermined 2000 words (core vocabulary) is performed. Therefore, it is easy to collect simplified sentences automatically given that fundamentally, the corpus only checks to examine the vocabulary composed of core vocabulary.

II. RELATED WORKS

A. Unsupervised machine translation

Various neural models have been devised for unsupervised learning, presenting the possibility of machine translation techniques that do not require a parallel corpus. Artetxe et al. [3] and Lample et al. [11] have managed to train a standard attentional encoder-decoder NMT system from monolingual corpora alone. For that purpose, they use a shared encoder for both languages with pretrained cross-lingual embeddings, and train the entire system using a combination of denoising, back-translation and, in the case of Lample et al., adversarial training. This method was further improved by Yang et al. [20], who use a separate encoder for each language, sharing only a subset of their parameters, and incorporate two generative adversarial networks. However, Artetxe et al. [2] adapted the cross-lingual n-gram embeddings from monolingual corpora based on the mapping to train an unsupervised SMT model, obtaining large improvements over the original unsupervised NMT systems. It was argued that the modular architecture of phrase-based SMT was more suitable for this problem.

More recently, for additional improvement is attempted to combine both SMT and NMT to build hybrid unsupervised machine translation systems. This idea was already explored by Lample et al. [12], who aided the training of their unsupervised NMT system by combining standard back-translation with synthetic parallel data generated by unsupervised SMT.

¹<https://www.e-stat.go.jp/>

²<http://www.jnlp.org/SNOW>

This work builds off of existing work in the unsupervised machine translation, based on phrase-based SMT[2]. This work acquires phrase translation pairs by mapping pre-trained n-gram embedding between two languages to shared cross-lingual space and enables unsupervised learning. Unsupervised mapping by the method of Artetxe et al. [1] was used to map the n-gram embeddings to shared cross-lingual space. Further, they used iterative back-translation to train two unsupervised translation systems in both directions in parallel and to generate a synthetic source to construct a development set for tuning the parameters of their unsupervised statistical machine translation system. Iterative back-translation is a joint training algorithm to enhance the effect of monolingual source and target data by iteratively boosting the source-to-target and target-to-source translation models[22; 5].

B. For Text simplification

In the unsupervised paradigm, Paetzold and Specia [17] proposed an unsupervised lexical simplification technique that replaces complex words in the input with simpler synonyms, which are extracted and disambiguated using word embeddings. For another work as an approach for simplification that does not require a parallel corpus, Kajiwara and Komachi [7] proposed an unsupervised method in which a large-scale pseudo-parallel corpus is automatically constructed for text simplification based on the word similarity among monolingual corpora. They computed the sentence similarity for all pairings of normal and simple sentences using maximum alignment. Alignment was performed for only word pairs with a word similarity equal to or greater than 0.49 and only sentence pairs with a sentence similarity equal to or greater than 0.53 were aligned. As a result, 492,993 sentence pairs were obtained from 126,725 article pairs of English Wikipedia and Simple English Wikipedia. Given that this investigation is on text simplification in Japanese, the result of Maruyama and Yamamoto [13] is shown compared to that using NMT or SMT models, and the accuracy was comparable to SNOW T15 corpus.

III. METHOD FOR UNSUPERVISED LEARNING

This system was trained on monolingual corpus only and thus complex and simplified sentences were extracted from Nihongo Web Corpus 2010 (NWC 2010)³ and a monolingual corpus was constructed for each. This corpus was created from approximately 100 million web pages collected from June to September 2010.

A. building Simplification corpus

For unsupervised translation learning, training corpora are used as the independent monolingual corpora by eliminating the alignment of the bilingual corpus. However, in the simplified corpus, this condition causes the task advantageous in an unintended way. In a simplified corpus, there are pairs of sentences where complex sentences and complex words are simply rewritten. Therefore, there

is a high possibility that the same peripheral words of the operation target word are shared when a sentence is simplified using only a local substitution operation. This inevitably results in getting close to between correct substitute pairs in word embedding that has learned co-occurrence in the window.

The ultimate goal of simplification that does not require a parallelized simplification corpus is to learn simplification operations from non-parallel monolingual corpora. We show this advantage and investigate whether a large unclean corpus is more effective than a small clean corpus.

As briefly mentioned in section I, the aim is simplification in SNOW T15+T23 as lexical compression by substitution to the core vocabulary (the named entity and symbols are exceptionally permitted), so that simplified sentences can be easily collected from a monolingual corpus by examining whether the sentences contain anything other than the core vocabulary and named entity and symbols. We check the words each sentences and classify NWC 2010 to simplified and complex sentences and build a pseudo-corpus. It should be noted that when we use NWC 2010, we remove three or more consecutive equivalent tokens and symbols for cleaning text.

B. Unsupervised SMT

The method of Artetxe et al. was used to train an unsupervised simplification system from monolingual corpora using their open-source implementation monoses⁴. We trained an unsupervised SMT system from monolingual corpora using the default settings in the implementation.

1) *Cross-lingual mapping*: Complex sentences were prepared and simplified sentences were built for each n-gram embedding. The method applies a frequency-based vocabulary cut-off by learning the mapping over the 20,000 most frequently used words in each language. We kept this cut-off to learn the mapping over the most frequent 20,000 uni-grams and then applied the resulting mapping to the entire embedding space including

2) *induce phrase table*: The extracted phrase translation pairs for every n-gram in the simplified corpus could be taken as a potential translation candidate for each n-gram in the complex corpus. We limit the translation candidates for each complex phrase to its 100 nearest neighbors of simplified phrase.

To estimate their corresponding phrase translation probabilities, we applied the SoftMax function over the cosine similarities of their respective embeddings. More concretely, given the source language phrase \bar{e} and the translation candidate \bar{f} , their direct phrase translation probability was computed as follows:

$$\phi(\bar{f}|\bar{e}) = \frac{\exp(\cos(\bar{e}, \bar{f})/\tau)}{\sum_{\bar{f}'} \exp(\cos(\bar{e}, \bar{f}')/\tau)} \quad (1)$$

In this formula, \bar{f}' iterates across all target language embeddings and τ is a constant temperature parameter that controls the confidence of the predictions. For tuning, we

³<https://www.s-yata.jp/corpus/nwc2010/>

⁴<https://github.com/artetxem/monoses>

induce a dictionary over the cross-lingual embeddings with nearest neighbor retrieval and use maximum likelihood estimation. However, inducing the dictionary in the same direction as the probability predictions lead to a degenerated solution (SoftMax approximates the hard maximum underlying the nearest neighbor as τ approaches 0). Therefore, we induce the dictionary in the opposite direction at the same time and apply maximum likelihood estimation:

$$\min_{\tau} \sum_{\bar{f}} \log \phi(\bar{f}|NN_{\bar{e}}(\bar{f})) + \sum_{\bar{e}} \log \phi(\bar{e}|NN_{\bar{f}}(\bar{e})) \quad (2)$$

To compute the lexical weightings, we align each word in the target phrase with the one in the source phrase that most likely generating it, and take the product of their respective translation probabilities:

$$\text{lex}(\bar{f}|\bar{e}) = \prod_i \max(\epsilon, \max_j w(\bar{f}_i|\bar{e}_i)) \quad (3)$$

The constant ϵ guarantees that each target language word also in out of vocabulary will yield a minimum probability mass, which is useful for modelling NULL alignments. In our experiments, we set $\epsilon = 0.001$, which is the same experimental setting used by Artetxe et al.

3) Iterative back-translation: In general, standard SMT uses MERT over a small parallel corpus to tune the weights of the different scoring functions combined via its log-linear model. Given that we only have access to monolingual corpora in our scenario, we generate a synthetic parallel corpus through back-translation and apply MERT tuning iteratively, repeating the process in both directions. For this purpose, we reserved a random subset of 10,000 sentences from each monolingual corpora and ran the proposed algorithm over them for 10 iterations, which was sufficient for convergence. Thereafter, final tuning was performed on the data that was divided into training and validation data. Pseudo data was created by back-translation for each group and the phrase table was updated by applying MERT with validation. To accelerate our experiments, we use each monolingual corpus for training, in addition to the 10,000 separate sentences that were held out as a validation set for MERT tuning and performed a fixed number of 3 iterations of the aforementioned algorithm.

IV. EXPERIMENT

The systems are evaluated using BLEU scores computed by the “multi-bleu.perl” script included in Moses and SARI scores. In general, evaluation was performed to determine simplicity. In our experiment, tokenizer is McCab[10] using dictionary of UniDic⁵.

A. Comparison between supervised and unsupervised

In the previous work, Maruyama and Yamamoto [13] conducted experiments using a PB-SMT and a standard

Table I
DETAIL PARAMETERS OF THE MACHINE TRANSLATION

Parameter	Value
learning rate	0.25
clip-norm	0.1
dropout	0.3
max_tokens	4000
hidden size	512
Number of layers	2
Number of headers	8

bi-LSTM seq2seq model without attention in only a T-15. In this experiment, a baseline was set up with similar experimental settings.

First, we conducted to compare supervised manner and unsupervised manner. We calculated the accuracy of simplification using only SNOW T15+T23 by SMT and transformer as supervised learning and USMT as unsupervised learning. The difference between supervised one and unsupervised one is whether the sentences are aligned or not in simplification corpus. We extracted data from the corpus to 83000/1000/747 as train/valid/test. Given that some original sentences are originally simple, they included data whereby the original and simplified sentences match. The test data was dropped to omit such data. In our experiments, we use Moses⁶ as SMT and transformer that were implemented in fairseq[16]. The transformer performs parameter tuning and as a result, the parameter is Tab.I.

B. Usefulness of pseudo-corpus

For unsupervised learning, we used two kinds of corpora. We experimented with the case of using the pseudo-corpus constructed from NWC 2010 and the case of combining it with the training data of SNOW T15+T23.

As a result of extracting each 50 million sentences from the NWC 2010 using the method described in section III-A, details of the corpus are shown in TableII. In order to train USMT in multiple data scales, we further reduced the scale of the extracted corpora from NWC 2010 by several stages to obtain several scales (5M, 500k, 50k) of the corpus. We used USMT to learn simplification with each scale. At the same time, we also try that with combining SNOW T15+T23.

V. RESULTS AND DISCUSSION

A comparison of each system using SNOW T15+T23 to learn text simplification is shown in Table III. The baseline gives no rewriting to the input sentence and thus the output is as same as the input. It is difficult to exactly compare the BLEU score of Maruyama and Yamamoto [14] with our score because the smoothing of BLEU calculation method is not clear in their experiment. However, it is possible to compare the results of the supervised approach that we conducted in SARI score. Thus TableIV shows that the transformer is the model that can generate the simplified sentence most because it is the highest SARI and reduces the vocabulary.

⁵<https://unidic.ninjal.ac.jp/>

⁶<http://www.statmt.org/moses/>

Table II
DETAIL OF DATASETS FOR TEXT SIMPLIFICATION (IN SNOW T15+T23, IT SHOWS EACH TRAIN/VALID/TEST.)

datasets	#sentences	Vocabulary	#tokens per sent.
SNOW T15+T23 (complex)	83,000/1,000/747	20,339/2,034/2,015	10.87/11.01/11.30
SNOW T15+T23 (simplified)	83,000/1,000/747	5,392/1,464/1,492	12.04/12.13/12.95
NWC 2010 (complex)	50,000,000	1,227,318	16.59
NWC 2010 (simplified)	50,000,000	81,925	10.25

Table III
SIMPLIFICATION RESULT IN SNOW T15+T23

model	BLEU	SARI	Vocabulary
baseline	48.22	22.19	2015
NMT (Maruyama et al.[14])	79.4	58.5	-
SMT (ours)	60.88	69.97	1372
transformer (ours)	60.46	72.44	1223
USMT (ours)	54.15	47.11	1755

Table IV
IMPROVEMENT ACCURACY WHEN EXTENDING CORPUS USING PSEUDO-CORPUS

data-size of NWC2010	BLEU	SARI	Vocabulary
without SNOW T15+T23 (83,000)			
50M	52.78	55.57	1685
5M	51.94	59.82	1547
500k	49.37	43.03	1722
50k	42.98	37.10	1766
with SNOW T15+T23 (83,000)			
50M	52.32	50.80	1677
5M	53.30	56.39	1616
500k	54.06	51.10	1699
50k	53.68	52.37	1728
5k	54.65	50.44	1715

As compared with the baseline, the result of the unsupervised learning shows that the score is improved and correct rewriting can be performed even if alignment information is lost.

Table IV shows the simplification accuracy by changing the learning data size of USMT. As a result, regardless of whether SNOW T15+T23 is included, when unsupervised SMT is learned, it is evident that the simplification accuracy improves and the vocabulary scale is reduced depending on the scale of the training data. However, using SNOW T15+T23, BLEU shows a different tendency. Although the SARI score improves, there is almost no difference in the BLEU score depending on the scale of the training data. It is considered that the vocabulary is further compressed because the case for which the solution is robustly solved is sufficiently learned by SNOW T15+T23 only, and expressions that the test data cannot consider are learned from the NWC 2010. Therefore, it is seen that even if the alignment information is dropped, it is advantageous for the unsupervised approach that the original was a parallel corpus. However, this result also shows that in its absence, comparable results can be obtained by expanding the scale of training data.

In Table V, we compares the output examples of each model. (a) shows the change with the size of the data, and (b) shows an error case often seen when expanding with a pseudo-corpus. As shown in (a), it can be seen that if the data is small, relatively many incorrect rewriting (for

example, the “洪水, flood” has been rewritten as “話し合い, discussion” in the case of 500k) or cases are not performed, whereas rewriting quality improves as data is increased. In (b), there is a case where the meaning of rewriting becomes opposite by combining pseudo data. When USMT learned only SNOW T15+T23, “がっかりした, disappointed” is rewritten to “気を落とした, discouraged”, but when expanded SNOW T15+T23 with a pseudo-corpus, it is rewritten to ”驚いた, surprised” and when learned only with a pseudo-corpus, it is rewritten to ”感動した, impressed”. The inability to distinguish antonyms is due that word embedding has a problem, and in order to eliminate such (b), it is necessary to devise measures to keep antonyms away when learning word embedding.

VI. CONCLUSION

We constructed each corpus of complex sentences and simplified sentences automatically as a pseudo-corpus from a nonparallel corpus and showed that expanding simplification corpus in USMT promotes more positive simplification. When USMT is trained using a data set obtained by dropping the alignment from the parallel corpus, the enhancement of BLEU by the expansion of the corpus is not seen. However, since the score of SARI has been improved and the vocabulary of the output sentence has been reduced, it has become possible to rewrite it into a more simplified sentence. In addition, in the case of learning only with a pseudo-corpus with a small scale, sufficient word embedding learning can not be performed, and BLEU becomes worse than the baseline. Since such a tendency is not seen when using a parallel corpus, we can see that the parallel corpus is still advantageous even if we learn from the parallel corpus that dropped the alignment information.

In the future, we would like to improve the system by incorporating model improvements, such as using a neural network, and better training schemes in order to address complex simplification operations.

ACKNOWLEDGMENT

This work was supported in part by JSPS KAKENHI Grants-in-Aid for Challenging Research (Exploratory) Grant ID 17K18481.

REFERENCES

- [1] M. Artetxe, G. Labaka, and E. Agirre. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*,

Table V
EXAMPLES OF WORD REPLACEMENT

(a)	
Original Sentence	彼らは大雨といえば洪水を連想した。 Speaking of heavy rain, they associate floods.
USMT(50M)	彼らは雨といえば災害をイメージした。 Speaking of rain, they imaged a disaster.
USMT(5M)	彼らはまた雨といえば自然災害をイメージした。 Speaking of rain, they imaged a natural disaster.
USMT(500k)	彼らは札幌といえば話し合いを連想した。 Speaking of Sapporo, they associated a discussion.
USMT(50k)	彼らは大雨といえば洪水を発展した。 Speaking of heavy rain, they developed floods.
SMT	彼らはひどい雨といえば被害を思い出させた。 Speaking of heavy rain, they reminded the damage.
transformer	彼らはすごい雨といえば水が災害を想像した。 Speaking of great rain, they imagined a disaster of water.
(b)	
Original Sentence	彼はその成績にがっかりした。 He was disappointed at the grade.
USMT(50M)	彼はその結果に感動した。 He was impressed at the result.
USMT(50M + SNOW T15+T23)	彼はその結果に驚いた。 He was surprised at the result.
USMT(SNOW T15+T23 only)	彼はその評価に気を落とした。 He was discouraged by the evaluation.
SMT	彼はその結果に残念な気持ちになった。 He felt disappointed with the result.
transformer	彼はその試験の結果を残念に思った。 He was disappointed with the results of the exam.

- pages 789–798, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/P18-1073>.
- [2] M. Artetxe, G. Labaka, and E. Agirre. Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3632–3642, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/D18-1399>.
- [3] M. Artetxe, G. Labaka, E. Agirre, and K. Cho. Unsupervised neural machine translation. In *Proceedings of the Sixth International Conference on Learning Representations*, April 2018.
- [4] R. Flesch. A new readability yardstick. *Journal of applied psychology*, 32(3):221, 1948.
- [5] V. C. D. Hoang, P. Koehn, G. Haffari, and T. Cohn. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/W18-2703>.
- [6] K. IWATA. The preference for english in linguistic services : apos;japanese for living: Countrywide surveyapos; and hiroshima(lt;special issuegt;changing japanese society and language issues). *The Japanese Journal of Language in Society*, 13(1):81–94, 2010. doi: 10.19024/jajls.13.1_81.
- [7] T. Kajiwara and M. Komachi. Text simplification without simplified corpora. In *The Journal of Natural Language Processing*, volume 25, pages 223–249, 2018. doi: 10.5715/jnlp.25.223.
- [8] A. Katsuta and K. Yamamoto. Crowdsourced corpus of sentence simplification with core vocabulary. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan, May 2018. European Languages Resources Association (ELRA). URL <https://www.aclweb.org/anthology/L18-1072>.
- [9] R. Kriz, J. Sedoc, M. Apidianaki, C. Zheng, G. Kumar, E. Miltsakaki, and C. Callison-Burch. Complexity-weighted loss and diverse reranking for sentence simplification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3137–3147, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/N19-1317>.
- [10] T. Kudo, K. Yamamoto, and Y. Matsumoto. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 230–237, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/W04-3230>.
- [11] G. Lample, A. Conneau, L. Denoyer, and M. Ranzato. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations (ICLR)*, 2018.
- [12] G. Lample, M. Ott, A. Conneau, L. Denoyer, and M. Ranzato. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.
- [13] T. Maruyama and K. Yamamoto. Sentence simplification with core vocabulary. In *2017 International Conference on Asian Language Processing (IALP)*, pages 363–366, Dec 2017. doi: 10.1109/IALP.2017.8300618.
- [14] T. Maruyama and K. Yamamoto. Simplified corpus with core vocabulary. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan, May 2018. European Languages Resources Association (ELRA). URL <https://www.aclweb.org/anthology/L18-1185>.
- [15] S. Nisioi, S. Štajner, S. P. Ponzetto, and L. P. Dinu. Exploring neural text simplification models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 85–91, Vancouver, Canada,

- July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2014. URL <https://www.aclweb.org/anthology/P17-2014>.
- [16] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.
 - [17] G. H. Paetzold and L. Specia. Unsupervised lexical simplification for non-native speakers. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
 - [18] S. Wubben, A. van den Bosch, and E. Krahmer. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1015–1024, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/P12-1107>.
 - [19] W. Xu, C. Napoles, E. Pavlick, Q. Chen, and C. Callison-Burch. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4: 401–415, 2016. doi: 10.1162/tacl_a_00107. URL <https://www.aclweb.org/anthology/Q16-1029>.
 - [20] Z. Yang, W. Chen, F. Wang, and B. Xu. Unsupervised neural machine translation with weight sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 46–55, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1005. URL <https://www.aclweb.org/anthology/P18-1005>.
 - [21] B. Zhang, D. Xiong, J. Su, Q. Lin, and H. Zhang. Simplifying neural machine translation with addition-subtraction twin-gated recurrent networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4273–4283, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/D18-1459>.
 - [22] Z. Zhang, S. Liu, M. Li, M. T. Zhou, and E. Chen. Joint training for neural machine translation models with monolingual data. *ArXiv*, abs/1803.00353, 2018.
 - [23] S. Zhao, R. Meng, D. He, A. Saptono, and B. Parhamo. Integrating transformer and paraphrase rules for sentence simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3164–3173, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/D18-1355>.

Cross Language Information Retrieval Using Parallel Corpus with Bilingual Mapping Method

Rinaldi Andrian Rahmada, Mirna Adriani, Dipta Tanaya

Faculty of Computer Science, Universitas Indonesia

Depok, Indonesia

rinaldi.andrian@ui.ac.id, mirna@cs.ui.ac.id, diptatanaya@cs.ui.ac.id

Abstract—This study presents an approach to generate a bilingual language model that will be used for CLIR task. Language models for Bahasa Indonesia and English are created by utilizing a bilingual parallel corpus, and then the bilingual language model is created by learning the mapping between the Indonesian model and the English model using the Multilayer Perceptron model. Query expansion is also used in this system to boost the results of the retrieval, using pre-Bilingual Mapping, post-Bilingual Mapping and hybrid approaches. The results of the experiments show that the implemented system, with the addition of pre-Bilingual Mapping query expansion, manages to improve the performance of the CLIR task.

Keywords-- *Cross Language Information Retrieval; parallel corpus; Bilingual Mapping; Language Model; Multilayer Perceptron*

I. INTRODUCTION

Cross Language Information Retrieval deals with searching documents written in different languages with user's queries. It requires a translation process due to language difference between queries and documents. In general, two methods which could be used to solve this problem are query/documents translation and bilingual model approach [1]. Translation approach looks for matching words in both languages by using dictionary [2], machine translation [3], or parallel corpus [4]. In this approach, a high-quality bilingual dictionary or machine translator resource are needed to get good results. Many studies have been carried out using this translational approach. Ballesteros and Croft's research uses a bilingual dictionary that can be read by a computer (Machine Readable Dictionary) to translate queries [2]. Oard's study uses a bilingual dictionary and machine translation in a comparative study [3]. Research by [4] uses a parallel corpus to search for pairs of words in both languages to form queries for the target language.

Meanwhile, in the bilingual model approach, each document in both languages is mapped into a vector space, represented by word embedding. Word embedding is capable to capture word similarity, context of a word in a document, word relation, etc. By using word embedding, the similarity between query and documents can be measured by calculating the vectors similarity. This

bilingual language approach has been studied by [5]. In this study, Vulic and Moens used the bilingual language model approach using word embedding for CLIR.

An excess potential of the bilingual language model approach using word embedding compared to the translational approach is how similar words will have high similarity values. This will be useful in cases where the words used in the query are translated differently from the words contained in the desired document, even though they have the same meaning. For example, in the case of an Indonesian-English CLIR, a query containing "senang" (in English: glad) was given. Meanwhile, documents that are expected to contain the word "glad". In the translation approach, the system will fail to search for documents if the word "senang" (in English: glad) is translated into words other than "glad", for example "happy". Fortunately, in the language model approach, this does not matter because of the high similarity of the two words, "senang" (in English: glad) and "glad".

The potential of the language model approach, and the success of Vulic and Moens' research [5], inspired this research. This study will try to apply language model approach in a CLIR system. To create a bilingual model that will be used by the CLIR system, the bilingual mapping method as done in [6]. First, we use a bilingual parallel corpus from Global Voices. the language model or language representation for both languages in the parallel corpus is formed using word embedding. Then, the bilingual model will map the representation of the language. This mapping system will use the neural network classifier to get the best transformation for all existing training data. Using this model, CLIR will be done by taking a query, constructing a vector representation, mapping the vector into English, then searching for the relevant documents.

Research on CLIR is important to do because of the gaps phenomenon in the availability of information in a specific language. For example, we can observe that the amount of information available on the internet in Indonesian is still less than the information available in English. The CLIR system can help overcome this problem by finding and displaying available information, even though the information is available in other languages. The research on Cross Language Information Retrieval is expected to help overcome the problem of information gap.

II. RESEARCH DESIGN

The following steps we employ to create bilingual model for CLIR task:

1. Preparing parallel corpus
2. Training word embedding as word representation
3. Learning bilingual mapping
4. Use the bilingual model for CLIR task
5. Improving CLIR results by query expansion

A. Parallel Corpus

Bilingual models for this system will be built with the help of a parallel corpus. The parallel corpus contains a collection of pairs of sentences in Indonesian and English. These pairs of sentences will be used in the training process of mapping words from Indonesian to English. We use a news collection published on the GlobalVoices website¹ in Indonesian and English, taken from the OPUS² open-source parallel corpus project [7]. The corpus has 11,488 sentence pairs, around 200,000 Indonesian language tokens and around 200,000 English token. Information taken from GlobalVoices is the parallel corpus as well as the pairs of Indonesian and English words found in the corpus. The dictionary corpus contains 4,851 pairs of words that can be found in the parallel corpus.

B. Making Word Representations from Parallel Corpus

We use word embedding to represent words in parallel corpus. Each word in the corpus will be converted into a vector representation in a certain dimension. Word vector representation is made using the Doc2Vec model from the gensim³, a library used for performing several NLP and IR tasks such as modeling topics, document indexing and similarity retrieval [8]. The word embedding building includes the following stages:

1. Prepare the GlobalVoices parallel corpus containing 11,488 pairs of sentences.
2. Initialize the Doc2Vec model. The model is initialized by selecting the Distributed Bag of Words (DBOW) training algorithm, using the dbow_words option so that the model produces the word vector. As for the dimensions of vectors, we will try several size variations and look for the best vector size.
3. Train the Doc2Vec. The training process will produce a word embedding model where each word in the corpus has a corresponding vector. By using the Doc2Vec model, each sentence in the corpus will also have a corresponding vector, as well as one or more labels as the identifier of the document.

C. Bilingual Mapping

Bilingual language models are formed using the bilingual mapping method. This method uses word representation vectors of documents in both languages. Then, using corresponding word vectors from both languages, a transformation that can map vectors from the original language to the target language is formed. Mapping will be done using Multilayer Perceptron.

After doing word embedding training from parallel corpus, word vectors will be obtained. The next step is to use the corpus dictionary to get pairs of corresponding words in the corpus. For each pair of words on this dictionary, take the word vector for the word from both languages, then use these two vectors as the input and output expected from the training process. Through this process, a bilingual language model that can predict the target language word from the input word of the original language will be formed. For example, if we have the Indonesian word vector "apel" (in English: apple) [0.5, 0.3, 0.1, 0.9, 0.3] and the "apple" English word vector [0.1, 0.6, 0.8, 0.9, 0.8], the transformation system is expected to find a transformation which can map vectors [0.5, 0.3, 0.1, 0.9, 0.3] to [0.1, 0.6, 0.8, 0.9, 0.8]. According to [9], in making bilingual models based on bilingual mapping, the resulting bilingual model will be more optimal if the dimensions of the original language vector representation are two to four times greater than the target language vector representation. In this study, several variations of word embedding dimension models will be used to see the relationship between the dimensions of the model and the quality of the results of the CLIR system.

D. CLIR System

Cross Language Information Retrieval is done by using the bilingual language model. The process of retrieving relevant documents is done by comparing document vectors and query vectors. For this reason, the document embedding method is carried out on news documents from the corpus in order to obtain document vector from each document.

To do the cross language information retrieval, several steps were taken from adaptations of the steps used in [5]:

1. take the query,
2. using the existing model, get the target language version of the query,
3. get the closest documents from the query by measuring the similarity score between the query and the document,

4. sort the document based on the similarity score.
Retrieval in this experiment consists of two types, monolingual retrieval and bilingual retrieval. The results of monolingual retrieval will be the baseline of the CLIR system. The retrieval process will be carried out on two corpus documents:

1. GH95, a news corpus from the 1995 Glasgow Herald.

¹ <http://globalvoices.org/>

² <http://opus.nlpl.eu/>

³ <http://radimrehurek.com/gensim/>

2. LAT94, a news corpus from the 1994 Los Angeles Times newspaper.

The query is taken from CLEF (Cross Language Evaluation Forum)⁴. We take 50 queries that have the Indonesian and English versions. Each query has a title section (topic), a description of what information is expected (description), and a brief narrative about the desired information (narration). From these sections, the title and description are taken only referring to previous research [5] [10].

E. Query Expansion

To improve the quality of the obtained relevant documents, we add query expansion to the retrieval process. According to [11], query expansion has the potential to improve the quality of information retrieval results because query expansion can overcome the main problem that decreases search quality, that is the problem when the words in the query do not match the expected document (vocabulary problem).

One of the query expansion methods that can be done is using the word embedding model. In this method, each word in the query is expanded using the words closest to the word in word embedding space. The similarity between words is measured using the cosine similarity value between word vectors. For example, if given a "car" query, and the closest words to the word "car" in word embedding space are "vehicles" and "automotive" then the query will be expanded to "automotive vehicle cars". This word embedding based method has been investigated in [12] and produce better results compared to queries that are not expanded.

In this study, we try 3 expansion query methods: Pre-BM Query Expansion, Post-BM Query Expansion, and a combination of both methods. Pre-BM Query Expansion expands the query before mapping proses to the target language is done. Post-BM Query Expansion expands the query after mapping query in Indonesian to English using a bilingual language model. The combined method combines Pre-BM Query Expansion and Post-BM Query Expansion.

F. Evaluation

IR system evaluation is done by comparing the retrieval results from the IR system with the gold standard that has been made based on the results of manual evaluations by humans. Evaluation is done by measuring the value of Mean Average Precision (MAP) from the results of the retrieval.

Average Precision is the average value of precision for each document that is considered relevant to a query. The formula for calculating Average Precision is as follows [13]:

$$AP = \frac{\sum_{d=1}^n P(d)x rel(d)}{number\ of\ relevant\ documents}$$

where n is the number of documents that should be considered relevant, d is the document number, $P(d)$ is the value of the precision query on the document and $rel(d)$ is 1 if the document is relevant; 0 if the document is not considered relevant.

For calculating MAP, we use formula as follows [13]:

$$MAP = \frac{\sum_{q=1}^n AP(q)}{n}$$

The formula calculates the mean of Average Precision from each query. The q variable represents a query, $AP(q)$ is Average Precision from the query, and n is the number of queries.

III. RESULTS AND DISCUSSION

A. Language Model

The results of the word embedding training for each language are monolingual language models for both languages. From the use of the DBOW + dbow_words algorithm for this training, word vectors are obtained for each word in the corpus. The results of training data in English can be directly used for monolingual English document retrieval.

In this study, a bilingual language model was also produced. Similar to the monolingual language model, bilingual language models can search for words with a high level of similarity. The difference is that when an Indonesian word is entered as input, the system gets a vector representation of the word, and converts that representation into an English form. Then the system will look for similar vectors in English using this new vector. For example, the search for the word "perang" (in English: war). The system will get the words war, wars, unrest, apartheid, perestroika, strike, warfare, battle, recession, and riot.

B. Retrieval Results

Information retrieval is done in several scenarios, and the results are assessed using the trec_eval application. Performance is measured using MAP metrics, or Mean Average Precision. First, as a baseline for this language model approach, retrieval is done using Okapi BM-25. Retrieval is done twice, monolingual (English to English) and cross-language (Indonesian to English). Cross language retrieval is done by previously translating the query Indonesian into English using Google Translate. This is called the method Google Translate + Okapi BM-25. The results of this method can be seen in Table 1.

Table 1 Okapi BM-25 Evaluation (Baseline)

	Method	MAP
1	Okapi BM-25	0.3869
2	Google Translate+Okapi BM-25	0.3370

⁴ <http://clef.isti.cnr.it/>

Then, retrieval using a monolingual language mode (English to English) is done, as a baseline for the performance of cross-language retrievals performed. The four dimensions of word embedding are used at this stage, namely 100, 200, 300, and 400. The results of this stage can be seen in table 2.

Table 2 shows the results of evaluation of monolingual information retrieval (English queries to English documents) using the language model of word embedding. In general the results obtained here are worse than those obtained from the monolingual Okapi BM-25 method.

Table 2 Monolingual (English-English) Retrieval Evaluation

Method	MAP	vs baseline	Method	MAP	vs baseline
WE-100	0.3623	-2.46%	WE-300	0.3654	-2.15%
WE-200	0.3693	-1.76%	WE-400	0.3697	-1.72%

Furthermore, cross-language retrieval is done. Several combinations of word embedding dimensions from both languages are used. The results of this stage in more detail can be seen in table 3. The following are the results of evaluation retrieval using the trec_eval application:

Table 3 Bilingual Retrieval Evaluation

	Method	MAP	vs baseline	vs mono-lingual
1	Monolingual (WE) -> WE-English-100	0.2967	-4.03%	-7.3%
2	WE-Ind-100 -> WE-English-100	0.2967	-4.03%	-7.3%
3	WE-Ind-200 -> WE-English-100	0.3328	-0.42%	-3.69%
4	WE-Ind-200 -> WE-English-200	0.2843	-5.27%	-8.54%
5	WE-Ind-300 -> WE-English-100	0.3353	-0.17%	-3.44%
6	WE-Ind-300 -> WE-English-200	0.3251	-1.19%	-4.46%
7	WE-Ind-300 -> WE-English-300	0.2798	-5.72%	-8.99%
8	WE-Ind-400 -> WE-English-100	0.3525	+1.55%	-1.72%
9	WE-Ind-400 -> WE-English-200	0.3369	-0.01%	-3.28%
10	WE-Ind-400 -> WE-English-300	0.3163	-2.07%	-5.34%
11	WE-Ind-400 -> WE-English-400	0.2651	-7.19%	-10.46%

Table 3 shows that the best results from the CLIR system are obtained when the system uses word embedding using an Indonesian language vector of 400 dimensions and an English language vector with a dimension of 100. The greater use of initial language vectors gives better retrieval results. However, large vector dimension results increasing training time.

Monolingual query is still better compared to the cross-language retrieval result. A possible cause of this phenomenon lies in the process of vector transformation between languages. In the process of bilingual mapping, the vector of an Indonesian word is not 100% mapped with English. For example, the word "perang" (in English: war) is not 100% the same as the word "war" in English (the similarity is 0.8491). This difference can cause changes in the value of the relevance of some documents and lead to the worse results.

The results of the developed CLIR system retrieval are better than the results of the Okapi BM-25 using Google Translate of 0.0155 or 1.55%. However, the results obtained using the monolingual language model are worse than the results of the Okapi BM-25 method, with lower MAP results of 0.0172 or 1.72%. The use of language models in information retrieval tasks like this certainly depends on the quality of the data used in the training steps. The data used must be large enough and cover the contents of the corpus that is the target of retrieval. The language style of training data may also have an effect. But the biggest factor why the use of this language model is still losing is the mapping done from documents in the form of words to vector space. By mapping words in a document into a vector, some information inherent in the document might be lost. This resulted in a decrease in retrieval performance that could be done, and resulted in the Okapi BM-25 algorithm as the baseline still leading in the monolingual retrieval aspect.

Using query expansion methods, the following results are obtained:

Table 4 Query Expansion Evaluation

	Method	MAP Non-QE	MAP QE Pre-BM	MAP QE Post-BM	MAP Pre-Post BM QE
1	WE-Ind-100 -> WE-English-100	0.2967	0.3119	0.3029	0.2987
2	WE-Ind-200 -> WE-English-100	0.3328	0.3538	0.3533	0.3589
3	WE-Ind-200 -> WE-English-200	0.2843	0.3018	0.3018	0.2977
4	WE-Ind-300 -> WE-English-100	0.3353	0.3564	0.3556	0.3554
5	WE-Ind-300 -> WE-English-200	0.3251	0.3446	0.3356	0.3398
6	WE-Ind-300 ->	0.2798	0.2952	0.2945	0.2938

	Method	MAP Non- QE	MAP QE Pre- BM	MAP QE Post- BM	MAP Pre- Post BM QE
	WE-English-300				
7	WE-Ind-400 -> WE-English-100	0.3525	0.3746	0.3739	0.3727
8	WE-Ind-400 -> WE-English-200	0.3369	0.357	0.3571	0.3559
9	WE-Ind-400 -> WE-English-300	0.3163	0.3343	0.3347	0.3345
10	WE-Ind-400 -> WE-English-400	0.2651	0.2807	0.2814	0.2803

Table 4 shows the results obtained from the use of query expansion methods, with three approaches. The table above shows an increase in MAP in all dimensions variations used. The results of Indonesia word embedding with dimensions 400 and English word embedding with dimension 100 by using the query expansion method Pre-BM managed to get MAP of 0.3746 which is higher than non-QE retrieval of 0.0221 or 2.21%. This result is 3.76% better from the baseline. This is the best result obtained by this system.

IV. CONCLUSION

In conducting Cross Language Information Retrieval, various methods can be applied. One of them is using a bilingual language model. Bilingual language model is created from the results of transformation of the original language word vector to the targeted word vector language.

In this study, an Indonesian-English CLIR system is developed using a bilingual language model approach. A parallel corpus containing documents in Indonesian language and English are used as the basis of the model. Word embedding and document embedding are carried out to get the vector representations of each language. Then, the Multilayer Perceptron is used to map vectors from Indonesian language to corresponding vectors in English to form a bilingual language model.

CLIR is then done by taking a query in Indonesian, getting the vector representation, mapping the vector into English, and finally looking for the documents that best match the query vector. Experiments were carried out by comparing the MAP values from the results obtained by the CLIR system to the Okapi-BM25 baseline and monolingual retrieval results using language models with several vector dimension combinations. The best results obtained showed a MAP value of 35.25%, 1.55% better than the baseline result, but 1.72% lower than the monolingual result.

To improve the search results obtained, query expansion methods are used to enrich search words. Three

query expansion approaches are carried out: query expansion before query mapping in bilingual language model (Pre-BM), query expansion after query mapping (PostBM), and a combination of the two approaches.

The experiment was done to examine the quality of the query expansion. The experiments show increasing results for all vector combinations, with the Pre-BM approach giving better results than the other two approaches. The best result obtained at this stage is the MAP value of 37.46%, which is 3.76% better from baseline.

To conclude, a CLIR system was developed using a bilingual language model with a bilingual mapping approach. The method of query expansion is done to enrich the query with the aim of improving search results. The best result obtained by the system is MAP of 37.46%, an increase from the baseline of 3.76%.

After conducting experiments and analyzing the results obtained in this study, there are some suggestions that might be useful for further research:

1. In conducting training in making bilingual language models, the quality and quantity of training data is very important. For further research, a larger parallel corpus can be used with more words and documents so that the mapping process can be done better.
2. In the process of bilingual mapping, other types of classifier other than MLP can be used, for example Convoluted Neural Network (CNN).
3. This Bilingual Mapping method can be tried in other languages to see the difference in performance.
4. The bilingual mapping approach used in this study is only one-way, from Indonesian to English. For further research, the mapping process can be done in two-way.
5. Additional features can be explored to conduct information retrieval, such as Latent Semantic Indexing.

ACKNOWLEDGEMENTS

We gratefully thank the Universitas Indonesia for the International Publication Grants (*Hibah PITTA-B*)

References

- [1] P. Sorg and P. Cimiano, "Cross-lingual information retrieval with explicit," in *In Working Notes for the CLEF 2008 Workshop*, 2008.
- [2] L. Ballesteros and W. B. Croft, "Phrasal translation and query expansion," *ACM*, vol. 31, pp. 84-91, 1997.
- [3] D. W. Oard, "A comparative study of query and document translation," in *Conference of the Association for Machine Translation in the Americas*, 1998.
- [4] J.-Y. Nie, M. Simard, P. Isabelle and Durand, "Cross-language information retrieval based on parallel texts and automatic mining of parallel texts

- on the web," in *22nd annual international ACM SIGIR conference on Research and development in information retrieval*.
- [5] I. Vulic and M. -F. Moens, "Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings," in *38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015.
 - [6] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient estimation of word representations in vector space," in *Workshop at ICLR*, 2013.
 - [7] J. Tiedemann, "Parallel data, tools and interfaces in opus," in *Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, 2012.
 - [8] R. Rehurek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *LREC 2010 Workshop on New Challenges for NLP Frameworks*, 2010.
 - [9] T. Mikolov, Q. V. Le and I. Sutskever, *Exploiting similarities among languages for machine translation*, CoRR, abs/1309.4168, 2013.
 - [10] I. Vulic, W. De Smet and M. F. Moens, "Cross-language information retrieval models based on latent topic models trained with document-aligned comparable corpora," *Information Retrieval*, vol. 3, no. 16, pp. 331-368, 2013.
 - [11] C. Carpineto and G. Romano, "A survey of automatic query expansion in information retrieval," *ACM Computing Surveys (CSUR)*, vol. 44, no. 1, 2012.
 - [12] D. Roy, D. Paul, M. Mitra and U. Garain, *Using word embeddings for automatic query expansion*, CoRR, abs/1606.07608, 2016.
 - [13] A. Turpin and F. Scholer, "User performance versus precision measures for simple search tasks," in *29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006.

Language Detection in Sinhala-English Code-mixed Data

Ian Smith

Computer Science Engineering
University of Moratuwa, Sri Lanka
royian.18@cse.mrt.ac.lk

Uthayasanker Thayasilvam

Computer Science Engineering
University of Moratuwa, Sri Lanka
rtuthaya@cse.mrt.ac.lk

Abstract— Language identification in text data has become a trending topic due to multiple language usage on the internet and it becomes a difficult task when it comes to bilingual and multilingual communication data processing. Accordingly, this study introduces a methodology to detect Sinhala and English words in code-mixed data and this is the first research done on such scenario at the time of this paper is written. In addition to that, the data set which is used for this research was newly built and published for similar research users. Even though there are well known models to identify Singlish Unicode characters which is a straightforward study; there are no proper language detection models to detect Sinhala words in a sentence which contains English words (code-mixed data). Therefore, this paper presents a language detection model with XGB classifier with 92.1% accuracy and a CRF model with a F1-score of 0.94 for sequence labeling.

Keywords—code-mixed, code-switching, Sinhala-English, language detection, social media data

I. INTRODUCTION

To begin with, language identification (LID) is the process of determining the natural language or the language features of a document, part of a document or in a line segment. On the other hand, one of the main characteristics of being a human being is the ability to use complex and sophisticated ideas and thoughts within the communication process using a common language. Accordingly, humans have the ability to identify languages or language features swiftly from a given line segment provided that the text contains languages those are which the human is familiar with. Therefore, it is noteworthy that even the humans cannot identify the languages used in a given document if they are not familiar with the languages used in the text. Hence, the aim of a language detection process is to identify an approach for humans to recognize specific languages used in the given text. Subsequently, there are number of researches those have been conducted to develop specific/general language identification models. Likewise, several data structures, algorithms and models have been developed to model the human ability to identify languages. Furthermore, most of the people were using various encoding methods to present text until the Unicode standard (preceded by ISO 8859, various others) was introduced. Subsequently, language detection becomes comparatively easier with the Unicode characters. Indeed, with these standards and algorithms, the ability to detect the languages based on the encodings in the text is straightforward. However, these simple models can only handle text based on the encoding and they simply rely on the encoding boundaries which fails in most of the occasions such as in social media data and other scenarios where code-mixing occurs.

On the other hand, usage of social media has become a day-to-day activity of human lives as a result of the advancement in technology. Accordingly, social media sites such as Facebook, Twitter and Instagram generate vast amount of data from their services. Moreover, users of social media come from all over the world, each bringing their own backgrounds and cultures into the mix [6]. With such global user base, social media becomes a melting pot of languages

used in different manners and for different purposes. This has resulted in creation of an alternate dialect in favors real-time communication such as acronyms and short forms of words that are used in instant messages. For example, people often use expressions such as LOL for laugh out loud, ROFL for rolling on floor laughing, NVM never mind and YOLO for you only live once and common reacts like HAHA for funny reactions. Indeed, these acronyms can be mostly seen in chat communications where users tend to type to express themselves with less characters rather than typing the whole meaning.

Even though English is considered as the native language for web communications, there are significant number of people who use social media with various native languages other than English. However, most of these people do not use Unicode characters to write in their language. Instead of Unicode, most of these people use phonetic typing, frequently inserting English elements in their communication. Accordingly, this has become the latest trend in social media due to the simplicity. Therefore, people express their languages using English characters and they even insert English words mixed up with the native language words. This phenomenon is known as the code-mixing or code-switching interchanging in some researches. Accordingly, in this study, code-mixing is used to identify this phenomenon. Further, the code-mixing can be mostly seen in peer to peer communication, group communication, comments and status updates and so on. On the other hand, people tend to mix Unicode characters with English characters as well and therefore, there can be many variations in each social media content data (SMCD) segment as shown in table 1.

TABLE 1: TYPES OF LANGUAGE VARIATIONS IN SOCIAL MEDIA

Content type	Example
Purely written in English	I'm going to the university today
Purely written in Unicode	අද මම විශ්ව විද්‍යාලයට යනවා
Written in a language other than English using English characters	Ada mama vishva vidyalayata yanawa
English characters with Unicode	අද මම university යනවා
Code mixing	Ada mama university yanawa

Likewise, people tend to use English alphabetical characters to express their language. Therefore, some new language variations have evolved amongst social media users such as, people mix English with their native language in communication (mostly in writing): Chinglish (Chinese written in English), Hinglish (Hindi written in English) and Singlish (Sinhala written in English). In fact, each of these language variations are expressed using English alphabet and some of the social media users combine English with Unicode characters as well. In Sri Lanka, many marketing campaigns conducted on social media, via emails or text messages use Singlish language features to convey the message to the users effectively. Due to the variety of derived languages used in social media, automatic language identification (ALID) has

become a key research area in natural language processing domain where types of languages and language features are to be detected on a given text. Accordingly, language detection models are expected to classify the language used in a given text or a line segment. This is mainly formed on text written in one language and some sophisticated models are used if the given text contain more than one language (code-mixed data). Further, most language identification models perform well on a single or multiple languages, yet, most of them fail when there are texts with mixed language features. Therefore, even though the language detection has been achieved up to a higher level of accuracy, it is not a fully achieved task in NLP domain.

Likewise, there are many elements to be discovered in language detection in NLP domain and these discoveries become difficult due to rapid changes in human behavior. In fact, it is even more challenging the evaluation of social media data where users communicate with multiple language mixtures. Accordingly, this research focuses on social media data (Facebook within this study) and attempts to develop a model to identify languages in code-mixed data. However, this study focuses only on Sinhala-English code-mixed data and for the best of author's knowledge this is the first study to research and develop such a model in Sinhala-English mixed data at the point of this paper is been developed. Therefore, there are no similar models to be compared with and this study will evaluate multiple techniques based on literature.

II. SIMILAR STUDIES

Although the language detection across documents is considered as a solved task, it is not the same with code-mixing. Nevertheless, there are many researches in language detection in code-mixing predominantly in Indian language context for Hindi and English code-mixing, Bangla and English code-mixing [1] [2] and for Hindi, Bangla and English code-mixing [3]. Accordingly, these studies highlighted that the language identification at word-level is a challenging task which is non-trivial specially in noisy and translated data found in social media platforms.

Accordingly, [2] is one of the leading studies conducted to address the automatic language detection problem in word level for code-mixed data found in social media. This study was conducted for Hindi-English and Bengali-English code-mixed data and the authors have used a dictionary-based classifier as the baseline mode, SVM classifier with 4 kinds of features; weighted character n-grams, dictionary features, minimum edit distance weight and word context information. Accordingly, the best performing model results a high precision more than 90% for Hindi-English data and 87% for Bangla-English data. Yet, the model has shown low recall of 65% and 60% respectively for two language mixtures with an overall F1 score of 76% and 74% respectively.

Further, [1] has introduced a language identification model for Hindi-English code-mixed data with a part-of-speech (POS) annotation system on social media data. Accordingly, the study has used a word-level logistic regression model for model training with 3,201 English words scraped from SMS data and with a separate Hindi corpus of 3,218 words. Even though the given model results a F1 score of 87%, it shows a low recall for Hindi data.

On the other hand, [4] presented a system to identify languages in Bangla-English code-mixed data. Accordingly,

the study has used two different datasets, one form FIRE 2013 and the second dataset from Facebook chat history. Similarly, the best performing model uses Bangla and English dictionary, n-gram and the percentage of surrounding words predicted as Bangla using the dictionary as features. Likewise, the proposed model gives a F1 score of 91.5% for Facebook chat dataset and 91.5% for FIRE data set.

A well performing model has been introduced by [3] for language identification in code-mixed data for Indian languages. This study has used multilingual dataset of Hindi-English-Bangla and the authors have collected 2,335 Facebook posts and 9,813 comments from Facebook groups for their experiment. Likewise, the best performing model of the study was a CRF model which was trained with 5 different features named; character n-gram, presence/absence in dictionaries, word length, capitalization and contextual information. Accordingly, the final system shows a 95.76% accuracy. On the other hand, the study has also tried an SVM classifier with the same features and has been able to get an accuracy of 95.52%.

Similarly, [5] introduced a new study to identify languages in code-mixed data with decision trees and SVM classifiers. Likewise, the study was presented using Assamese-English-Hindi code-mixed data for the first time in the domain. Accordingly, the study has collected 4,768 Facebook comments with a total of 20,781 tokens after manual annotation task. The proposed model has used only three features named; word unigrams, prefixed and suffixes and finally the contextual information. On account of the results presented within the study, SVM model outperformed the Decision tree model with an accuracy of 96.01%.

Further, [6] have studied language identification in code-mixed data using a dataset with 30 languages. In this study, 6 features have been used individually as well as with combinations of multiple features. Accordingly, character unigrams, bigrams, trigrams, 4-grams, 5-grams, and the full word have used as features to train their models. Likewise, the study has used CRF model trained with GE (generalized expectation) model, HMM (Hidden Markov Model) trained with EM (Expectation Maximization) and finally a logistic regression model trained with GE. Finally, each model has been compared with Naïve Bayes model as the baseline model. Accordingly, CRF model trained with GE has been able to outperform all the other models with an overall accuracy of 95%.

In contrast, [7] has studied code-mixing data with the aim of language identification using audio data and with their translated text data. Accordingly, the study has used 242,475 words of text in English and Spanish languages. Similarly, the authors have used word n-grams, character n-grams and character prefixes/suffixes as the text features in the study. Moreover, CRM model, logistic regression and a deep neural network model named LSTM has been used in the study. Based on the results, the character level and word level features with the combination of CRF model has been able to outperform the deep neural network model LSTM with an accuracy of 91%.

A well-trained model has been introduced by [8] for Persian and Dari text data. Since the Dari is a low-resourced language, the authors have created a new dataset of 28,000 Dari sentences from an American news website. Like most of the scholars, the authors have used character n-gram and word

n-gram features in their training and SVM classifier has been evaluated with the identified features. Accordingly, an accuracy of 96% has been acquired with the proposed model. On the other hand, the study has also tried out of domain cross-corpus evaluation to test the discriminative models' generalizability, achieving 87% accuracy in classifying 79,000 sentences from the Uppsala Persian Corpus.

The work presented by [9] identified a langue detection model for code-mixed data found in Tweeter platform. Accordingly, the study has used a dataset of 1.1 million tweets in five different European languages collected using tweeter scraping methods. The authors have used weighted n-gram features for language identifications on post level and reported an accuracy of 92.4%. On the other hand, [10] investigated a language identification model at the utterance level on a dataset from one of the largest online communities in The Netherlands for Turkish-Dutch speakers. The study has accommodated dictionary-based language models, logistic regression models and linear-chain CRF in analysis and has been able to reach accuracy of 97.6%, but with a substantially lower accuracy on post level 89.5%, even though 83% of the posts were monolingual.

III. MOTIVATION OF THE STUDY

The scope of this study is to evaluate Sinhala-English code-mixing data written in English alphabetical characters rather than Sinhala Unicode data. That is, Unicode characters can be directly identified with their base boundaries and therefore, it is straightforward to identify the language in sentences with Unicode characters. Even though there are many studies conducted on code-mixing data analysis, those models cannot be directly applied to Sinhala-English code-mixing scenario. That is, there are many ambiguous words within Sinhala-English code-mixing. As presented in table 2, even though some of the tokens are present in English language, the meaning of each token is completely different when it comes to Sihala-Singlish code-mixing. On the other hand, Sri Lankan users tend to use some words with short forms like “prens” to represent “friends” and “okkk”, “oki”, “k” to represent “okay”. In addition to that, people also tend to use character “k” or “i” at the end of numbers which will eventually turned to Sinhala token. As an example, “100k” is used to indicate exactly 100 rather than 100,000 in common practice.

TABLE 2: AMBIGUOUS WORDS

Token	Annotation	Annotator justification
royal	Name	Name of a school in Sri Lanka
oke	Sinhala	Sinhala term for “that” in English
maxaa	Sinhala	Form of a complement in Sinhala
okkkk	English	Refers to “Okay” in English even though it is similar to “oke” term in Sinhala
sup	English	Short term used for “Support”
shape	English/Sinhala	Even though this is an English term, sometimes it is used in Sinhala to express terms like “never the less” or a compliment etc.
100k, 6k and 14k etc.	Sinhala	In practice, it refers to 100,000 and 6,000. But here it means “exactly 100 and exactly 6” in Sinhala based on the context
prenzz	English	Short term used English term “Friend”
4i, 5i	Sinhala	Refers to “exactly 4, exactly 5” in Sinhala

IV. METHODOLOGY

As stated earlier, this is the first Sinhala-English code-mixing analysis study done by the time of this paper is written, there are no proper datasets to be used. Therefore, a new dataset was created to conduct this study using Facebook data. Further, the study was divided into two parts as code-mixing detection and sequence tagging. Each of the studies have used multiple techniques and compared their performance to select the best performing model.

A. Dataset

A new data set was created by scraping Facebook chat history and publicly available page comments and posts [12]. Following a manual cleanup on the collected data, 7,500 sentences with 40,915 tokens were collected. All the sentences with only Unicode characters, and emoticons were removed to get sentences with English alphabetical characters.

The data was annotated by three undergraduate students who are fluent in Sinhala and English languages with the help of Google sheets. The annotation was done in two phases to annotate sentences with code-mixing and to annotate each word in code-mixed data with the language used by the token. In addition, spelling mistakes are common in any kind of communication and it may occur frequently in social media communication where the users are communicating in near real-time manner and social media users tend to use words in short form in practice which also leads to spelling mistakes within analysis. Therefore, each annotator was asked to avoid all spelling errors or short form of words (“Flm” for “Film”, “Tkt” for “Ticket” and “Tnx” for “Thanks” etc.) and annotate the erroneous word with its most probable language.

B. Annotation evaluation

Cohen’s Kappa measure is used to measure the annotation accuracy which provides the proportion of agreement beyond that expected by chance. That is, the achieved “beyond chance” agreement as a proportion of the possible “beyond chance” agreement [11].

C. Level 1 annotation

In the first level of annotation, each annotator was requested to annotate each of the sentence with the type of language mix used in the sentence. Accordingly, 5 tags were used in this annotation process as shown in table 3.

Accordingly, a total of 7,500 sentences were annotated in three batches of 1,500, 3,000 and 3,000 sentences per batch. Further, the annotations were carried out as one batch at a time and each sentence was annotated by all three annotators. Furthermore, when assigning a data batch to an annotator, the dataset was shuffled before the assignment to maintain the randomness. Each data batch was given with the instruction table (table 3) to guide annotators.

Once the annotation is finished for a given batch, inter annotator agreement was calculated for the batch. Table 4 shows Cohen’s Kappa value for each of the level 1 annotation batch. Final annotation of each sentence was decided based on the total agreement of all three annotators. Likewise, table 5 shows language wise summary of each data batch after their annotation. Even though 7,500 sentences were used for annotations, only 7,080(total count in table 5) sentences got total agreement (same annotation tag from all three annotators) and thus, the final level 1 annotated sentences

selected for level two annotations have 7,080 sentences with the annotations.

TABLE 3: LEVEL 1 ANNOTATION TAGS

Language type	Description	Example
English	Every single word in the text is in English language	Good morning
Singlish	Sinhala words written in English characters	Mama ennam
Sinhala (Unicode)	Sinhala words written in Unicode characters	සුඩ උද්‍යෝගක්
Code-mixed	Sinhala written in English (as Singlish), but there are some English words in the sentence	mama ennam film hall ekata
	Unicode characters with Singlish	ඔබ yanawa
	English words with Unicode characters	Dreams පුද්ගල හිතෙනවා
	English words with Unicode characters and Singlish	පුද්ගල හිතෙන dream ekak
Unknown	Anything other than the above	

TABLE 4: INTER ANNOTATOR AGREEMENT FOR LEVEL 1 ANNOTATION

Data batch ID	Number of sentences	Cohen's Kappa value
1	1,500	0.806948108
2	3,000	0.878913945
3	3,000	0.936152024

At the end of all three data batches, the inter annotation agreement calculation (Kappa value) was 0.88772595 which shows a high agreement between all three annotators. Table 5 illustrates the overall language mix in the total dataset of 7,500 sentences. Accordingly, Singlish is the dominant language used in the data set. Further, there are considerable amount of code-mixed data as well. Even though there are 476 English sentences, English language usage is negligible when considering the total dataset. On the other hand, there are only 13 sentences with the *unknown* annotation which is also negligible. A manual inspection was done to validate all the sentences with unknown tag, and it was clear that those sentences were either with one or more question marks (??) or a sentence with just a number. Furthermore, the dataset consists of single word sentences as well as which are important to the language detection study.

TABLE 5: OVERALL LANGUAGE USAGE

Language type	Sentence count
Singlish	4,691
Code-mixed	1,900
English	476
Unknown	13

D. Level 2 annotation

Level 2 annotation is the word level annotation of the code-mixed sentences selected from level 1 annotations.

Accordingly, all the sentences where all three annotators have annotated as code-mixed were selected to be used in the second annotation phase. Likewise, 1,900 code-mixed sentences were used for word level annotation. Since the dataset is created to be used for a language detection study in Sinhala-English code-mixed data, level 2 annotation was intended to annotate each word based on the language used. However, there were many named entities and some numeric characters in the middle of sentences. Therefore, the second level annotation was carried out with two main tags named *Sinhala* and *English* along with additional tags named *Unknown* and *Name*. Name tag was used to annotate all named entities and Unknown tag will be assigned to all the other cases which do not fall into above categories.

Annotators were assigned with the total data set with the annotation slots for each word in the sentence and the dataset consisted of some acronyms and named entities and the annotators were requested to annotate such words based on the base language of those tokens. The important part in this annotation is that the tag of each word does not depend only in the word itself. That is, the language tag depends on the surrounding words of the sentence (sentences with more than one word). Likewise, there can be many instances where a given token may appear as an English word in this context, but it may be a Sinhala word written using English characters (Singlish). As an example, the word “me” in “me ahanna” sentence, appears as an English word (me, myself). Yet, it is a Sinhala term for English word “hey”. That is, the true meaning of the sentence is “hey, listen” in practice. Therefore, annotators had to consider language tags of sounding words in the annotation.

Accordingly, 11,795 tokens were annotated in this annotation phase with their base language. Final annotation of each word was decided based on the majority vote in this annotation level to avoid removal of words in the middle of a sentence (to avoid loss of context clues of each word) if the word fails to achieve a total agreement. Since, there were no annotations without a majority vote, the overall inter annotation agreement Kappa statistics for level two annotation was 1. Likewise, 8,568 Sinhala tokens, 2,824 of English tokens, 350 Name tokens and 53 unknown tokens were identified by the annotation.

In this annotation phase, Sinhala words dominated the code-mixed dataset which is an expected scenario from Sri Lankan users. On the other hand, each Sinhala word mixed with English tokens were contributed to the whole sentence to be a code-mixed sentence in the first level annotation. Further, the domination of Sinhala tokens was also expected because, within the level one annotation results were dominated by Singlish sentences.

E. Code-mixed sentence classification

As the first part of the study, a multiple machine learning models were evaluated to classify distinguished Sinhala-English code from non-code-mixed data. Accordingly, 7,486 sentences with four annotations (Singlish, English, code-mixed and unknown) were used to train each model and BOG, word level TF-IDF, char-n-gram TF-IDF, n-gram TF-IDF, pre-trained word embedding (Genism Word2Vec) model and a word embedding model trained with the dataset were tried with each machine learning model.

As machine learning models, Naïve Bayes model is used as the baseline model and logistic regression, SVM, Random Forest, XGB, shallow NN, deep NN, LSTM, CNN, recurrent CNN, bidirectional RNN and GRU models were tested with the dataset and optimized individually to select the best performing model.

F. Sequence tagging

Word level sequence tagging was done as the second part of the study. Accordingly, 1,900 Sihala-English code-mixed sentences resulted in 11,795 tokens. The annotations consisted of Sinhala, English, Name and unknown tags. Since the study is identifying each token with the language presented, all tokens annotated as Name are also considered as unknown. CRF, LSTM, SVM, K-nearest neighbor and random forest models were trained and optimized with the data set. Likewise, character n-gram, annotation of three to left side and right side of the word, capitalization and whether the token has a digit or not is used as features.

V. RESULTS

Fig 1 shows the comparison of all the models used for sequence tagging with Naïve Bayes model and fig 2 and fig 3 shows the comparison of non-neural network models and neural network models respectively. Accordingly, it is observed that the character n-gram gives the highest accuracy for most of the models. Further, multiple n-grams were tested and finally bigram was selected to use for all the models based on the accuracy.

When comparing all the models, XGB outperformed all the models with an accuracy of 92.1% with bigram features. On the other hand, it is also observed that all the neural network models were underneath the benchmark model accuracy and even the SVM model did not perform well compared to Naïve Bayes model. In addition, the best performing XGB model underperforms than the baseline model with n-gram-TF-IDF features and with BOG features.

Further, LSTM and GRU models performed better than other neural network models and it is also with character n-gram feature. On the other hand, deep neural network model has barely touched the 50% accuracy mark. Yet, other neural network models have been able to perform well compared to deep networks even though they were below the benchmark.

Fig 4, fig 5, fig 6 and fig 7 show the results for sequence annotation study with the identified machine learning models. Based on the results, CRF model outperformed all the other models in precision, recall as well as in F1 score which is a formulation of both precision and recall scores. Further, each model showed a high precision and recall for English tokens than Sinhala and unknown tokens. Accordingly, based on fig 7, CRF model results F1 score of 0.94, precision of 0.95 and a recall of 0.94 on average for all three tokens. Finally, the test output of the CRF model is manually inspected to check the ambiguous word labeling capability of the model and the CRF model has been able to label tricky words such as “100k, 5i and royal” based on the language usage.

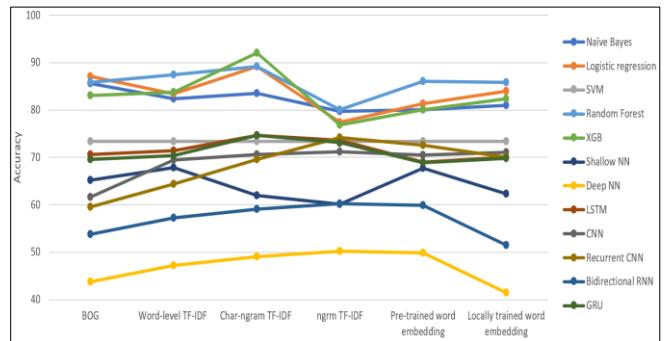


Figure 1: Comparison of all models for code-mixing classification

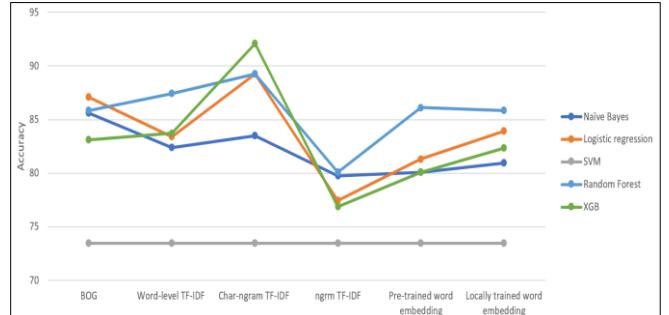


Figure 2: Comparison of non-neural network models for code-mixing classification

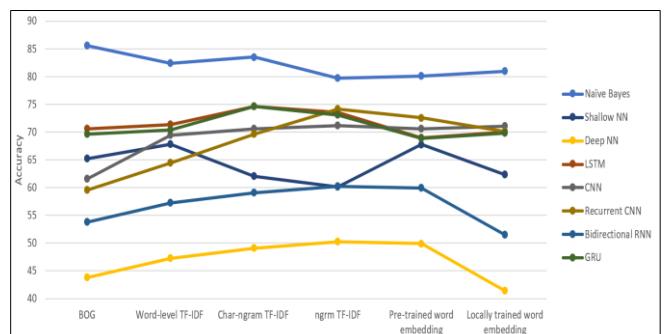


Figure 3: Comparison of neural network models for code-mixing classification

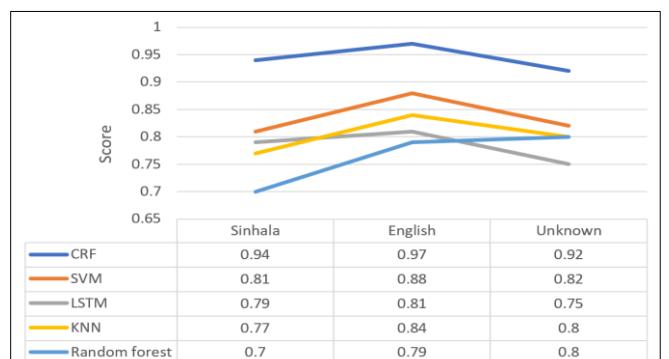


Figure 4: Precision scores

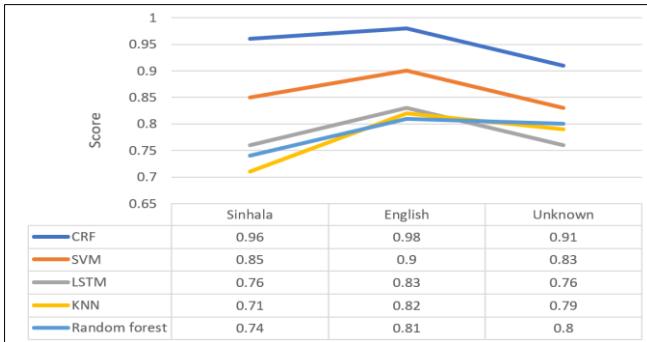


Figure 5: Recall scores

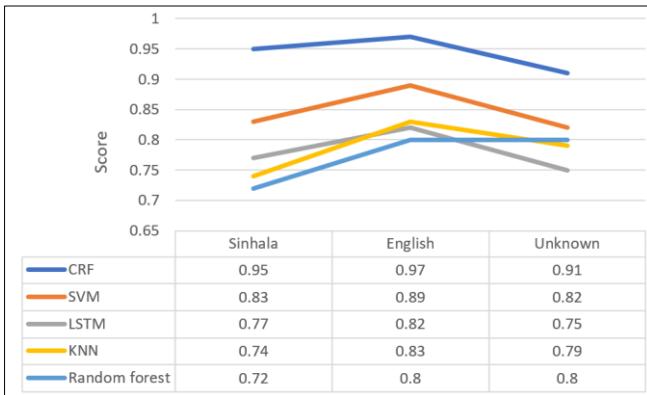


Figure 6: F1 scores

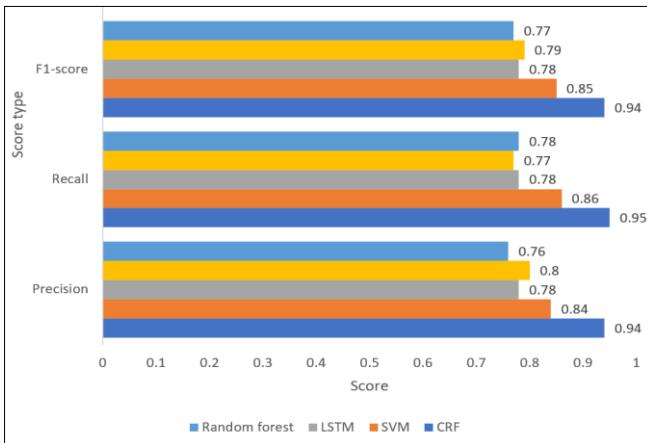


Figure 7: Average scores

VI. DISCUSSION

This study was an attempt to develop a high performing machine learning model to detect language of each token in Sinhala English code-mixed data. Since, this is the first attempt in Sinhala English code-mixed data analysis, a novel data set from Facebook chat history and public posts were created with manual annotation process resulting in high inter annotator agreement. Finally, a well optimized XGB model with an accuracy of 92.1% in code-mixed data classification and a CRF model with an overall F1 score of 0.94 was built in the study after comparing with multiple machine learning models. Even random forest model outperformed XGB in all other cases except for char-n-gram feature. Thus, it was recognized that tree-based models are more suitable for this kind of code-mixed data classification. However, it is deemed that the dataset used within this study is insufficient to train the neural network models to perform well with sequence tagging. In addition, multiple situations were identified in data

annotation process where spelling mistakes were identified and some cases where multiple words have been merged due to typing errors. Thus, the most appropriate method to deal such scenarios were to use n-gram methods and it was clearly visible in the results as well. As future studies, the dataset is expected to be expanded with more annotated data with multiple social media data other than Facebook. Further, combinations of multiple techniques like CRF with LSTM and CRF with GRU and some other ensemble combinations are to be examined as future study with the advancement of the dataset.

ACKNOWLEDGMENT

I express my sincere thanks to LK Domain Registry for the grant given for publishing this study and to the NLP center of university of Moratuwa for providing guidelines and knowledge at each step of the study.

REFERENCES

- [1] Das, A. and Gambäck, B. (2014). Identifying Languages at the Word Level in Code-Mixed Indian Social Media Text. In Proceedings of the 11th International Conference on Natural Language Processing.
- [2] Vyas, Y., Gella, S., Sharma, J., Bali, K. and Choudhury, M. (2014). POS tagging of English-Hindi Code-Mixed Social Media Content. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 974–979, Doha, Qatar, October. Association for Computational Linguistics.
- [3] Barman, U., Das, A., Wagner, J. and Foster, J. (2014). Code Mixing: A Challenge for Language Identification in the Language of Social Media. In Proceedings of the First Workshop on Computational Approaches to Code Switching.
- [4] Arunavha, C., Das, D. and Mazumdar, C. (2016). Unraveling the English-Bengali Code-Mixing Phenomenon. In Proceedings of the Second Workshop on Computational Approaches to Code Switching, pages 80 – 89.
- [5] Bora, M. J., & Kumar, R. (2018). Automatic word-level identification of language in assamese english hindi code-mixed data. In 4th Workshop on Indian Language Data and Resources, Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018) (pp. 7-12).
- [6] King, B., & Abney, S. (2013). Labeling the languages of words in mixed-language documents using weakly supervised methods. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 1110-1119).
- [7] Ramanarayanan, V., Pugh, R., Qian, Y., & Suendermann-Oeft, D. (2018). Automatic Turn-Level Language Identification for Code-Switched Spanish–English Dialog. In Proc. of the IWSDS Workshop.
- [8] Malmasi, S., & Dras, M. (2015, May). Automatic language identification for Persian and Dari texts. In Proceedings of PACLING (pp. 59-64).
- [9] Simon Carter. 2012. Exploration and Exploitation of Multilingual Data for Statistical Machine Translation. PhD Thesis, University of Amsterdam, Informatics Institute, Amsterdam, The Netherlands, December.
- [10] Dong Nguyen and A Seza Do˘gruoz. 2013. Word level language identification in online multilingual communication. In Proceedings of the 2013 EMNLP, pages 857–862, Seattle, Washington, October. ACL.
- [11] Cohen J. A coefficient of agreement for nominal scales. *Educ Psychol Meas*. 1960 ;20:37–46.
- [12] Smith, I. & Uthayasaneker, T. (2019). Sinhala-English Code-Mixed Data Analysis: A Review on Data Collection Process. In Proceedings of the 19th International Conference on Advances in ICT for Emerging Regions (ICTer2019)
- [13] Dilshani, W., Yashothara, S., Uthayasaneker, T., Jayasena, S. (2018) Linguistic Divergence of Sinhala and Tamil Languages in Machine Translation. In Proceedings of 2018 International Conference on Asian Language Processing (IALP) (pp. 13-18).

Sinhala and Tamil Speech Intent Identification From English Phoneme Based ASR

Yohan Karunananayake, Uthayasanker Thayavasivam, Surangika Ranathunga

Department of Computer Science and Engineering

University of Moratuwa

Katubedda 10400, Sri Lanka

{yohan.13, rtuthaya, surangika}@cse.mrt.ac.lk

Abstract—Today we can find many use cases for content-based speech classification. These include speech topic identification and spoken command recognition. Automatic Speech Recognition (ASR) sits underneath all of these applications to convert speech into textual format. However, creating an ASR system for a language is a resource-consuming task. Even though there are more than 6000 languages, all of these speech-related applications are limited to the most well-known languages such as English, because of the availability of data. There is some past research that looked into classifying speech while addressing the data scarcity. However, all of these methods have their own limitations. In this paper, we present an English language phoneme based speech intent classification methodology for Sinhala and Tamil languages. We use a pre-trained English ASR model to generate phoneme probability features and use them to identify intents of utterances expressed in Sinhala and Tamil, for which a rather small speech dataset is available . The experiment results show that the proposed method can have more than 80% accuracy for a 0.5-hour limited speech dataset in both languages.

Keywords-Low-Resource; Speech Intent Classification; Sinhala; Tamil;

I. INTRODUCTION

Automatic Speech Recognition (ASR) has evolved widely, and recent research shows human-level performance in some tasks [1]. Popular application areas of ASR include intent identification of spoken commands, topic identification of speech, and call center call classification [2]–[4]. For example, in a smart speaker use case, when a user issues a command such as “Play some music”, it is capable of identifying the intent of the given free-form command as a request to turn on the music. Here, the requirement is to identify speech, based on the spoken content. In the currently available topic or intent identification systems, this is enabled by cascading an ASR system and a Natural Language Understanding (NLU) system or a text classification model [2], [3].

The task of the ASR system is to convert a sequence of acoustic features into the most likely sequence of words. Normally Mel-frequency Cepstral Coefficients (MFCC) of the speech signals are used as input features. Earlier ASR models incorporated an acoustic model, a pronunciation lexicon that maps phones into words, and a language model to rank the likelihood of words [5]. Today we can see the use of deep neural network-based end-to-end ASR models [1], [6]. Advantage of these models is that they

are capable of modeling all the acoustic, pronunciation lexicon and language into a single model.

The output of the ASR module is the input for the NLU model. Then, the NLU model outputs semantic labels for a given text sequence, and it is trained with labeled data using supervised learning. Further, there is research that uses either an n-best list of the ASR output or the intermediate features of the ASR. This is to eliminate the errors made by using the single best output of the ASR [4], [7], [8].

Still, the ASR based applications are only available for most widely used languages, but not for low-resource languages (LRLs). Here, languages that lack electronic resources for speech and/or language processing are referred to as LRLs [9]. Because of the data scarcity, it is difficult to create robust ASR systems for LRLs. Normally more than 1000 hours of transcribed speech data is required to train an accurate ASR model [9]. Besides, the accuracy of the ASR model is a very important factor for the above described cascaded system for speech intent identification. Errors made by the ASR component can propagate into the NLU system and can result in false outputs [7]. Hence, this limits the development of speech intent identification systems for LRLs.

Recent research has presented some end-to-end speech intent identification models for languages such as English by utilizing transfer learning [4], [10]. Our previous work [11] presented the successful use of this transfer learning technique for LRLs using character probability values and shows good results. Further, there have been approaches to predict the intent using different features [2], [12].

In the past, researchers have suggested methods to create ASR for LRLs using phoneme annotated speech datasets [13]. Phonemes represent the perceptually distinct units of sound in a language, and it is closer to the sound representation compared to characters. Inspired by this and our previous work [11], we are presenting a phoneme-based domain-specific speech intent identification methodology in this paper. Experimental results show that phoneme based features perform better compared to the previous work [11]. Using Sinhala and Tamil speech data, we were able to reach an overall classification accuracy of 80% using not more than 500 speech samples.

II. RELATED WORK

The major issue in low-resource speech intent identification is the limited amount of annotated speech data. This restricts the development of robust ASR systems, and without ASR we cannot have speech intent identification. One obvious way to solve this is by compiling a sufficiently large speech corpus in the targeted language. However, this is a time and resource-consuming task [9]. Currently, we can observe few approaches that try to address this. One method is focused on improving the low-resource ASR, while another focuses on speech intent identification using different features other than ASR text output. This includes intermediate features of the ASR models.

In this first method, researchers have focused on developing ASR systems optimized for languages that have smaller speech corpora. One successful approach for this is adapting or retraining of an ASR system trained on a high resource language [9]. Another way is to use multiple smaller speech datasets and training ASR models with multitask-learning [9].

Some research focuses on topic identification of the speech. Here, the topics are similar to intents, but they vary in a broad range and tries to represent the whole subject presented in the spoken content. Work of Wiesner et al. [13] presents such topic classification methodology. They use a low-resource ASR development method and a multilingual speech corpus with universal phones annotations [14]. It has provided promising results when there is very limited training data. The text output of this ASR is used on a classifier model that can identify the corresponding topic. There are 11 different topics and some of them are “Evacuation”, “Food Supply”, “Urgent Rescue”, “Medical Assistance” and, “Shelter”. In this method, we need to have a phone-annotated multilingual speech corpus to train the ASR. Further, having a good understanding of the targeted LRL phonology is mandatory.

An ASR is trained to output the most probable word sequence for a given acoustic feature sequence [4]. Hence, when we use ASR generated text as input for others, the 1-best output of the ASR becomes input for the rest of the components. Here, it is difficult to ensure that the best output of the ASR is always a correct one. He et al. [8] and Yaman et al. [7] explored this in their works and proposed some approaches to overcome the issues by using the n-best list of the ASR output and joint optimization techniques. This still has the overhead of ASR development for the targetted language.

As mentioned above, in the second approach, features generated from the respective speech queries are directly used for intent/topic classification instead the final text output of the ASR. We can identify few prominent techniques in past literature. Liu et al. [2] and Wiesner et al. [13], used features such as phone-like units discovered via acoustic unit discovery (AUD) [15], [16], or word-like units discovered via unsupervised term discovery (UTD) [17]. These are unsupervised feature extraction methods and do not require speech data with transcripts or

annotations. However, these unsupervised methods require more data to identify better feature representations and more computational power to process data. In contrast to this, Buddhika et al. [12] presented a low-resource speech intent classifier that uses MFCC features directly. They use classifier models such as Support Vector Machines (SVM), and Convolution Neural Networks (CNN) to identify intents from MFCC features. This approach achieved a 74% classification accuracy for a 10 hour domain specific Sinhala dataset.

Chen et al. [4] presented an intent identification method for the English language queries using intermediate features of a pre-trained English ASR model. Here, they used character probability values generated by the ASR as features for a CNN based intent classification model, and obtained good results for call center call classification. Lugosch et al. [10] presented another such similar work and showed good results using a 14.7-hour dataset while utilizing the pre-training strategy. In their work, they identify not only the intent, but also the slot values such as action, object, and location mentioned in the speech query. In this way, we do not need to worry about 1-best output of the ASR, and can optimize jointly. However, in both of these works, an ASR trained on a large English corpus is used to identify intent on the same language.

Utilizing this pre-training strategy, our previous work [11] demonstrated a successful method for speech intent identification for LRLs. In this work, we used a pre-trained English model to identify the intent of low-resource Tamil and Sinhala speech commands. This method could reach to an overall accuracy of 80% using 1 hour of speech data containing 1000 samples.

In summary, previous research has tried to tackle low-resource speech intent identification via either developing low-resource ASR or using different features generated from the speech query. The latter approach can eliminate the issue of having an accurate ASR, since they rely on different input features other than the 1-best output of an ASR. Because of this reason, this method is much more suitable for low-resource scenarios.

III. METHODOLOGY

In our previous work, we used ASR generated character probabilities as features to identify intent. Characters are more language-specific and try to follow the syntactic and semantic rules of a particular language. Compared to characters, phonemes try to represent perceptually distinct units of sound in a specified language. Hence phonemes have more ability to represent sounds than the characters. In this work, we try to exploit this.

In Section II, we highlighted the benefit of the pre-training strategy. In this work, we use a pre-trained ASR model of a source language to generate phoneme based features for another language. Then we can use these features to identify the intent in the LRL. In machine learning paradigm, this is known as transfer learning, where we try to reuse a model trained on one task in another similar or related task [18]. Here, we use an ASR

model trained to convert high resource speech into text, in low resource speech intent identification. For a better understanding of the methodology, first, we introduce the character related features and then phoneme features.

A. Character Probabilities

In Section I, we mentioned that there are well-performing all neural end-to-end ASR models [1], [6]. Such an ASR model is trained to predict the most probable word sequence for a given sequence of acoustic features. To enable predicting words, most of the end-to-end ASR models produce character probability values in a given time step. Hence, when we input a sequence of acoustic features $x^{(i)}$, the model converts this into a sequence of character probabilities $y^{(i)}$, with $\hat{y}_t = \mathbb{P}(c_t|x)$ where c_t represents the possible character set including other special characters-(to represent spaces, silents) in the training language.

B. Phoneme Probabilities

In this work, our focus is to use phoneme probability values rather than character probabilities and examine the effectiveness. ASR systems that output a phoneme sequence for a given audio sequence are not so common. Most of the ASR systems are trained on characters, which makes it easy to predict probable word sequence. Lugosch et al. [10], presented an ASR system that uses phonemes as intermediate targets. This model outputs a sequence of phoneme probabilities $p^{(i)}$, $\hat{p}_t = \mathbb{P}(p_t|x)$ when given a sequence of acoustic features $x^{(i)}$ where p_t represents the possible phoneme set in the training language. Hence we use this model to generate phoneme based features.

C. Feature Classification Model

To identify a fixed set of intents, previous research has used a classifier model on features generated from audio recordings as described in Section II. They have experimented with models such as Support Vector Machines (SVM), Feed-forward Neural Networks (FFN), and Convolutional Neural Networks (CNN). Our previous work showed that SVM and CNN based classifiers work better with character probability features and highlighted the superior performance of CNN [11]. Further, there were two different CNN types: 1D CNNs and 2D CNNs which was experimented. In 1D type, the convolution and pooling operations are done along only on one dimension, while in 2D this happens along on two directions.

The above-described character/phoneme probability features are two dimensional and one dimension is used to represent time steps while other dimension represents different characters or phonemes. Visualization of these features is presented in Figure 2. For SVM models, these features need to be converted into a series format to provide as inputs. Further, when we use 1D CNN models, convolution and pooling operation are done along the time axis. For 2D CNN, those operations are done along both axes.

In summary, our speech intent identification methodology is as follows. First, we train a phoneme based

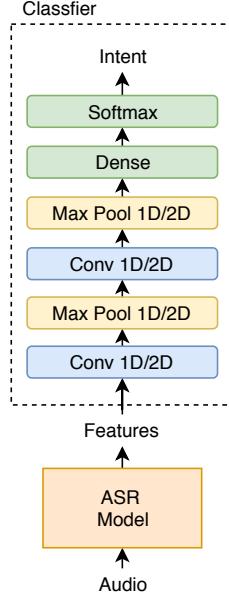


Figure 1. Arrangement of the final model

ASR using high resource language data. Then this ASR is used to generate features. Using these features we train different classifier models to identify the intent of the speech. Figure 1 shows the complete arrangement of the system with the pre-trained ASR and the CNN classifier.

IV. DATASET

To evaluate the performance of the proposed approach, we use the same dataset used in our previous work [11]. It includes Sinhala and Tamil speech data related to the banking domain with intent labels. This Sinhala and Tamil data has been collected respectively from 215 and 40 students. Datasets contain both male and female voices. All the contributors were in the age between 20 to 25 years. Further all these audio clips have been collected through mobile phones via crowd sourcing. Because of that we can expect domestic noises in the audio. In the Sinhala dataset, all the queries are expressed in Sinhala. However, Tamil queries contain some code-mixed speech with English terms. Table I shows the statistics of the dataset. In the table, 'I' represents the number of infections, i.e. different ways of expressing intents. 'S' represents the number of samples.

V. EXPERIMENT

Training an ASR model on a large dataset requires high computational power. Hence we used already trained openly available ASR models for the experiments. To get the character probability features, we used the DeepSpeech (DS)¹ model [1]. This DS model has been trained on the Common Voice American English corpus, and reports a 11% word error rate on the LibriSpeech clean test corpus. To get phoneme based probability values, we adapted

¹DeepSpeech Model
<https://github.com/mozilla/DeepSpeech>

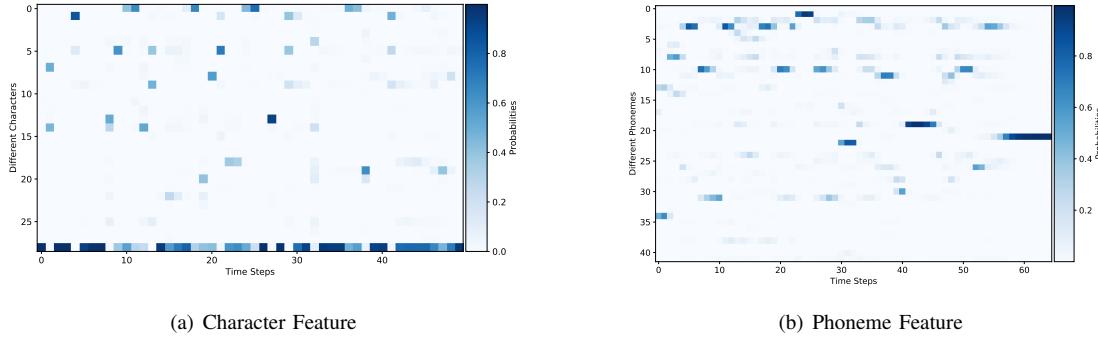


Figure 2. Visualization of features

Table I
DETAILS OF THE DATASETS (I-INFLECTIONS, S-NUMBER OF SAMPLES)

Intent	Sinhala		Tamil	
	I	S	I	S
1. Request Acc. balance	8	1712	7	101
2. Money deposit	7	1306	7	75
3. Money withdraw	8	1548	5	62
4. Bill payments	5	1004	4	46
5. Money transfer	7	1271	4	49
6. Credit card payments	4	795	4	67
Total	39	7624	31	400
Unique words	32		46	
Size in hours	7.5		0.5	

the pre-trained ASR model of lugosch et al. [10]². This model was trained on the LibriSpeech 690 hour English corpus [10].

Then we extracted the probability features using the ASR models and fed them into the classifier models for training. For character probability features, character set of $\{a, b, c, \dots, z, space, apostrophe, blank\}$ is used since it is trained on English speech. The phoneme set had a set of 42 symbols that includes the ARPAEnglish phoneme set (39 phonemes), and 3 non-speech annotations [10].

Since there is a limited amount of data, we employed 5-fold cross-validation to measure the overall classification accuracy. Models are trained up to the maximum accuracy without getting over-fitted into the training data set. Additionally, we used the Bayesian optimization-based algorithm for hyper-parameter tuning [19]. The optimization algorithm is employed with 500 iterations to select the suitable hyper-parameters, which improves the overall accuracy. This was very significant for the CNN model parameters such as the number of filters and kernel sizes. For the SVM models, a linear kernel is used after experimenting with several different kernels types (Polynomial, Radial Basis Function (RBF)). Table II shows the final overall classification accuracy of different classifier models and a comparison between previous work.

Further, we evaluated the overall accuracy change with

respect to the number of available training samples. To do that we drew a random data sample with a particular size, and performed 5-fold cross-validation. We did this for 20 times to get the average accuracy for a given sample size. This task was performed on the Sinhala dataset since it contained more than 5000 samples. Figure 3 summarizes these results. In Figure 3, connected lines represent the overall accuracy change in the Sinhala data. Points on the vertical line represent the Tamil dataset accuracy values with 400 data samples.

Table III presents the most probable character/phoneme sequences for the two selected sentences in Sinhala and Tamil languages. For better understanding, we present 39 phonemes with their IPA (International Phonetic Alphabet) notation.

VI. DISCUSSION

Overall results presented in Table II, and the graph in Figure 3 emphasize that phoneme probability features are more effective for speech intent identification compared to character features. For Sinhala and Tamil datasets, the proposed method achieves an overall accuracy of 97.38% and 81.70%, respectively. Further, these values indicate the usefulness of phoneme probability features despite the targeted LRL. According to the Figure 3, having 500 is enough to reach up to more than 80% accuracy. It needs more than 1000 data samples to achieve similar results using character probability features [11].

For comparison purposes, we marked the Tamil dataset results in the graph presented in Figure 3. Disconnected dots on the dashed-line represent the Tamil dataset experiment results for a 400 data sample. There we can see that the accuracy for the Tamil dataset also lies close to the Sinhala dataset trend line. However, when we examined closely, we can find one exception - 2D CNN shows a similar accuracy for both character and phoneme features in Tamil.

In the Tamil dataset, 61% of the sentences contain code-mixed speech queries with at least 1 English word. This can be a reason for such a higher result. In contrast to this, we cannot identify such anomaly results with phoneme features or 1D CNN character feature results. Further, this can happen because of having proper hyper-parameters for the 2D CNN character model. Because of the limited

²Phoneme Based Model
https://github.com/lorenlugosch/pretrain_speech_model

Table II
SUMMARY OF RESULTS

Features	DS Character Prob [11]			Phoneme Prob		
Classifier	SVM	1D CNN	2D CNN	SVM	1D CNN	2D CNN
Accuracy Sinhala	70.04%	93.16%	92.09%	78.21	97.31%	94.16%
Accuracy Tamil	23.77%	37.57%	76.30%	49.83	81.70%	76.28%

Table III
MOST PROBABLE CHARACTERS AND PHONEMES FOR A GIVEN UTTERANCE

data, it is difficult to identify the effect of code mixing. However, in general phoneme probability features give better results compared to character probability features regardless of the language.

We can identify some common patterns when we closely examine the most probable character or phoneme sequences presented in the Table III. Most of the time these patterns do not occur sequentially, there are some other symbols in between them. The intent identification model trained on probability features tries to identify those hidden patterns. These sequential patterns can be affected by the language model of the high resource training language. Here, the ASR models used to generate features are based on the Recurrent Neural Networks (RNNs). and they are capable of language modeling [20], [21]. Therefore the ASR models try to predict character or phoneme sequences as observed in the training language (English in this case).

This effect is more visible in character probability outputs in the Table III. Here, first few characters with corresponding sounds have been detected. Sometimes it has predicted the English words with a similar sound. However, when it comes to the middle and end of a sequence, it is difficult to find any patterns and all look random. This is quite different for the phoneme sequence. If we inspect the generated symbol sequence for the Tamil sentence, even in the middle we can find some patterns. Hence, phoneme-based features gives a better representation compared to character-based features for a given audio. Consequently, we can observe higher results.

The next visible effect on results is the performance difference of the 1D and 2D CNN models. When there is limited data, 2D CNN outperforms 1D CNN in character probability features. This is changed when training data size increases. With phoneme features, 1D CNN always outperforms 2D CNN. However, when we examine the feature visualization for intensity points (most probable symbol), these values change quite rapidly in the character feature map compared to phonemes. Additionally, we were able to identify visible patterns inside the generated phoneme sequences. Hence 1D CNN can perform better

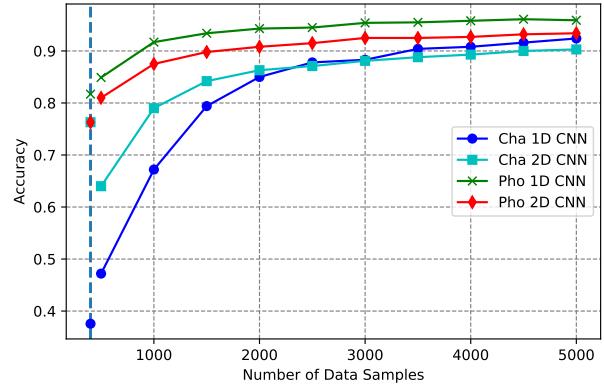


Figure 3. Overall accuracy change with the samples size (Connected dots - Sinhala, Disconnected dots - Tamil

with phoneme features. There are more rapid distortions in character features. Hence, the 2D CNN may be more useful for character probability features.

VII. CONCLUSION AND FUTURE WORK

In this paper, we presented an English phoneme based feature generation and intent identification method for Sinhala and Tamil speech data using a pre-trained English ASR model. We can identify the expressing intent of speech queries more accurately using these features. To evaluate the effectiveness of the proposed method, we used two datasets from the two different languages in the banking domain, which have a limited number of data samples. Experiment results show that phoneme base probability features are more effective compared to character features in low resource scenarios. Additionally, we observed that 1D CNN models perform better compared to 2D CNNs in classifying phoneme based features. The proposed method can reach to an 80% accuracy even with a dataset that has 0.5 hours of speech data.

In the future, we hope to extend this work using different language datasets to examine the effectiveness and the generalizability across languages, and the effect of having code-mixed speech.

ACKNOWLEDGMENT

This research was funded by a Senate Research Committee (SRC) Grant of University of Moratuwa.

REFERENCES

- [1] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, “Deep speech 2: End-to-end speech recognition in english and mandarin,” in *International conference on machine learning*, 2016, pp. 173–182.
- [2] C. Liu, J. Trmal, M. Wiesner, C. Harman, and S. Khudanpur, “Topic identification for speech without asr,” in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2017-August, 2017, pp. 2501–2505.
- [3] A. Ram, R. Prasad, C. Khatri, A. Venkatesh, R. Gabriel, Q. Liu, J. Nunn, B. Hedayatnia, M. Cheng, A. Nagar *et al.*, “Conversational ai: The science behind the alexa prize,” *arXiv preprint arXiv:1801.03604*, 2018.
- [4] Y.-P. Chen, R. Price, and S. Bangalore, “Spoken language understanding without speech recognition,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6189–6193.
- [5] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, “The kaldi speech recognition toolkit,” IEEE Signal Processing Society, Tech. Rep., 2011.
- [6] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, “Deep speech: Scaling up end-to-end speech recognition,” *arXiv preprint arXiv:1412.5567*, 2014.
- [7] S. Yaman, L. Deng, D. Yu, Y.-Y. Wang, and A. Acero, “An integrative and discriminative technique for spoken utterance classification,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 6, pp. 1207–1214, 2008.
- [8] X. He and L. Deng, “Speech-centric information processing: An optimization-oriented approach,” *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1116–1135, 2013.
- [9] L. Besacier, E. Barnard, A. Karpov, and T. Schultz, “Automatic speech recognition for under-resourced languages: A survey,” *Speech Communication*, vol. 56, pp. 85–100, 2014.
- [10] L. Lugosch, M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio, “Speech model pre-training for end-to-end spoken language understanding,” *arXiv preprint arXiv:1904.03670*, 2019.
- [11] Y. Karunanayake, U. Thayasivam, and S. Ranathunga, “Transfer learning based free-form speech command classification for low-resource languages,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, 2019, pp. 288–294.
- [12] D. Buddhika, R. Liyadipita, S. Nadeeshan, H. Witharana, S. Jayasena, and U. Thayasivam, “Domain specific intent classification of sinhala speech data,” in *2018 International Conference on Asian Language Processing (IALP)*. IEEE, 2018, pp. 197–202.
- [13] M. Wiesner, C. Liu, L. Ondel, C. Harman, V. Manohar, J. Trmal, Z. Huang, N. Dehak, and S. Khudanpur, “Automatic speech recognition and topic identification for almost-zero-resource languages,” in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2018-September, 2018, pp. 2052–2056.
- [14] K. M. Knill, M. J. Gales, A. Ragni, and S. P. Rath, “Language independent and unsupervised acoustic models for speech recognition and keyword spotting,” in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2014, pp. 16–20.
- [15] L. Ondel, L. Burget, and J. Černocký, “Variational inference for acoustic unit discovery,” *Procedia Computer Science*, vol. 81, pp. 80–86, 2016.
- [16] C. Liu, J. Yang, M. Sun, S. Kesiraju, A. Rott, L. Ondel, P. Ghahremani, N. Dehak, L. Burget, and S. Khudanpur, “An empirical evaluation of zero resource acoustic unit discovery,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5305–5309.
- [17] A. Jansen and B. Van Durme, “Efficient spoken term discovery using randomized algorithms,” in *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*. IEEE, 2011, pp. 401–406.
- [18] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [19] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox, “Hyperopt: a python library for model selection and hyperparameter optimization,” *Computational Science & Discovery*, vol. 8, no. 1, p. 014008, 2015.
- [20] T. Mikolov, M. Karafiat, L. Burget, J. Černocký, and S. Khudanpur, “Recurrent neural network based language model,” in *Eleventh annual conference of the international speech communication association*, 2010.
- [21] A. Graves and N. Jaitly, “Towards end-to-end speech recognition with recurrent neural networks,” in *International conference on machine learning*, 2014, pp. 1764–1772.

An experimental Tagalog Finite State Automata spellchecker with Levenshtein edit-distance feature

Joseph Marvin R. Imperial

College of Computing and Information Technologies
National University
Manila, Philippines
jrimperial@national-u.edu.ph

Czerittonnie Gail V. Ya-On, Jennifer C. Ureta

College of Computer Studies
De La Salle University
Manila, Philippines
czerittonnie_ya-on@dlsu.edu.ph,
jennifer ureta@dlsu.edu.ph

Abstract— In this paper, we present an experimental development of a spell checker for the Tagalog language using a set of word list with 300 random root words and three inflected forms as training data and a two-layered architecture of combined Deterministic Finite Automaton (DFA) with Levenshtein edit-distance. A DFA is used to process strings to identify if it belongs to a certain language via the binary result of accept or reject. The Levenshtein edit-distance of two strings is the number (k) of deletions, alterations, insertions between two sequences of characters. From the sample trained wordlist, results show that a value of 1 for the edit-distance (k) can be effective in spelling Tagalog sentences. Any value greater than 1 can cause suggestion of words even if the spelling of words is correct due to selective and prominent usage of certain characters in the Tagalog language like a, n, g, t, s, l.

Keywords— Deterministic Finite State automata, spell checker, Tagalog, Levenshtein edit-distance

I. INTRODUCTION

A spell checker is a tool used to identify a misspelled word based on a selected training set. A generated list of spelling suggestions is created upon detection of a misspelled word. Using Levenshtein edit-distance, the word with the minimum number of character pair transpositions is selected. The Tagalog language can have several possible inflections and morphological rules. It is not efficient to examine all the words in Tagalog to model training. Thus, encoding all the rules in the automaton is not practical. Rules are created by using training set to train the automaton.

The finite state machine (FSM) created is the input Filipino alphabet. Each character changes the state of the model. If the final character put the model is accepted by the final state, then the word belongs to the language. A Deterministic Finite Automaton

In this study, a Levenshtein automaton model, a DFA with Levenshtein-distance feature, was trained with a Tagalog wordlist consisting of 300 words and three of its inflected forms. Once trained, varying values of k-edit distance has been explored to identify the effectiveness of employing such model to a morphologically rich language.

II. REVIEW OF RELATED LITERATURE

Filipino is a morphologically rich language whose word combination uses different verb affix such as prefix, infix, suffix, and syllable or word duplication. [1] To search a Filipino word in a dictionary, the keyword should

consider the tense associated with the action word. The past (*pangnangdaan*), present (*pangasalukuyan*), and future (*panghinaharap*) tense form are treated different entries in a dictionary system. Say, the word “*punta*” which means “go to” is a different entry from the word “*pumunta*” meaning “went to or move forward”. It can be space-wasting to create an individual entry for each inflection. Moreover, there is difficulty in maintaining such a system.

The existing Filipino spellcheck solution applied the entire spelling rule and guidelines, namely, Komisyon sa Wikang Filipino 2001 Revision of the Alphabet and Guidelines in Spelling Filipino Language (or KWF), and the Gabay sa Editing Wikang Filipino (or GABAY) rulebooks into the system. The spellchecker carried out its tasks through manual-formulated and learned rules. [6]

Development of spell checker using finite state machines was also explored and applied to Malayalam, a morphological-rich Dravidian language spoken in the Indian state of Kerala [7]. A finite state transition model was generated using a predefined training set of Malayalam root words and their inflected forms. The FSM was trained in such a way that it can recognize the root words and its inflections and if correctly spelled, the string word is accepted by the final state of the FSM.

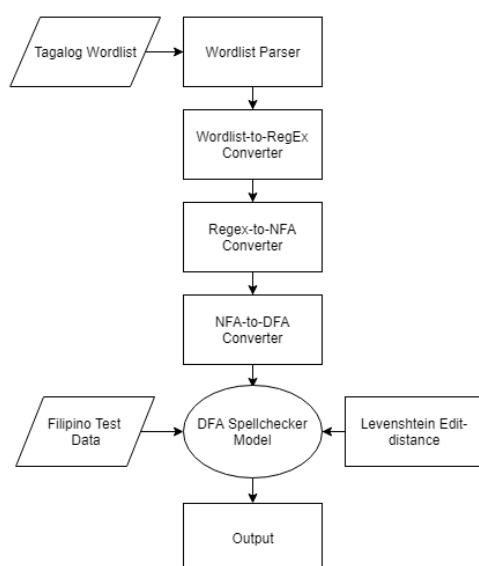


Fig 1. Methodology for Spellchecking using DFA with Levenshtein edit-distance.

III. METHODOLOGY

The research follows a step-by-step methodology as shown in Figure 1.

A. Word List Data

For the training data, a sample Tagalog wordlist was used containing 300 Tagalog root words with its inflected forms. Inflected forms are stemmed from the root words added with suffixes connoting various factors in the Tagalog language like repetition, action, and time. For the training set of wordlists, three inflected forms are used. However, additional forms can be added. Figure 2 shows a sample of Tagalog words list used for training. All root words have corresponding three or more inflected forms.

Root	Form 1	Form 2	Form 3
kain	kainan	kainin	kaingin
kagat	kagatin	kagatan	kagatum
baba	babain	babasagin	babashin
madali	madaliin	madalian	madaling
abut	abutin	abutan	abut-abot
bali	baliin	balian	balikan
galaw	galawin	galawgaw	galawgawin
gulu	gulugod	guluhin	gulonggulungan
hasa	hasaan	hasahasa	hasang
hati	hatiin	hatian	hatid
ipun	ipunas	ipundar	ipunan
kuskos	kuskusin	kuskusan	kuskosalungos
aral	aralin	aralan	araling
tugtug	tugtugin	tugtugan	tugtuging
tugtug	tugtugin	tugtugan	tugtuging
bili	bilihan	bilik	bilinan
benta	betahan	bentahin	bentahen
kabit	kabitan	kabitin	kabituinan
lito	litograpo	litograpiko	litograpia
takip	takipin	takipmata	takipsilim
baba	babae	babad	babaan

Fig 2. The sample Tagalog wordlist training data

B. Parsing Wordlist to RegEx

To make the wordlist understandable by a Deterministic Finite Automata, it is essential to convert it to a regular expression. Regular Expressions, or RegEx, describe regular languages. It can be used to determine if a given string or sequences of characters are accepted by a language [2]. Each set of the root word and inflected forms have their own RegEx group to be combined as one for a single pass to the RegEx-to-NFA converter.

C. RegEx to Nondeterministic Finite Automaton (ϵ -NFA)

The first phase in the conversion process is transforming the input RegEx string to an NFA with epsilon transitions. Using Thompson's Construction algorithm [3], each expression will be split into subexpressions. These subexpressions can be categorized into one of the following rules in Thompson's construction algorithm based on its operator. The algorithm for converting the RegEx to ϵ -NFA is described below.

For every character in the RegEx string, the converter identifies if it is a member of the predefined alphabet or not, or if it is one of the operators and closures accepted.

Three separate Python lists were created functioning as stacks to keep track of the conversion. One is for storing the operators, the second contains the Thompson construction for every alphabet in the automata, and the last records all the alphabet in the RegEx.

For every operator (union or concatenation) or in the stack, two objects from the automata stack will be popped to be processed.

For every closure property, the last object in the automata stack will be popped and passed to one of the three available Thompson Construction function for conversion. Once processed, the converted forms will be appended back to the automata stack

D. Nondeterministic Finite Automata with Epsilon Transitions (ϵ -NFA) to Deterministic Finite Automata (DFA)

After processing every alphabet, operators, and closure from the RegEx string to an ϵ -NFA, it will now be converted into a Deterministic Finite Automaton (DFA).

The algorithm implemented by the proponents for the said conversion is as follows:

1. Iterate through every character of the language (previously saved from ϵ -NFA).
2. The ϵ -closure for the beginning states of NFA are taken as beginning states of DFA.
3. The states that can be traversed from the present to each symbol (union of transition value and their closures for each states of NFA present in the current state of DFA and including itself) are searched. If any new state is found, it is considered as current state.
4. The search is continued until there is no new state present in DFA transition table.
5. The states of DFA which contains final states of NFA are marked as final states of DFA.

The final output of the DFA will be responsible for first-tier spell checking of the sample Tagalog test sentences to identify if the words are recognized by the language.

E. The Levenshtein Edit-Distance Feature

The Levenshtein edit-distance is a metric for identifying the number of changes via character edits of insertions, deletions, and alterations of one source string from the other. [4] An example of Levenshtein edit-distance calculation is shown below.

1. $kainan \rightarrow kainin$ (edit-distance of 1 by substituting 'a' to 'i')
2. $bago \rightarrow bagay$ (edit-distance of 2 by substituting 'o' to 'a' and appending 'y')
3. $gamitin \rightarrow gamit$ (edit-distance of 2 by deleting the last two characters of the first string)

For the implementation of the Tagalog DFA spellchecker with Levenshtein automaton, the applicable edit-distance value was experimented using the values 1 and 2.

F. Suggesting Possible Spellings

After the first tier and second tier spellchecking of the DFA and Levenshtein edit-distance feature, possible suggested spellings are printed out as an additional feature. Since the developed spell checker program recognizes misspelled words from the trained wordlist, it can also suggest words based on the wordlist based on the number of edit-distance. For Tagalog, an edit-distance of 2 and 1 was explored.

IV. RESULTS AND DISCUSSION

The collected Tagalog wordlist containing 300 sets each with a single root word and three inflected forms were trained using the specified methodology. The total number of states generated for the DFA using the various converters was 226, with 79 final states. The DFA performs the first-tier spell checking by indicating if each word is accepted by the language. The second-tier spell checking is performed by the Levenshtein edit-distance feature of the DFA. Figures 3, 4, 5, and 6 below shows the checking and suggestive capabilities of the developed program using DFA and Levenshtein edit-distance using sample sentences.

Test Sentence A = “*Huwag mong gulohin and tinupi kong mga damit.*” (*Don’t mess with my folded clothes*)

Test Sentence B = “*Pwede bang abotan mo ako ng tubig?*” (*Can you pass me some water?*)

```
Enter string: Huwag mong gulohin ang tinupi kong mga damit.

-----
SPELLING SUGGESTIONS:
-----
Huwag mong gulohin ang tinupi kong mga damit.
gulohin = ['guluhin']
```

Fig 3. Spell checking Test Sentence A with edit-distance of 1

```
Enter string: Pwede bang abotan mo ako ng tubig?

-----
SPELLING SUGGESTIONS:
-----
Pwede bang abotan mo ako ng tubig?
abotan = ['abutan']
```

Fig 4. Spell checking Test Sentence B with edit-distance of 1

```
Enter string: Huwag mong gulohin ang tinupi kong mga damit.

-----
SPELLING SUGGESTIONS:
-----
Huwag mong gulohin ang tinupi kong mga damit.
gulohin = ['guluhin']
damit. = ['kabit']
```

Fig 5. Spell checking a Test Sentence B with edit-distance of 2

```
Enter string: Pwede bang abotan mo ako ng tubig?

-----
SPELLING SUGGESTIONS:
-----
Pwede bang abotan mo ako ng tubig?
bang = ['baba', 'bali', 'baba']
abotan = ['abutin', 'abutan', 'kabitan']
```

Fig 6. Spell checking a Test Sentence B with edit-distance of 2

From the results of spellchecking of sample sentences, the number of spelling suggestions may increase with the number of Levenshtein edit-distance used. With this, even words of correct spelling may be highlighted as incorrect and other words of unrelated meaning can be suggested. In view with this, the proponents inferred that the main cause of this is how alphabet is used in the Tagalog language. For the Tagalog language, the top 10 characters widely used are shown below. The list of most frequently occurring characters in the Tagalog language from a 2,151,963-character (368,905 words) document was curated by Stefan Trost of Word Creator [5].

TABLE I. MOST FREQUENTLY USED CHARACTERS IN THE TAGALOG LANGUAGE

Character	Frequency
A	24.25%
N	11.77%
G	8.51%
I	7.89%
S	5.6%
T	4.87%
M	4.27%
O	4.19%
L	3.77%
K	3.61%

V. RESULTS AND DISCUSSION

The collected Tagalog wordlist containing 300 sets each with a single root word and three inflected forms were trained using the specified methodology. The total number of states generated for the DFA using the various converters was 226, with 79 final states. The DFA performs the first-tier spell checking by indicating if each word is accepted by the language. The second-tier spell checking is performed by the Levenshtein edit-distance feature of the DFA. Figures 3, 4, 5, and 6 below shows the checking and suggestive capabilities of the developed

program using DFA and Levenshtein edit-distance using sample sentences.

REFERENCES

- [1] Roxas, R.R, & Mula, G.T, "A morphological analyzer for Filipino verbs," 22nd Pacific Asia Conference on Language, Information and Computation, 2008.
- [2] Regular expression. (2018, December 11). Retrieved from https://en.wikipedia.org/wiki/Regular_expression
- [3] Anon. 2017. Theory of Computation | Minimization of DFA. (May 2017). Retrieved October 23, 2018 from <https://www.geeksforgeeks.org/program-implement-nfa-epsilon-move-dfa-conversion/>
- [4] Levenshtein, Vladimir I. (February 1966). "Binary codes capable of correcting deletions, insertions, and reversals". Soviet Physics Doklady. 10 (8): 707–710. Bibcode:1966SPhD...10..707L.
- [5] Trost, S. (n.d.). WordCreator. Retrieved from <https://www.sttmedia.com/characterfrequency-filipino>
- [6] C. Cheng, C. P. Alberto, I. A. Chan, and V. J. Querol, "SpellCheF: Spelling Checker and Corrector for Filipino," Journal of Research in Science, Computing and Engineering, vol. 4, no. 3, 2008.
- [7] N. Manohar, P. T. Lekshmipriya, V. Jayan, and V. K. Bhadran, "Spellchecker for Malayalam using finite state transition models," 2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS), 2015.

Statistical Machine Learning for Transliteration: Transliterating names between Sinhala, Tamil and English

H.S. Priyadarshani, M.D.W. Rajapaksha, M.M.S.P. Ranasinghe, K. Sarveswaran, G.V. Dias

Department of Computer Science and Engineering

University of Moratuwa

Katubedda 10400, Sri Lanka

{shanika.15, wimukthi.15, sharmila.15, sarves, gihan}@cse.mrt.ac.lk

Abstract—In this paper, we focus on building models for transliteration of personal names between the primary languages of Sri Lanka - namely Sinhala, Tamil and English. Currently, a Rule-based system has been used to transliterate names between Sinhala and Tamil. However, we found that it fails in several cases. Further, there were no systems available to transliterate names to English. In this paper, we present a hybrid approach where we use machine learning and statistical machine translation to do the transliteration. We built a parallel trilingual corpus of personal names. Then we trained a machine learner to classify names based on the ethnicity as we found it is an influencing factor in transliteration. Then we took the transliteration as a translation problem and applied statistical machine translation to generate the most probable transliteration for personal names. The system shows very promising results compared with the existing rule-based system. It gives a BLEU score of 89 in all the test cases and produces the top BLEU score of 93.7 for Sinhala to English transliteration.

Keywords-statistical machine translation; transliteration; naive bayes; sinhala; tamil

I. INTRODUCTION

Transliteration can be defined as the phonetic translation of names across languages [1]. However, transliteration may be based on pronunciation, spelling, or a combination of the two. It is complicated by the varied approaches taken by different people. Therefore, it is a challenging task to write names in another script. One problem is that the target script may not have a way to represent the sounds in the name. Therefore, in practice names are transliterated to produce the closest pronunciation. For example, the name අනගාරික දර්මපාල in Sinhala language can be transliterated to Tamil language as அனகாரிக்க தர்மபால and to English language as Anagarika Dharmapala which gives a close pronunciation.

Apart from pronunciation, writing names also may be influenced by other factors like numerology. For instance, people sometimes may include characters intentionally to comply with their numerology requirements. For instance, a person can write his name as Anagaarika instead of Anagarika. However, it is not very straight forward to generate the closest transliteration of names. Sinhala and Tamil are the official lan-

guages of Sri Lanka and English is the link language. All these are used widely in government documents and other varieties. The department for registration of people in Sri Lanka has a system to transliterate between Sinhala in Tamil which is used to transliterate names in national identity cards ¹. However, there are not many formal studies on transliteration among Sinhala, Tamil and English languages are found except the work by Hettige et al [2].

II. MOTIVATION

The need for a transliteration system emerges due to several reasons. As per the policy in Sri Lanka, it is a must to represent names in all three languages. Especially, the government documents are issued mostly in all three languages and those documents contain a lot of personal names. Currently, these names are transliterated manually in government documents. Different translators find different ways to transliterate as there is no one to one mapping between Sinhala – Tamil and English characters and sounds. As an example, the characters ග, ඩ, ත, එ in Sinhala are mapped to letter ஜ in Tamil.

Therefore phonemes which can be transcribed using Sinhala cannot be done exactly in Tamil or English and vice versa also true. One such case is that the phoneme ‘z’ is not in Sinhala or Tamil languages. As an example, how to write the name Fouza in Sinhala is somewhat confusing as some write it as ගුසා and some others as ஜூஸා.

In some cases, the influence of other languages including Sanskrit, Dutch, Portuguese, Arabic can be seen in proper names and especially in personal names. As an example, names like Fernando and de Silva have a Portuguese origin. In Sinhala language, those become අන්තෝ and ගැ සිල්වා, which does not represent the exact phonetic mapping to English.

On the other hand, a transliteration system of this nature is also useful for Natural Language Processing (NLP) tasks such as machine translation between these languages, corpus or sentence alignment, cross-language information retrieval, information extraction and automatic lexicon acquisition.

¹<https://bit.ly/2Z8TEBV>

Therefore there is a need for transliteration system which can transliterate names between Sinhala – Tamil and English Languages.

III. RELATED WORK

There have been several research studies done on this area for different languages. Basically, transliteration is considered as a key component of translation. In this field, there are a number of models have been developed based on machine transliteration approaches such as phoneme based transliteration model, grapheme based transliteration model, hybrid transliteration model and also correspondence-based transliteration model [3].

Grapheme based transliteration model is a direct mapping of spellings or grapheme from a source language to a target language. Most of the time this is an orthogonal mapping. Channel Model and Decision Tree Model are such transliteration methods proposed using the above approach [4]. Phoneme-based transliteration model is basically built on pronunciation or the source phoneme of words in the language rather than the spellings. A Hybrid approach is one that uses both of the above for more accurate transliteration since some of these do not match with certain language specifications.

In 2007, Jiang et al [4] have proposed a method of improving the translation of named entities with the help of transliteration together with web mining. There in the transliteration model, they consider both the similarity in pronunciation and co-occurrence of bilingual contextual information of the words. A list of generated candidates is kept scoring with the help of web mining and improve the quality of the translation with better transliteration.

Later in 2014, Mathur et al [5] have reported a transliteration method for named entities in Hindi language following a Hybrid approach. There, they have used a rule-based approach for the extraction of named entities and a statistical approach in converting named entities in English to the corresponding Hindi representations. This is why it is called a hybrid approach. Further, they have applied this mechanism to their system and calculated the accuracy using precision, standard recall, and f-score. The results obtained from these experiments were compared with the results of manually transliterated named entities that was done with the help of human translators. Those results have shown better progress than the previous occasions. In 2016, Y. Lin et al [6] have reported how the transliteration methods ignore source context information and inter-dependency among entities for entity disambiguation. They bring out a novel approach to leverage state-of-the-art Entity Linking (EL) techniques to automatically correct name transliteration results, using collective inference from source contexts and additional evidence from the knowledge base.

Grundkiewicz and Heafield [7] proposed a neural machine translation based approach for transliteration in 2018 using deep attentional RNN encoder-decoder models. Mihaela and Breuel presents another approach for neural network based model for transliteration using a sequence to sequence model [8]. Their data is based on Arabic and English parallel text. Kundu et al [9] proposed an approach for transliteration based on both recurrent neural networks and convolutional sequence to sequence based neural machine translation.

When it comes to Sinhala language machine transliteration have been done for over a decade in terms of translation. In 2007, Hettige and Karunananda [2] propose a system for transliteration from English to Sinhala language. There the approach they used is based on finite-state automata. They have used a finite state automaton to develop transducers for language transliteration and the system is developed using Prolog server page and SWI-PROLOG. Those generated transducers are tested using Sinhala Chatbot and English WordNet to obtain the expected results. They claim that handling the pronunciations of an English word is critical as one letter could have different sounds and it causes to leave some ambiguity in transliteration. The team intended to improve this system by incorporating IPA to their system as future work.

In 2010, B. Hettige et al [10] reported a methodology for English to Sinhala machine translation supported by a transliteration agent using finite-state transducers.

In 2018, Tennage et al [11] have built a system for neural machine translation supported with transliteration from Tamil to Sinhala language. This transliteration model was implemented using English as a middle language. The model has given a BLEU score of 8.36 for Tamil to Sinhala transliteration, which was a rule based approach that used character mapping without considering named entities.

In 2018, Thayaparan [12] has proposed a named entity translation model which encompassed word-embedding models to improve translation in between Sinhala and Tamil. The model was able to gain BLEU score of 35.28 for Sinhala to Tamil, and 23.75 for Tamil to Sinhala, after integrating into existing translation system.

In 2003, Kohen et al [13] present about phrase-based machine translation and in 2007 Kohen and his team introduce Moses toolkit [14] for statistical translation. In 2009, Chinnaikotla and Daman [15] talks about using Moses toolkit for transliteration, following a phrase-based SMT approach. There the words are replaced by characters and sentences are replaced by words. Again Rejwanul et al [16] follow the same approach for English-Hindi transliteration.

IV. TRANSLITERATION SYSTEM

A. Our Approach

There are several transliteration systems that were built following a rule-based approach [5, 17, 18]. However, there are many cases where the rules cannot correctly handle mappings in between languages [2]. Considering these difficulties, we moved with a statistical machine translation approach. We also realised that the way names were transliterated depended on the ethnicity of the name. In Sri Lanka, the main ethnicities are Sinhalese, Tamils and Muslims. Therefore, before the transliteration of personal names we classified names based on ethnicity. This was done with a simple classification followed by the Naive Bayes algorithm. These classified names were then used to construct separate models based on both ethnicity and language. Therefore, altogether we developed 10 models for the transliteration of Sinhalese, Tamil and Muslim names in the context of Sinhala, Tamil and English languages.

In transliteration, there are several techniques which have been used in previous research studies. Direct mapping approach is one of them [19], which generates the results using parallel corpus which is given for training. It consumes minimum time than other approaches [19], however, it can transliterate only words which are present in the parallel corpus.

There are several other transliteration systems that were built basically following a rule-based approach. In this approach, different rules will be used to generate transliteration results. Rules can be created by considering the key properties of source and target languages. The rule-based approach is not widely used since it takes time, money and trained personnel to make and test the rules.

Other than the above two approaches of direct mapping and Rule-based, neural machine translation (NMT) is also a current technique which learns directly and treats the words as smallest units for transliteration. Its slower training and inference speed, ineffectiveness in dealing with rare words, and sometimes fail to translate all the words in the source sentence. NMT lacks robustness in translating rare words [20] and it needs large parallel data sets to train the model to obtain better results than SMT [11].

Considering those difficulties, we used a statistical machine translation (SMT) approach. SMT is a language-independent and data-oriented approach to transliterate text from a source language to a target language. SMT has high accuracy results and time efficient than rule-based approach [19]. SMT transliterates not only Sinhala origin names but other names as well. When corpus size is small, SMT performs better than the NMT according to results obtained by Tennage et al. [21]

Our transliteration system development involves the following steps:

- Preprocessing the data in Sinhala - Tamil parallel name corpus:
 - * cleaning the data to remove duplicates, correct spelling, correct Unicode errors and break names to labels
- Training and tuning models for Sinhala to Tamil and Tamil to Sinhala transliteration using Moses decoder ². Then testing the models, measuring the BLEU score and identifying the issues with transliteration model.
- Manually classifying a part of Sinhala Tamil parallel name corpus based on the ethnicity of the names.
- Creating a model for ethnicity-based classification of names using the manually classified data using the Naive Bayes algorithm.
- Classifying the rest of the data in the parallel corpus with the built classification models, and manually validating the accuracy.
- Retraining models for Sinhala to Tamil and Tamil to Sinhala transliteration based on ethnicity and building six models representing each transliteration with Moses decoder. Then tuning and testing each model for the BLEU score.
- Conducting a survey to collect proper names from all three languages.
- Scraping web and collecting a list of English proper names. Then transliterating them to Sinhala language using a rule-based approach and manually correcting the result and building an English Sinhala parallel name corpus.
- Building models for Sinhala to English, and English to Sinhala Transliteration, with the data from English Sinhala parallel corpus, and testing for the accuracy of the model.
- Using the developed Sinhala to English Transliteration model, transliterating a set of Sinhala names from the Sinhala English name corpus, to English language. Then manually validating the names and creating an English Tamil parallel corpus.
- Building the models for English to Tamil and Tamil to English transliteration with Moses decoder, using the English Tamil parallel data. Then tuning and testing the models for the accuracy.

B. Dataset

A parallel name corpus of 100,000 Sinhala and Tamil was obtained from a government department. However, some of these names, especially Sinhala names had multiple tokens in their names. For instance, the name Chathuri Ishaka Harshani has three tokens all belonging to one personal name and also mapped to corresponding Tamil labels in the corpus.

However, no publically available Sinhala-English or Tamil-English transliterator was found. Therefore, we tried a survey approach to collect names from university students and the general public through a survey

²<https://github.com/moses-smt/mosesdecoder>

Table I
EXAMPLES FOR CONFUSING RESULTS FROM ENGLISH TO SINHALA
RULE-BASED TRANSLITERATION

Name in English	Transliterated result in Sinhala	The expected result in Sinhala
Sandakelum	සන්දකේලුම (sandakelum)	සඳකැලුම (sandakalum)
Imasha	ශ්‍රීමා (imāʃā)	ශ්‍රීමා/ශ්‍රීමා (imāʃā:/ imāʃā)
Menike	මෙනික (Menika)	මැනික (Mənike:)
Yogaraj	යොගරජ (jogaradž)	යොගරජ (jo:gara:dž)
Margret	මර්ග්‍රේට (mərgret)	මාග්‍රට (ma:grat)

and we ended up collecting 2000 names which were insufficient for us to train the system. Then we crawled the web to find proper names in English, mostly from the websites with examination results published. Finally, we collected around 80000 proper names in English. These names were then transliterated to Sinhala language with a rule-based approach followed by H.M. Weerasingha [17] which had a BLEU score of 80.03%. However, there were a lot of issues with the transliterated results, some of which are shown in Table 1.

It is not that these transliterations are incorrect, but when it comes to proper names in Sri Lankan context the name in the last column is preferred or widely used than the transliterated output in the second one as in Table 1. Therefore, all these transliterations were checked for errors and corrected manually. But still, there are some ambiguities in names, especially gender-wise. As an example, the name Maneesha can be transliterated either as මැනීෂා or මැනීෂ, a depending on whether the name is female or male.

C. Model Creation

In each model creation process, between pairs of two languages from Sinhala, Tamil and English, the Moses-decoder was fed with the parallel corpus of personal names in source and target languages. The corpus contained names in random order and it was also partitioned to a ratio of 5:2:1 for training, tuning and testing respectively. We converted words into a sequence of characters, i.e., we used character segmentation rather than word segmentation.

Before training the transliteration model, a language model was built with the target language using KenLM. It was to find the most widely used or preferred from the number of outputs generated by the SMT. In this case, a three-gram language model is generated and also binarised with KenLM³ to achieve faster loading. We used Giza++⁴, which is the default tool in Moses, to build the translation models, in our case the transliteration models. Once the extraction

³<https://khefield.com/code/kenlm/>

⁴<https://github.com/moses-smt/giza-pp>

Table II
EXAMPLES FOR CONFUSING RESULTS FROM SINHALA TO TAMIL
TRANSLITERATION

Input name in Sinhala	Transliterated result in Tamil	The expected result in Tamil
රමණායකලාගේ (ra:māna:yakala:ge)	ராமநாயக்கலாகே (ra:māna:jakala:ke:)	ராமனாயகலாகே (ra:māna:jakala:ke:)
වේශ්‍රාමී (ve:sra:mi)	வெலுச்சாமி (ve:lusa:mi)	வெலுச்சாமி (ve:lusa:mi)
සෙලුවරා :අ~ඩා :	செல்வராஜா (sellvara:da)	செல்வராசா (sellvara:sa)
ඇවුසේෂ (alvusef)	அவුசේප் (alvusep)	ஆவුசේප் (o:lvusep)
මැරික්කාර (mærikkar)	மெரிக்கார் (merikka:)	மரிக்கார் (marikka:)

Table III
EXAMPLES FOR CONFUSING RESULTS FROM TAMIL TO SINHALA
TRANSLITERATION

Input name in Tamil	Transliterated result in Sinhala	The expected result in Sinhala
அபිලாஷா	අපිලාඡා (apila:ʃa:)	අභිලාඡා (abilā:ʃa:)
கங்கா	காகா (kaŋkā)	ஙாஙா (gāŋga:)
சித்திரவேல்	சிட்டிரவேல் (sittiravēl)	சிட்டவேல் (sidravēl)
ஜெப்றின்	ஜெப்றின் (dʒebrin)	ஜெற்ன (dʒerfin)
வகாப்தீன்	வகாப்தீன் (vakā:pṭī:n)	வகாப்தீன் (vakā:pdi:n)

of terms, scoring and lexicalized reordering tables creations are done, the final Moses configuration file is taken as the output of the training phase. Then each model is tuned for better results and tested for the BLEU (Bilingual Evaluation Understudy) score.

D. Sinhala-Tamil Transliteration Model

The first model we created was to transliterate between Sinhala and Tamil languages. As stated above it was done with a parallel corpus of 100,000 entries of full names. Though the model gave good results, still there were some confusing cases and some such examples are displayed in Table 2 and Table 3.

In these cases, also, it is not that the character mapping in transliteration is incorrect, but the possibility of having such names is rare. However, when analysing further we found that this irregularities arose may be due to the ethnicity of those names belongs to. In Sri Lanka, the way names are written in a language depends on their ethnicity. At the same time, the mapping in between the characters in Sinhala, Tamil and English names are not one to one. Therefore, we had to address this issue of diversity by classifying names based on these irregularities in ethnicity.

E. Classification of personal names based on ethnicity

In most cases of Sri Lankan context, a name could reveal the person's ethnic group and the names have their own specifications based on that origin. In this case, we observed that there is a significant pattern

of transliterating a name based on ethnicity. Therefore, the names were first classified according to their ethnicity (Sinhalese, Tamil or Muslim) before transliteration.

Then we built a machine learning model to classify the personal names into Sinhalese, Muslim and Tamil names. In making the training data set for the classification model, we manually classified around 30,000 full names into each category. Next, we split the dataset into training and validation sets so that we can train and test the classifier. Also, we encoded our target column so that it can be used in machine learning models.

As features, raw text data was transformed into feature vectors and new features were created using the existing dataset. We implemented TF⁵-IDF⁶ (Term Frequency - Inverse Document Frequency) vectors as features in order to get relevant features. TF-IDF score represents the importance of terms appears in the entire corpus. We considered two feature vectors as N-gram level TF-IDF and character level TF-IDF. N-gram level TF-IDF vector represents TF-IDF scores of N terms together and character level TF-IDF represents the scores of character level N-grams in the corpus. After analyzing the accuracy of models using both feature vectors, Character level N-gram TF-IDF feature give better results than the other features.

Finally, we implemented a Naive Bayes model using Sklearn⁷ implementation with different features. Naive Bayes is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. Then, we trained the classifier using the training data set and validated using the rest of the data. This model gave a 96.1% accuracy on ethnicity-based classification. Sinhala Tamil Transliteration Models based on ethnicity Using the classified names from the classification model, we trained three separate models based on ethnicity as Sinhalese, Tamil and Muslim, by feeding the Moses decoder with relevant data for each language pair for each ethnic categories using the same way as we followed earlier. The transliterations were done both back and forth between Sinhala and Tamil languages. This approach gave better results than the previous time and most of the confusions in transliteration mapping were resolved.

F. Sinhala English Transliteration Models

As described previously, using the trilingual corpus we created, two other models were built for back and forth transliteration in between Sinhala and English languages. In this case, the models gave a good accuracy even without classification on ethnicity. Therefore, for this direction, we considered names belonging

⁵TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document)

⁶IDF(t) = log_e(Total number of documents / Number of documents with term t in it)

⁷<https://github.com/scikit-learn/scikit-learn>

to all ethnic groups together as it is an overhead to create more models.

G. Tamil English Transliteration Models

The rule-based approach was giving poor results in the transliteration of names from English to Tamil. Therefore, to generate an English Tamil parallel corpus with personal names, we used the models trained by Moses for English to Sinhala transliteration and Sinhala to Tamil transliteration. They were giving better results than the Rule-based approach and still we had to correct them manually to create a Tamil and English parallel corpus with no spelling mistakes.

V. EVALUATION AND DISCUSSION

Sinhala to Tamil and Tamil to Sinhala transliterations models without ethnicity-based classification were based on a corpus of all types of personal names found in Sri Lanka. There the model for Sinhala to Tamil transliteration gave a BLEU score of 82.33% while the model for Tamil to Sinhala transliteration presented a BLEU score of 80.02%. There were several issues with these transliterated results caused by the variations of the originality of these proper names. As an example, mostly in the Tamil language, the letter ‘க’ represents the [h] phoneme in personal names. However, when it comes to Muslim names they are used to write with the ‘ஹ’ character, which is a Grantha consonant but not used widely in the names of Tamils. In some cases, some letters are missing in source or target language. As an example, there is no separate phoneme for ‘ஓ’ [f] in Tamil language. Then the name வாதிமா (Fathima) is written as பாத்திமா in Tamil where character ‘ப’ represents the phone for [p] as well. The name அருமீ is also written using the same character, as பாருமீ but representing the phoneme [p]. Again all the phonemes ‘க்’, ‘ஓ்’, ‘ஏ்’, ‘ஃ’ in Sinhala language is mapped to single ‘க’ in Tamil language in general use. In all these cases the choice was mainly based on the ethnic group. This is one example and there are many more such many to one or one to many mappings that confuses the system when all types of names are considered together. But, this confusion gets decreased, once when we consider the ethnicity behind the origin of these names. BLEU score quality metric increased with the application of the classification model. The system shows the BLEU score greater than 89% for all pairs of transliteration. For the transliteration from Sinhala to English, a BLEU score of 93.7% was obtained for the backward transliteration the score was 92.37%. All the BLEU Scores used in this paper are in BLEU-4 metric.

VI. CONCLUSION

In this paper, we have presented a statistical machine translation approach to transliterate personal names in Sri Lankan context using Moses SMT toolkit for Sinhala, Tamil and English languages. We have improved the results further by using the ethnic origin

Table IV
BLEU SCORES IN ETHNICITY-BASED MODELS

Ethnicity	Source Language	Target Language	BLEU Score (%)
Sinhalese	Sinhala	Tamil	89.15
Sinhalese	Tamil	Sinhala	91.47
Tamil	Sinhala	Tamil	93.62
Tamil	Tamil	Sinhala	91.29
Muslim	Sinhala	Tamil	89.35
Muslim	Tamil	Sinhala	89.61

Table V
BLEU SCORES IN TAMIL-ENGLISH AND SINHALA-ENGLISH MODELS

Source Language	Target Language	BLEU Score (%)
Sinhala	English	93.70
English	Sinhala	92.37
Tamil	English	86.11
English	Tamil	91.74

of a given name, whether Sinhalese, Tamil or Muslim. We developed a classification model to classify names before feeding to Moses for transliteration. Our system shows a BLEU score of more than 89% for all the language pairs of consideration.

VII. FUTURE WORK

The personal name transliteration module is just a part of our named entity translation project. Therefore we will extend this to cover location names, organizational names and designations. In these cases, just transliteration would not be enough. Therefore, we will also use other techniques like terminology integration to improve quality.

REFERENCES

- [1] N. Chen, X. Duan, M. Zhang, R.E. Banchs , H. Li, "Whitepaper on NEWS 2018 Shared Task on Machine Transliteration"
- [2] B. Hettige and A. S. Karunananda, "Transliteration system for English to Sinhala machine translation," 2007 International Conference on Industrial and Information Systems, Peradeniya, 2007, pp. 209-214.
- [3] O. Jong-Hoon, C. Key-sun, I. Hitoshi, "A comparison of Different Machine Transliteration models", Journal of Artificial Intelligence Research, pp 119- 151, 2007.
- [4] L. Jiang, M. Zhou, L.F. Chien, C. Niu, "Named Entity Translation with Web Mining and Transliteration", The International Joint Conference on Artificial Intelligence, Inc. (pp. 1629-1634). Hyderabad: Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 2007
- [5] S. Mathur, V. P. Saxena, "Hybrid approach to English-Hindi name entity transliteration," 2014 IEEE Students' Conference on Electrical, Electronics and Computer Science, Bhopal, 2014, pp. 1-5
- [6] Y. Lin, X. Pan, A. Deri, H. Ji, K. Knight, "Leveraging Entity Linking and Related Language Projection to Improve Name Transliteration", 2016
- [7] R. Grundkiewicz, K. Heafield "Neural Machine Translation Techniques for Named Entity Transliteration", Proceedings of the Seventh Named Entities Workshop, July 2018
- [8] M. Rosca, T. Breuel "Sequence-to-sequence neural network models for transliteration", 2016
- [9] S. Kundu, S. Paul and Santanu Pal, "A Deep Learning-Based Approach to Transliteration", 2018
- [10] B. Hettige and S. K. Asoka, "An evaluation methodology for English to Sinhala machine translation," 2010 Fifth International Conference on Information and Automation for Sustainability, Colombo, 2010, pp. 31-36.
- [11] Tennage, P., Herath, A., Thilakarathne, M., Sandaruwan, P. (2018). Transliteration and Byte Pair Encoding to Improve Tamil to Sinhala Neural Machine Translation. Moratuwa Engineering Research Conference (MERCon). Moratuwa, Sri Lanka: IEEE.
- [12] M. Thayaparan, "Translation of Named Entities Between Sinhala and Tamil for Official Government Documents", M.S. thesis, Dept. of Comp. Science and Eng, Univ. Moratuwa, Sri Lanka, 2018.
- [13] P. Koehn, F. J. Och, D. Marcu. 2003. "Statistical phrase-based translation", Proc. of HLTNAACL 2003, Edmonton, Canada, pp. 48-54
- [14] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin and E. Herbst. 2007. "Moses: open-source toolkit for statistical machine translation", Proc. of ACL, Prague, Czech Republic, pp. 177- 180.
- [15] M. K. Chinnakotla, O. P. Damani. 2009. "Experiences with English-Hindi, English-Tamil and English-Kannada transliteration tasks at NEWS 2009", In Proc. ACL/IJCNLP Named Entities Workshop Shared Task.
- [16] R. Haque, S. Dandapat, A. K. Srivastava, S. K. Naskar, A. Way, "English-Hindi transliteration using context informed PB-SMT", In Proc. ACL/IJCNLP Named Entities Workshop Shared Task, 2009.
- [17] H. M. Weerasinghe, "Transliteration of Names from English to Sinhala", M.S. thesis, Dept. of Comp. Science and Eng, Univ. Moratuwa, Sri Lanka, 2006.
- [18] S.C. Fernando, "Inexact matching of proper names in Sinhala", M.S. thesis, Dept. of Comp. Science and Eng, Univ. Moratuwa, Sri Lanka, 2007 .
- [19] V. Kaur, A. K. Sarao, J. Singh, "A Review on Hindi to English Transliteration System for Proper Nouns Using Hybrid Approach", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 3, Issue 5, September 2014.
- [20] Y. Wu, M. Schuster, Z. Chen, Q. V Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. "Google's neural machine translation system: Bridging the gap between human and machine translation". arXiv preprint arXiv:1609.08144, 2016.
- [21] P. Tennage, P. Sandaruwan, M. Thilakarathne, A. Herath, S. Ranathunga, "Neural Machine Translation for Sinhala and Tamil Languages", in International Conference on Asian Language Processing, 2017.

Automatic Recognition of Allusions in Tang Poetry Based on BERT

Xuemei Tang^{1,2}, Shichen Liang^{1,2}, Jianyu Zheng³, Renfen Hu^{1,2}, Zhiying Liu^{1,2}

¹Institute of Chinese Information Processing, Beijing Normal University, Beijing, China

²UltraPower-BNU Joint Laboratory for Artificial Intelligence, Beijing Normal University, Beijing, China

{tangxuemei, shhichen, irishu}@mail.bnu.edu.cn,liuzhy@bnu.edu.cn

³Advanced Innovation Center for Future Education, Beijing Normal University, Beijing, China

zheng_jianyu@126.com

Abstract—In this paper, we propose an automated method for recognize allusions in Tang poetry. The representation of text is trained by BERT pre-trained by *The SiKuQuanShu*. The TOP-20 candidate allusions have the highest semantic similarity to the original sentence. Then update the ranking of candidate allusions by rule-based ranking algorithm. In the final experimental results, the precision of the correct allusion same as the final ranking TOP-1 reached 63.74%, the precision of the correct allusion appears in the final ranking TOP-3 reached 70.66%, and the precision of the correct allusion appears in the final ranking TOP-5 reached 74.82%.

Keywords-Tang poetry; allusions; automated recognize; BERT

I. INTRODUCTION

Tang poetry is another monument after the "Book of Songs" and "Chu Ci" in the history of Chinese poetry. Tang poetry often expresses rich ideological significance in a concise form. For example, Tang Dynasty often quoted allusions, and allusions often contain far-reaching meanings. Poets can quote stories or words in ancient books to express their thoughts. The Tang Dynasty poets preferred to quote allusions in many methods, they also good at adopting the methods of explicitly quote, implicitly quote, positively quote, oppositely quote, practicality, virtually quote, etc. The Tang Dynasty intellectuals had broad vision, rich knowledge structure and the allusions they quoted almost all over the classic literature before Tang Dynasty [1]. The method of poet quoted allusions change frequently and allusions is far from the source, which increases the language barriers for readers to read and appreciate. In order to solve this problem, in 1980s, some scholars have compiled the "Full Dictionary of Tang Poetry" to help Tang poet learners learn and understand Tang poetry [2].

In the environment of high-speed informationization, the way of knowledge propagation in the original book form can no longer meet people's needs. How to enable learners to acquire knowledge efficiently and conveniently is a problem that researchers are solving. With the rise of artificial intelligence, it has made breakthroughs on syntactic analysis and semantic analysis. This paper takes the automatic recognition the allusions in Tang poetry as a natural language processing task, combine semantic computing and ranking algorithm to automatically recognize the allusions in Tang poetry.

When the poet quotes the same allusion in poem sentence, the thoughts or feelings expressed by sentence may be very similar. For example, the allusion of "桃花源" comes from Tao Yuanming's "桃花源记 (Peach Blossom Spring)", which is mostly referred to as fairyland or seclusion in the Tang poetry. Based on this idea, we assume that if two poems quoted a same allusion then two poems are semantically similar. Therefore, we select some candidate allusions by calculating the semantic similarity, and then use the ranking algorithm to rank the candidate allusions to get the final recognition result. According to the sixth edition of the Modern Chinese Dictionary, the concept of the allusion: "The story or expression in the ancient book quoted in the poetry" [3]. The allusions to be recognized in this paper refer to the stories or words in the ancient books quoted in the poems of Tang poetry.

In this paper, we propose a method to automatically recognize the allusions in Tang poetry, and obtain better experimental results. At the same time, we propose a rule-based ranking algorithm suitable for Tang poetry allusion recognition. We hope that our work will provide assistance to Tang poetry learners and provide some reference for Tang poetry researchers.

II. RELATED WORK

The automatic recognition of allusions in Tang poetry is a special task, and some researchers have achieved some achievement in related tasks. Yao Rui [4] in 2011 segmented the allusions and poems, and then used the word matching method to identify the allusions in the poems. The dictionary method and mutual information method are combined in the word segmentation method. If the allusion completely appears in the poem, the method can effectively identify the allusion of the poem, but when the allusion quoted indirectly, the method has a poor recognition effect. Moreover, the method is based entirely on keyword matching and does not incorporate any natural language processing techniques. Dan Roth [5] 2014 used natural language processing techniques to identify entities in Wikipedia documents and link to Wikipedia pages of entities that match the current context. The method of recognize entities, utilize n-gram, phrase extraction and other methods, while adding external entity resources. After the entity is recognized, an entity often corresponds to multiple Wikipedia pages. By computing the semantic similarity between Wikipedia pages of the entities and the context in which the entity currently located, choose the entity page that best matches the current context.

The method proposed by Dan Roth can effectively recognize the entities and give the explain pages of the entities. Inspired by this method, this paper uses this method to recognize the allusions quoted in Tang poems. Because of the wide variety of allusions, named entity recognition is not suitable. Therefore, we first calculate the semantic similarity between the example sentences (below called original sentence) and the poems in the candidate set. The poems in the candidate set are known which allusion quoted. Then the TOP-K candidate allusions with the highest semantic similarity are ranked. Through ranking, the highest order is the allusion quoted by the original sentence.

III. DATA AND METHOD

A. Data

The data used in this paper comes from the "Complete Dictionary of the Complete Tang Poetry" written by Fan Zhilin [2], which includes the dictionary of the allusions, interpretation, and some example Tang poems quoted the allusions. The dictionary includes two volumes, which basically include all the allusions quoted in the Tang Dynasty poems. The example contents of the dictionary are shown in Figure 1.

After obtaining the data of the entire dictionaries from the CNKI China Tool Book Network Publishing Library, the data was cleaned and simplified, obtain 19,630 allusions examples. In the dictionaries, there are a large number of synonymous allusions. For example, the interpretation of the "团扇" is "见班女扇条"(equal to "班女扇"). In the experiment, the "团扇" and the "班女扇" will be merged into synonymous allusions, retain one of them. Through the synonymous merger, obtain 4,961 allusions. We extracted 815 allusions with more than 5 example Tang poems' sentences from the 4961 allusions. Then, one sentence is randomly selected as the test data, other four sentences are selected as the candidate example sentences, and a total of 815 example sentences in the test set. The candidate set has a total of 3260 example sentences, and each example sentence corresponds to the allusion quoted in the sentence. At the same time, we retain the interpretation of 815 allusions. Table 1 shows a part of test data.

Table 1. A part of test data

NO.	Allusions	Candidate
1	非熊(FeiXiong)	非熊德愧当周辅, 称杰叨惭首汉臣
2	许由(XuYou)	不学尧年隐, 空令傲许由
3	嵇阮(JiRuan)	嵇阮没来无酒客, 应刘死后少诗人
4	五柳(WuLiu)	旧业传一经, 新官裁五柳
5	冯唐(FengTang)	此地含香从白首, 冯唐何事怨明时

B. Method

The experiment mainly includes three parts: text vectorization, semantic similarity calculation and ranking. The experimental methods used by these three modules are described below.

Text vectorization is a very important step in natural language processing tasks. In this paper, we use Google's 2018 release Bidirectional Encoder Representation from Transformers (BERT) to represent text. This model was pre-trained by two tasks: 1) predict central word by

context words; 2) predict next sentence for current sentence.

BERT training process: First, some words of the corpus is masked, so that the model predicts the masked words according to the context, and the general model is initially trained. Then, pick out successive context sentences from the corpus and let the transformer model identify whether the sentences are the next sentences. These two steps together to complete the pre-training, become a language representation model that can achieve omnidirectional prediction of the context.

The BERT training process is unsupervised, don't require manual intervention and labeling, making it possible to train with very large-scale corpora at low cost. At the same time, the model combines the context of all layers of the neural network for training, so that the trained model can understand the semantics in combination with context and achieve more accurate text prediction generation when dealing with question-answer or language reasoning tasks. The model achieves optimal results in multiple natural language processing tasks through pre-training and fine tuning. The structure of the BERT model is shown in Figure 2. The training parameters are set as Table 2.

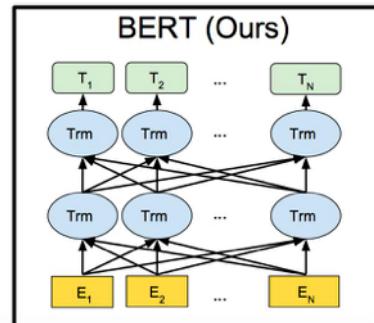


Figure 1. BERT model structure [6]

Layers	Hidden	Self-	All
12	768	12	110M

Table 2. BERTBASE parameters set

Because the data of our paper is poetry, most are single-character words, it may be not suitable to directly use the BERT pre-trained Chinese character model published by Google. For our task, we use *The SiKuQuanShu* to pre-train the BERT model to get the classical Chinese BERT language model.

The BERTBASE model structure has 12 layers, and each layer can output a sentence representation vector. Inorder to choose which layer of output is more suitable as a representation for allusion recognition task, we output the poem sentence representation from the BERTBASE model -1, -2, and -3 layers, respectively, to compare the effect of the semantic representation of the poem sentences by three layers. We use three layers' representation to calculate three Tang poem sentences with the highest semantic similarity to the poem sentence "杜陵犹识汉, 桃源不避秦" (original sentence). The semantic similarity calculation function shown as equation (1). As shown in Table 3, through calculating withthe -1 layer representation. The sentence with highest semantic similarity to original sentence is "不知今有汉, 唯言昔避秦", not only semantic similar, but also quoted same

【出典】《東觀漢記》卷二十三《載記 魄靄》：“靄將王元說靄曰：‘……元請以一丸泥為大王東封函谷關，此萬世一時也。’”

【釋義】東漢時魄靄部將王元用一丸泥比喻函谷關地勢險要，極易扼守。后世用作扼守險關的典故。

【例句】①不將千里隔，何用一丸泥。(張說《奉和圣制潼關口號應制》977)這里是以本典表示江山一統，潼關的險要已無軍事上的意義。②系越有長纓，封關只一丸。(獨孤及《賈員外處見中書賈舍人巴陵詩集覽之懷舊代書寄贈》2762)這里用一丸稱頌賈舍人有據關能守之才。③平戎七尺劍，封檢一丸泥。(李益《再赴渭北使府留別》3220)這里用一丸泥喻指邊府有險塞可守。④空留三尺劍，不用一丸泥。(李賀《奉和二兄罷使遣馬歸延州》4417)這里以不用一丸泥喻指有一夫當關之才的二兄被罷還。(共 344 字)

Figure 1. Example contents of the dictionary

allusion "避秦". The results output by -2 and -3 layers are not ideal. Since we choose the sentence representation vector output from the last layer of the model as the semantic representation for the poems and allusions' interpretation.

After vectorization of all poem sentences and allusions, then we calculate semantic similarity between example sentence and candidates. In the experiment, we use the cosine similarity as the semantic similarity calculation method, shown as equation (1).

$$s^n = \frac{A \cdot B^n}{\|A\| \|B^n\|} = \frac{\sum_{i=1}^d A_i \times B_i}{\sqrt{\sum_{i=1}^d A_i^2} \sqrt{\sum_{i=1}^d B_i^2}} \quad (1)$$

A is the representation of original sentence; B is the representation of the candidate sentence. The dimension of vector is 768, $i \in [1, 768]$. $S^4 = \{s^1, s^2, \dots, s^n\}$, $n \in [1, 3260]$. If the value of s^n close to 1, illustrate that NO. n candidate is more similar to sentence A .

By calculating the semantic similarity between the original poem sentence and the candidate sentences, get the K-sentence candidates with the highest semantic similarity to the original sentence, there is an initial ranking among K-sentence, and then the candidate example sentences and their corresponding allusions are ranked by the ranking algorithm. Finally, the highest ranking is the recognize result.

If the same N-gram phrase appears in two poems, then the two poems are more semantically related, and the content to be expressed may have certain overlap, and they are more likely to quote the same allusion; the Tang Dynasty poets often directly quote allusion, there may be some N-gram overlap between the poem sentence and the allusion; at the same time, the interpretation of the allusion is the meaning expressed by the allusion, then the poem sentences quoted allusion and the interpretation of allusion may be semantic similar. Therefore, in the ranking process, we use the number of N-gram overlap to update the initial semantic similarity $s^{1..n}$ and update the ranking of the candidate allusions. In this paper, because the poem sentence is short, bi-gram and unigram are used when counting the number overlap grams. The whole ranking process is as follow:

RANK0: According to the semantic similarity, there are TOP-K candidate sentences with the highest semantic similarity, we get an initial ranking for TOP-

K sentences. If the candidate sentence of the similarity TOP-1 has a bi-gram overlap with the original sentence, we maintain the initial ranking. If not, entry RANK1.

RANK1: Count the number of overlap bi-gram between TOP-K candidate sentences and the original sentence respectively, N . $s_1^{1..n}$ updated by N to $s_1^{1..n}$ according equation (2). To ensure that the semantic similarity is not cleared to 0, the initial value of N is set to 1. The candidate sentences are ranked by the new $s_1^{1..n}$.

$$s_1^{1..n} = N \times s_1^{1..n} \quad (N \geq 1, n \in [1, 3260]) \quad (2)$$

After RANK1, if the new ranking is different from the initial ranking, the ranking of RANK1 would be finally result; if the ranking is the same as the initial ranking, imply the RANK1 is invalid, and entry RANK2.

RANK2: We choose the allusions corresponding to the TOP-K candidate sentences as candidate allusions, calculate the unigram overlap degree between the candidate allusions and the original sentence, and obtain the unigram overlap number M . Use M to update the semantic similarity $s_2^{1..n}$, as calculated by equation (3).

$$s_2^{1..n} = N \times s_2^{1..n} \quad (s_1^{1..n} = N \times s_1^{1..n}) \quad (3)$$

$s_2^{1..n}$ is different from $s_1^{1..n}$ and $s_1^{1..n}$, and $s_2^{1..n}$ is the semantic similarity between the interpretation of the candidate allusion and the original sentence, and the calculation method is consistent with $s_1^{1..n}$. The candidate allusions are ranked by a new semantic similarity $s_3^{1..n}$.

The final highest-order allusion is the final result of the recognition of the allusion of original sentence.

IV. EXPERIMENT RESULT AND DISCUSS

A. Experiment result

Based on the sentence embedding trained by pre-training BERTBASE, the semantic similarity of the 815 original sentence in the test set and the 3260 candidate sentences in the candidate set are calculated respectively, and the TOP-20 sentences with the highest similarity are chosen as the candidate poem sentences,

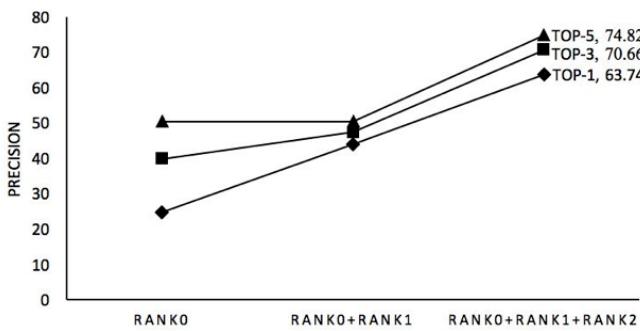


Figure 2. After different ranking method, the changes of precision

Table 3. The results of different ranking method (%)

	TOP-1	TOP-3	TOP-5
RANK0	24.71	39.72	50.34
RANK0+RANK1	43.87	47.34	50.34
RANK0+RANK1+RANK2	63.74	70.66	74.82

and then the allusions corresponding to TOP-20 sentences are candidate allusions, then rank candidates by rule-based ranking method. In the experiment, we choose precision metrics to measure the experimental result, and the precision function as equation (4). c is the number of correct recognition; a is the size of test data.

$$p = \frac{c}{a} \quad (4)$$

B. Maintaining the Integrity of the Specifications

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

The precision of different ranking methods is shown in Table 3. TOP-1 indicates that the precision of the correct allusion appears in the first place of the final ranking; TOP-3 indicates that the precision of the correct answer appears in the TOP-3 of the final ranking result; TOP-5 indicates that the precision of the correct answer appears in the top five of the final ranking result.

It can be seen from the change of the accuracy rate in Figure 3 that the precision from RANK0 to RANK2 shows an upward trend; as the ranking range of the calculation precision expanding (TOP-1 to TOP-5), the precision also increases. From the experiment results of RANK0, the precision from TOP-1 to TOP-5 increased by 25.63%, which proves our assumption that the poems quoted the same allusion have similar characteristics in semantics. More than 50% of the poems quoted same allusion as the original sentence are gathered into the TOP-5. The precision increase from 24.71% to 43.87% by RANK1 combined with the N-gram overlap number, indicating that RANK1 is a valid ranking method, and the similarity is updated by the number of bi-gram overlap. The method improves

correct candidate to the more advanced position from the previous order; the combination of RANK1 to RANK2, the precision increases by 19.87%, indicating that the way is effective to update the similarity by the candidate allusion and the original sentence unigram overlaps.

After combining the three RANK, the precision of TOP-1 allusion recognition reached 63.74%; the recognition precision of TOP-3 reached 70.66%; the recognition precision of TOP-5 reached 74.82%. experiment result imply that we proposed semantic similarity and rule-based ranking method can effectively recognize the allusions in some Tang poems.

B. Discuss

The ranking algorithm proposed in this paper has achieved good results in the experiment. The following will specifically analyze the role of the two ranking algorithms in the allusion recognition experiment. The original sentence is "天生逸世姿, 竹马不曾骑", the allusion quoted in this sentence is "竹马 (Bamboo horse)", and the candidate sentence of the similarity TOP-20 obtained by calculating the semantic similarity with the 3260 candidate sentences. As shown in Table 4, the poem sentence that quoted the same allusion as the original sentence, "爱向竹栏骑竹马, 懒于金地聚金沙" is ranked 14th. The original sentence and this sentence has bi-gram overlap, so when entry RANK1, the similarity of the sentence will become higher and the ranking will improve. As shown in Table 5, the sentence has a bi-gram overlap with the original sentence. At this time, N is 2, s_1 becomes twice as big as s , and the ranking rises to the position of TOP-1. The ranking result and initial ranking is different, since don't need to carry out RANK2, then "竹马 (Bamboo horse)" corresponding to is the TOP-1 sentence, "竹马 (Bamboo horse)" is the result of the recognition, same as the allusion quoted in the original sentence, the recognition is correct. The method of updating the ranking by using the bigram overlap number is more effective for the poems directly quoted allusions, and the effect for the poems indirectly quoted allusions is poor.

When RANK1 is invalid, the candidate allusions entry RANK2. The original sentence "治长空得罪, 夷甫岂言钱" quoted the allusion "治长非罪", in this poem sentence did not directly quote the allusion completely, resulting in RANK1 invalid. Then carried out RANK2 and we obtain the corresponding allusion through the TOP-20 candidate example sentences, that is, the TOP-20 candidate allusions. Then we calculate semantic similarity s_2 between the interpretation of the candidate allusion and the original sentence, get a new ranking for Top-20 candidate allusions. M is the number of unigram overlap of the candidate allusions and the original sentence, the semantic similarity is updated by M to s_3 . According to s_3 , update the new ranking. As shown in Table 7, the candidate allusion "治长非罪" is ranked TOP-1, and the allusion "一钱(one penny)"

Table 4. Candidate sentences initial ranking for original sentence “天生逸世姿，竹马不曾骑”

Ranking	Candidate allusions	Candidate sentences	S
TOP-1	仁风(mercy wind)	衣同莱子曾分笔，扇似袁宏别有天	0.794
TOP-2	将军树(General tree)	虽投定远笔，未坐将军树	0.790
TOP-4 --Top-13
TOP-14	竹马(bamboo horse)	爱向竹栏骑竹马，懒于金地聚金沙	0.774

Table 5. After RANK1, the ranking result of “天生逸世姿，竹马不曾骑”

Ranking	Candidate allusions	Candidate sentences	S ₁
TOP-1	竹马(bamboo horse)	爱向竹栏骑竹马，懒于金地聚金沙	1.549
TOP-2	仁风(mercy wind)	衣同莱子曾分笔，扇似袁宏别有天	0.794
TOP-3	将军树(General tree)	虽投定远笔，未坐将军树	0.790
TOP-4	酷似仰牢之	酷似牢之玉不如，落星山下白云居	0.789

Table 6. After RANK1, the ranking result of “治长空得罪，夷甫岂言钱”

Ranking	Candidate allusions	Candidate sentences	S ₁
TOP-1	献赋(offer poem)	甘泉未献扬雄赋，吏道何劳贾谊才	0.849
TOP-2	原宪贫	贾生独未达，原宪竟忘贫	0.832
TOP-7	治长非罪	公治本非罪，潘郎一为民	0.818
TOP-11	一钱(one penny)	宋均颜未老，刘宠骨应贫	0.815

Table 7. After RANK2, the ranking result of “治长空得罪，夷甫岂言钱”

Ranking	Candidate allusions	S ₃
TOP-1	治长非罪()	2.101
TOP-2	一钱(one penny)	1.085
TOP-3	曹刘(CaoLiu)	0.565
TOP-4	张博望(Zhang Bowang)	0.539

coincides with the original sentence, so the ranking is also improved. RANK2 has a better recognition effect on poem sentences partially quoted allusions.

The allusion recognition bad case mainly includes two cases. 1) We need to calculate the semantic similarity between many original sentences and candidate example sentences. We only take out the TOP-20 candidate allusions, it is found that the correct allusion does not appear in TOP-20, resulting in recognition errors. For example, "秋风能再热，团扇不辞劳". Although the original sentence is a directly quoted the "团扇(Circular fan)" allusion, but because the TOP-20 semantically most similar candidate poem sentences does not appear the "团扇(Circular fan)" allusion, ranking is invalid, causing the final recognition result to be incorrect. In order to solve this problem, TOP-30 and TOP-50 candidate poem sentences can be selected for ranking, and try to ensure that correct allusions appear in the candidate sentences. 2) Because the number of bi-gram and unigram overlap between the original sentence and the candidate sentence and the candidate allusions need to be counted during ranking, when there

is no overlap, the ranking method fails. For example, the original sentence "古时填渤澥，今日凿崆峒" completely indirectly quoted the allusion "精卫(JingWei)", there is no bigram and unigram overlap between the candidate sentence and the allusion and the original sentence, and the ranking method fails, only depend on initial ranking. To solve this problem, the semantic similarity can be updated by the number of synonym pairs between candidate sentences and original sentence in the ranking process. Although the original sentence and the candidate sentences do not have bi-grams and unigrams overlap, but there may be multiple pairs of synonyms, then the two poems may also quote the same allusion.

V. CONCLUSION

In this paper, we use the semantic similarity and ranking algorithm to recognize the allusions quoted in Tang poetry, and obtain good experimental results. Firstly, BERTBASE pre-train by The SiKuQuanShu corpus, the pre-trained BERT model is used to get the

representation for poems and allusions. Then calculate the semantic similarity between the original sentence and the candidate sentences. Next, select the TOP-20 candidate poem sentences with the highest semantic similarity, get a initial ranking. Finally, according to the number of bi-gram and unigram overlap between the original sentence and the candidate sentences or candidate allusions to update the semantic similarity, update the initial ranking by the new semantic similarity. After two rounds of ranking, get the final ranking. In the final experiment results, the precision of the correct allusion same as the final ranking TOP-1 reached 63.74%, the precision of the correct allusion appears in the final ranking TOP-3 reached 70.66%, and the precision of the correct allusion appears in the final ranking TOP -5 reached 74.82%. In the future work, we will continue to improve the ranking algorithm, hoping to improve the precision of the allusion recognition.

ACKNOWLEDGMENT

This work is supported by the Fundamental Research Funds for the Central Universities, National Language Committee Research Program of China (No. ZDI135-42) and National Social Science Fund of China (No. 18CYY029).

REFERENCES

- [1] Yang Liu. Reading "The Complete Dictionary of the Complete Poems of Tang Dynasty" [J]. Literary Heritage. 1989-01: 123-126
- [2] Fan Zhilin, Wu Gengqi. "The Complete Dictionary of the Complete Poems of Tang Dynasty" [M]. Hubei Dictionary Press. 1989
- [3] Dictionary Editing Room, Institute of Linguistics, Chinese Academy of Social Sciences. Modern Chinese Dictionary [M]. The Commercial Press. 2012
- [4] Yao Rui. An automatic analysis system for poetry based on the ontology of allusions [J]. Software Guide. 2011, Volume 10, Issue 06: 80-82
- [5] Roth D, Ji H, Chang MW, et al. *Wikification and beyond: the challenge of entity and concept Grounding*[C]. Proceeding of the 52nd Annual Meeting of the Association for Computational Linguistics: Tutorial, 2014: page7
- [6] Devlin J, Chang MW, Lee K, et al. *BERT: pre-training of deep bidirectional transformers for language understanding*[C]. Proceeding of the 16th North American Chapter of the Association for Computational Linguistics, 2018
- [7] Xiong Qingyuan. *The Emotion of the Complete Dictionary of the Complete Poems of Tang Dynasty*[J]. Journal of Huanggang Teachers College. 1997, Volume 17, 02: 24-26
- [8] Yu Tingting. Research on the Allusions of the Analects of Confucius in Tang Dynasty[D]. Shaanxi Normal University, Master's Degree. 2017
- [9] Wang Guanghan. *Rethinking on the Origin of Allusions* [J]. Ancient Chinese Studies. 2004, 04: 90-93
- [10] Summer. *Research on Semantic Similarity Calculation of Chinese Words*[J]. Computer Engineering. 2007, 06: 1191-19
- [11] Liu Hongzhe. *Research on text semantic similarity calculation method* [D]. Beijing Jiaotong University, doctoral thesis, 2012
- [12] Xu Dezhi, Wang Huaimin. *Research on semantic similarity calculation method based on ontology*[J]. Computer Engineering and Applications. 2007-08: 154-156
- [13] Yoon Kim. 2014. Convolutional neural networks for sentence classification. In Proceedings of EMNLP 2014:1746–1751.
- [14] Jeffrey Pennington, Richard Socher, Christopher D. Manning. 2014. *Glove: Global vectors for word representation*. In Proceedings of EMNLP 2014:1532–1543.
- [15] Melanie J.Bell, Martin Schafer. 2013. Semantic Transparency Challenges for Distributional Semantics. Proceedings of the IWCS 2013 workshop: Towards a formal distributional semantics. Potsdam: Association for Computational Linguistics: 1-10.

Exploring Characteristics of Word Co-occurrence Network in Translated Chinese

Jianyu Zheng*, Kun Ma^{§†}, Xuemei Tang^{§†}, Shichen Liang^{§†}

*Advanced Innovation Center for Future Education, Beijing Normal University, Beijing China

zheng_jianyu@126.com

§Institute of Chinese Information Processing, Beijing Normal University, Beijing China

†UltraPower-BNU Joint Laboratory for Artificial Intelligence, Beijing China

{201821090021, tangxuemei, shichen}@bnu.edu.cn

Abstract—The translation activity involves both the source language and the target language. Compared to the standard texts in the two language, translated texts show unique language characteristics. In order to explore them from the perspective of integrality and complexity, we introduce complex network method into the study on translated Chinese. Firstly, selected the experimental texts from The ZJU Corpus of Translational Chinese(ZCTC) and its corresponding six sub-corpora, such as Press reportage and Popular lore. And then removed the punctuation and did word segmentation. Secondly, constructed a word co-occurrence network of translated Chinese. After analyzing and counting the parameters, such as their shortest path lengths, degree distributions and clustering coefficients in these networks, we verify that the word co-occurrence network of translated Chinese has small world effect and scale-free property. Finally, by constructing co-occurrence networks of standard Chinese and calculating their network parameters, we compare and verify the differences between translated Chinese and standard Chinese: “simplification” and the more usage of common words. Our work expands the application of complex network in translation studies, and provides a feasible approach for studying translated Chinese based on complex networks.

Keywords-translated Chinese; complex network; word co-occurrence; small world effect; scale-free property

I. INTRODUCTION

Translation gets involved in the source language and target language. So the translated text is closely related to these two languages. Since the 1980s, linguists started to pay increasing attention to the linguistic characteristics of translated texts. The researchers[1][2] in early stage described characteristics of translated language by vocabulary and syntax with the help of corpus. Based on that, they revealed the general characteristics of translation as follows:

- “simplification”: “translators unconsciously simplify language, information or the both at the same time.”[1]. In other words, translated texts are simpler than standard texts in original language, such as the smaller range in word usage and more common words;
- “explication”: the information implied in the source language or need to be deduced from the context will be expressed directly in the translated texts[3]. This characteristic is mainly reflected in

sentence extension, modifier increase and pronoun explicitation;

- “normalization”: typical and normative ways in the target language will be used to produce translated texts. It is mainly reflected by the reduction of special words and sentences, the stronger logic.

However, language is a special symbol system. The linguistic units in different levels do not exist independently. They depend on and restrain each other so that the language system has stringency and complexity. If using the traditional statistical approach, the association of different linguistic units will be cut apart to some extent. So it will be impossible to consider characteristics of translated language from the “systematic perspective”.

Facing to the complexity and systematization of language, it is more reasonable to describe it through a perspective of network. The network with self-organization, self-similarity, small world effect , scale-free property is called complex network, which studies the complex system from the global perspective. Regardless of how complex and large the network is, it always adopts two basic elements—nodes and edges for study. The complex network has already become a powerful tool in the study of the complex system. At present, the study and application of the complex network are not only involved into mathematics, computer science, physics, biology and engineering technology, but also widely used in society, politics, economics, management and language. Therefore, we attempt to use the complex network method to analyze translated Chinese text. The main contributions of this paper are stated as follows:

- to study translated Chinese from the perspective of the complex network and reveal small world effect and scale-free property;
- to compare the measured results of translated Chinese network with the standard Chinese’s with the similar stylistics, and reveal speciality of the translated Chinese network.

II. RELATED WORKS

The translated Chinese is a special kind of modern Chinese taking English as the source language and Chinese as the target language. Even if translated Chinese uses Chinese characters as the carrier, it obviously shows some characteristics different from the standard Chinese. Before the large-scale corpus appears, most of researches on translated Chinese belong to impressive type and comment

type, while the corpus method opens a new world for studying translated language. Yifan Zhu et al.[4], once used Chinese translated news corpus to generate subject-word lists and analyzed the quantified characteristics between translated Chinese and standard Chinese. Xiao and Dai(2014))[5] utilized ZCTC and LCMC corpus, and made a comparison on sentence length of the two from different texts. They found out that the largest difference in average sentence length between the two relied in academic texts. While in the novel texts, their gap was closest. By comparing ZCTC with LCMC corpus, Dai Guangrong(2013)[6] studied the matches of 10 high frequency nouns, and found that matching range of high frequency nouns in translated Chinese was wider than the standard Chinese's. The vocabulary change mode could be more diverse.

With a complicated network structure, language network system shows extreme complexity in terms of words, syntax, semantics. Liu Haitao[7] indicated that the complex network analysis method could reveal the overall characteristics of language as a relational system in the large-scale real corpus. Based on the English-speaking country corpus(BNC) with 10^7 word frequency, Cancho and Sole[8] constructed the co-occurrence network of English words and found out the small world effect and scale-free property. Liu Zhiyuan and Sun Maosong[9][10] constructed co-occurrence network of Chinese words and Chinese dependence syntax network to inspect characteristics from the perspective of the complex network. The experimental results indicated that these complex networks all still hold the similar characteristics like above.

III. THE NETWORK MODEL OF TRANSLATED CHINESE

In this paper, we take four steps to explore the characteristics of translated Chinese based on complex network.

- 1) Acquire translated Chinese corpus and preprocessing. According to actual needs, we acquire translated Chinese corpora. During preprocessing, it is necessary to remove the punctuation and segment words of the corpus.
- 2) Build word co-occurrence networks. As to word co-occurrence, it means two words within the distance n in a sentence. These two words have a co-word relationship, which is the basis of constructing word co-occurrence network.
- 3) Calculate network parameters & analyze characteristics. According to the indicators commonly used in the study of complex networks, we calculate the corresponding parameters of the word co-occurrence network. Then we analyze the characteristics of the network according to those results.
- 4) Compare with other networks. By Comparing with the language networks with the similar stylistics, we study the characteristics of translated Chinese further.

A. Construct word co-occurrence network of translated Chinese

The word co-occurrence network is a kind of network which is used to describe language through co-occurrence relation between words. A word co-occurrence network

can be abstracted into an undirected graph G , where nodes delegate the corresponding words, and edges delegate the co-occurrence relationship between words. At present, the mainstream methods for constructing co-occurrence networks are n-order Markov co-occurrence model and similarity-based co-occurrence model. In this paper, we adopted n-order Markov co-occurrence model. According to the definition, if there is a co-occurrence relation between the two words within the distance n , an edge need to be added between the two nodes. By processing all the sentences in the corpus according to this method, a word co-occurrence network can be constructed. This method can not only fully reflect the contextual information between words, but also control the complexity of the model better[9]. According to Liu's study on word co-occurrence network[9], it is ideal when the distance is 2. The specific rule is: represent the words in the translated Chinese corpus as nodes in the co-occurrence network, and then create an edge between the two nodes in the same sentence.

We choose a sentence from a translated Chinese corpus ZCTC randomly[11]and take it as an example, “他一面开车，一面思忖着在教堂和牧师寓所听到的那些话，更不用说教堂外发生的事了。” ("While driving, he was thinking about the words heard in the church and the pastor's apartment, not to mention the words happened outside the church."). After removing punctuation and segmenting the sentence, the word co-occurrence network is shown as Fig 1.

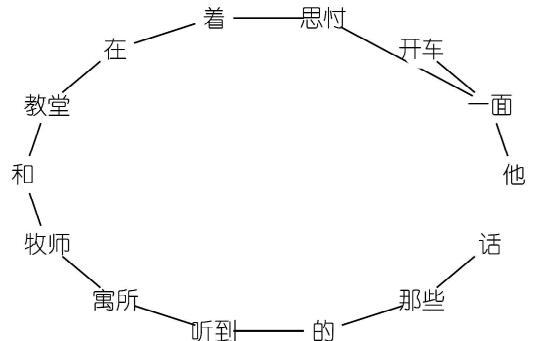


Figure 1. a co-occurrence network of translated Chinese

B. parameters and characteristics

The following parameters are usually used and analyzed when studying a complex network:

- **the number of nodes:** An actual network can be abstracted into a graph consisting of node sets and edge sets. Where, the number of nodes refers to how many nodes in this network, denoted by N .
- **the number of edges:** the sum of edges in a network, denoted by E .
- **degree:** the number of edges connecting the node i in an undirected network, denoted by k_i .
- **average degree:** the average degrees of all nodes in a network, denoted by $\langle k \rangle$

$$\langle k \rangle = \frac{\sum_i k_i}{N} \quad (1)$$

- **degree distribution:** select a node randomly from the network, the probability with a degree k among all the nodes, denoted by $p(k)$

$$p(k) = \sum_{i=k} \Pr(i) \quad (2)$$

- **cumulative degree distribution:** the probability of nodes' degrees no less than k in a network

$$P(k) = \sum_{i=k}^{\infty} \Pr(i) \quad (3)$$

- **power law index:** the index corresponding to the power law distribution function, if the degree distribution of a network obeys that distribution, denoted by γ . Where,

$$\Pr(k) \propto k^{-\gamma} \quad (4)$$

- **shortest path length:** the length of a path with the minimum edges connecting node i and node j in the network, denoted by d_{ij}
- **average shortest path length:** the average shortest path length of all pairs of nodes in the network, denoted by L

$$L = \frac{1}{N(N-1)} \sum_{i,j} d_{ij} \quad (5)$$

- **diameter:** the maximum of the shortest path lengths among all node pairs in the network, denoted by D

$$D = \max_{i,j} d_{ij} \quad (6)$$

- **reference coefficient of average shortest path length:** the average shortest path length in a random network with the same number of nodes and edges, denoted by L_r
- **clustering coefficient:** E_i delegates the actual number of edges of node i when it is connected by k nodes. Then the ratio between the maximum number of edges $k(k-1)$ and E_i is the clustering coefficient, denoted by C_i

$$C_i = \frac{2E_i}{k(k-1)} \quad (7)$$

- **average clustering coefficient:** the average clustering coefficients of all nodes in the network, denoted by C

$$C = \frac{1}{N} \sum_{i=1}^N C_i \quad (8)$$

- **reference coefficient of average clustering coefficient:** the average clustering coefficient in a random network with the same number of nodes and edges, denoted by C_r

As to a complex network, it usually holds the following two characteristics:

- **Small world effect**

The average shortest path length and clustering coefficient are important indicators for measuring whether the network has small world effect. The small world effect means that although the network is very large, there is a short path between any two nodes in the network, and the clustering coefficient is much larger than that of a random network, that is $L \approx L_r$, $C >> C_r$.

- **Scale-free property**

If the degree distribution of a network follows power-law distribution, it has the scale-free property.

IV. EXPERIMENT AND ANALYSIS

A. Corpus collection and preprocessing

Our experimental corpus is mainly from The ZJU Corpus of Translational Chinese(ZCTC). ZCTC was created, taking The Lancaster Corpus of Mandarin Chinese(LCMC) as a reference. The range of material involves 15 different types of written texts, such as Press reportage, Religious writing, Popular lore and General fiction. A total of 500 samples with 2000 words each was selected. The corpus size is one million words. The ZCTC corpus is one of the earliest translated Chinese corpus with high quality. In addition, ZCTC takes the balance of category into consideration well, so it can be used to investigate the characteristics of the word co-occurrence network of translated Chinese. About the detailed introduction to the corpus, please refer to the related works of Richard Xiao[11] and Guangrong Dai[6].

In addition, we also explore complex network characteristics of various translated Chinese texts. Because the number of texts in each category is different in ZCTC, we rearranged the 15 types of texts in a descending order according to the number of texts. And then selected out 6 types of texts with the maximum numbers. By doing that, we had sufficient corpora to carry out experiments and ensured the rationality of the experimental results. The six types are: Press reportage, Skill/trade/hobby, Popular lore, Biography and essay, Miscellaneous, and Science-academic prose. Then we randomly selected 30 texts from each type to carry out experiments later.

When preprocessing the corpus, we mainly removed the punctuation in the corpus and separated the sentences. Then with the Language Technology Platform(LTP), we segmented the sentences into words.

B. Characteristics of word co-occurrence networks of translated Chinese

1) Small world effect

In order to study the characteristics of word co-occurrence network of translated Chinese from various perspectives, we collected the whole corpus ZCTC, and the sub-corpora according to the six categories mentioned in section 4.1. As to the 7 corpora, we calculated the corresponding parameters according to section 3.2. The results are shown as Table I.

From Table I, all the 7 word co-occurrence networks have shorter average shortest path L , and $L \approx L_r$. In all networks, the average shortest path of Science-academic prose is the smallest, while Popular lore's is the largest. It indicates that the word usage is more compact in Science-academic prose, but Popular lore's is more sparse; In addition, as to the average clustering coefficient, there exists $C >> C_r$ in all the seven networks, where the clustering coefficient of Miscellaneous-reports and the official document is the largest, which indicates that the inter-connectivity of words in this category is the closest..

It can be seen that there is a significant small world effect both in the whole corpus and the six sub-corpora. Besides, the word usage of translated Chinese is more compact, and the distance between any two words in this network is no more than 11. It means that the word linkage is more closer, and the word grouping is more obvious. The above analysis indicates that although translated

Chinese might be different from standard Chinese in some aspects, it still takes Chinese as language carrier. When using translated Chinese to bear and convey information, it still needs to be complied with expression forms and habits

of this language. Therefore, as to small world effect commonly owned by all human languages, translated Chinese also hold this characteristic.

TABLE I. STATISTICAL PARAMETERS OF WORD CO-OCCURRENCE NETWORK OF TRANSLATED CHINESE

Type	Length	N	E	D	$\langle k \rangle$	L	L_r	C	C_r
A	60k	8820	30359	10	6.8841	3.1789	4.9264	0.1793	8.7717×10^{-4}
E	60k	7673	28679	9	7.4753	3.1529	4.6751	0.1876	8.0688×10^{-4}
F	60k	9027	29782	11	6.5984	3.2019	5.0425	0.1780	7.2260×10^{-4}
G	60k	8942	30109	10	6.7343	3.1624	4.9952	0.1910	8.5438×10^{-4}
H	60k	5724	22507	8	7.8641	3.0762	4.4305	0.2014	1.4227×10^{-3}
J	60k	6870	27289	10	7.9444	3.0751	4.4979	0.1952	1.1433×10^{-3}
Total	1M	47296	318782	9	13.4803	2.8911	4.4345	0.3372	2.7233×10^{-4}

Note: Considered that the space, we represent every sub-corpus according to their category numbers in corpus ZCTC, they are:

- 2)A: Press reportage 2)E: Skill/trade/hobby 3)F: Popular lore 4)G: Biography and essay
5)H: Miscellaneous-reports and official document 6)J: Science-academic prose

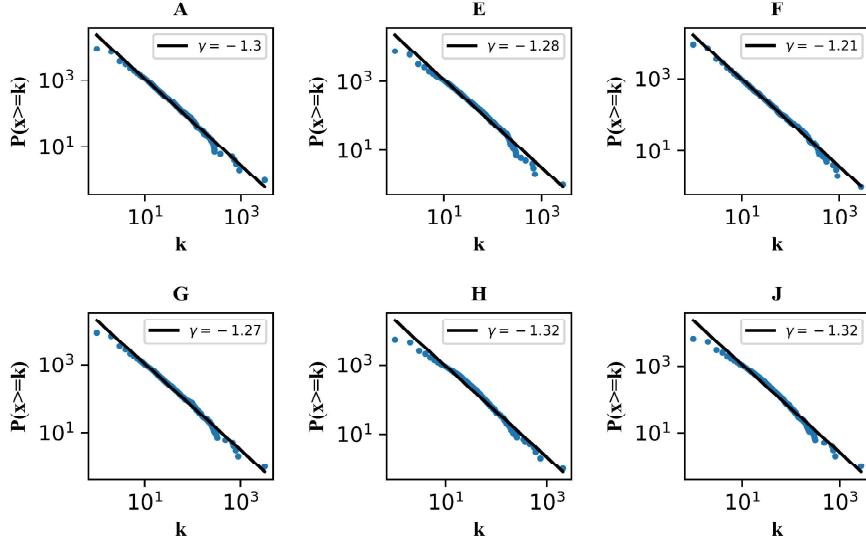


Figure 2. Cumulative degree distribution of word co-occurrence network about 6 various sub-corpus

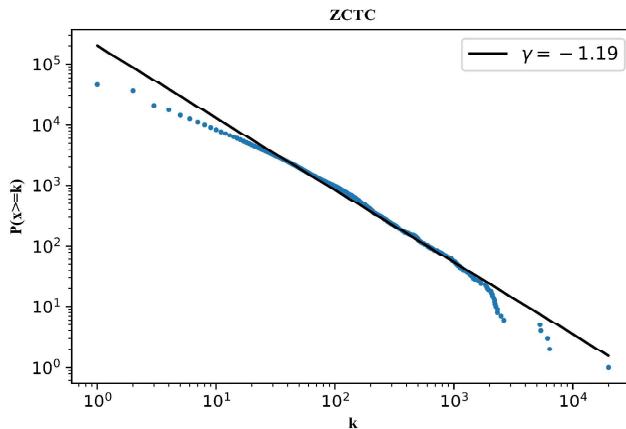


Figure 3. Cumulative degree distribution of word co-occurrence network about ZCTC

TABLE II. STATISTICAL PARAMETERS OF WORD CO-OCCURRENCE NETWORK OF TRANSLATED CHINESE IN THE ASPECT OF SCALE-FREE PROPERTY

Type	γ	R2
A	-1.3002	0.9879
E	-1.2797	0.9794
F	-1.2100	0.9927
G	-1.2701	0.9893
H	-1.3213	0.9803
J	-1.3191	0.9794
Total	-1.1898	0.9832

2) Scale-Free Property

We also measured their cumulative distributions, as shown in Fig. 2 and Fig. 3. Those results also follow the power-law distribution according to the location of points in these figures. It indicates that all the networks hold the scale-free property.

For further exploration, we fit parameters of these power law distributions. The power law indexes and R-squared are shown in Table II. From this table, the results of R-squared can all pass the test, and these power law indexes all belong to (1, 2]. Wang Lin pointed out[12] that when the power law index belongs to (2,3], the network obeys a mechanism of priority link; While belonging to (1,2], there may be other link mechanisms besides the priority link. We try to explain this phenomenon. In Zhiyuan Liu's study[9], the words with higher degrees are usually function words in Chinese, which are used to ease the sentences. When a node representing a word enters the network system, in addition to following the mechanism of priority link, it must be collocated with some function

words, so that the sentences can follow the grammar rules. The particular law is also suitable to translated Chinese, so that we can obtain the parameter fitting results in Table II. In addition, if these function words are removed from sentences, people can still understand the meaning, but the network behind it will become fragmented[13].

C. Parameter comparison between word co-occurrence networks of translated Chinese and standard Chinese

In order to reveal the speciality of the word co-occurrence network of translated Chinese better, we constructed a standard Chinese network to compare with it. The corpus was specifically selected from The Lancaster Corpus of Mandarin Chinese(LCMC), which is a blueprint for the construction of ZCTC. The two corpora are comparable both in total size and the proportion of text types. According to the same experimental steps, we measured the word co-occurrence network parameters of the LCMC corpus and the corresponding six sub-corpora, as shown in Table III.

By Comparing Table III to Table I above, between the corresponding corpora, the number of nodes in translated Chinese network is less than the standard Chinese's, but the average degree $\langle k \rangle$ of translated Chinese network is also larger than standard Chinese's. Those results above are all caused by the characteristic "simplification" in translated Chinese. That is to say, under the same scale of corpus, there are fewer word types in translated Chinese, but the use of vocabulary in standard Chinese is more diverse. In addition, the diameter and average shortest path lengths in these translated Chinese networks are often shorter than the corresponding standard Chinese's, but their clustering coefficients are higher than the standard Chinese's. This is because words are more closely related in translated Chinese, and the usage frequencies of common words are more higher.

TABLE III. STATISTICAL PARAMETERS OF WORD CO-OCCURRENCE NETWORK OF STANDARD CHINESE

Type	Length	N	E	D	$\langle k \rangle$	L	Lr	C	Cr
A	60k	10428	30253	12	5.8023	3.3975	5.4563	0.1314	4.9969×10^{-4}
E	60k	9243	27890	10	6.0348	3.3504	5.2787	0.1332	5.9738×10^{-4}
F	60k	9847	29758	11	6.0441	3.3673	5.3164	0.1402	4.8351×10^{-4}
G	60k	10755	29863	11	5.5533	3.4096	5.6029	0.1404	4.6109×10^{-4}
H	60k	6096	22422	10	7.3891	3.3579	4.5870	0.1422	1.1883×10^{-3}
J	60k	6935	26546	13	7.6557	3.1894	4.5791	0.1752	7.8820×10^{-4}
Total	1M	59743	342959	10	11.4811	3.0423	4.7693	0.2714	2.0183×10^{-4}

V. CONCLUSION

In this paper, it starts from the viewpoint of complex network to build the word co-occurrence network with ZCTC and its six sub-corpora, such as Press reportage, Skill/trade/hobby and Popular lore. By counting the corresponding parameters of these networks, it reveals and verifies the small world effect and scale-free property of the translated Chinese network. Then compared with the standard Chinese network, we discover that the translated Chinese network has less nodes, higher average degree and clustering coefficient, which further reveals the

characteristics of the translated Chinese: "simplification" and higher use frequency of common words.

Our experiment in this paper has verified that the complex network is effective as a measure of language and translation research. Meanwhile, we also realize that the above indexes are far from revealing the other characteristics of translated Chinese completely. In future, an more scientific and comprehensive index system will be formulated to deeply explore laws and characteristics in the translated Chinese network.

REFERENCES

- [1] M. Baker, "Corpus-based Translation Studies: The challenges that Lie Ahead," Benjamins Translation Library, 1996, 175-186.
- [2] M. Olohan, "Introducing Corpora in Translation Studies," Routledge, 2004.
- [3] Vinay, J. Paul and J. Darbelnet. "Comparative stylistics of French and English: A methodology for translation", John Benjamins Publishier, 1995.
- [4] Y. Zhu and X. Li. " A Quantitative Study on Lexical Features of Translated Chinese: Based on the Corpus of E-C Translated News Articles and the Corpus of Chinese News Articles(in Chinese)," Foreign Languages in China, vol. 88, Apr. 2019, pp. 81-90.
- [5] R. Xiao and G Dai, " Lexical and grammatical properties of Translational Chinese:Translation universal hypotheses reevaluated from the Chinese perspective," Corpus Linguistics and Linguistic Theory, vol. 10, Jun. 2013, pp.11-55.
- [6] G. Dai, "Collocational Features in Translated Chinese: A Case Study of Source Language(SL)(in Chinese)," Contemporary Foreign Languages Studies, vol.229,Jan. 2013, pp.50-55.
- [7] H. Liu, "Language Is a Complex Network(in Chinese)," Journal of Shanxi University(Philosophy and Social Science Edition), vol.167, Sep. 2013, pp. 66-69.
- [8] R. Cancho and R. Sole, "The Small World of Human Language," Proc. the Royal Society B: Biological Sciences, The Royal Society, July. 2001, pp.2261-2265.
- [9] Z. Liu, M. Sun, "Chinese Word Co-occurrence Network:Its Small World Effect and Scale-free Property(in Chinese)," Journal of Chinese Information Processing, vol.96, Nov. 2007,pp.52-58.
- [10] Z. Liu,Y. Zheng and M. Sun, "Complex Network Properties of Chinese Syntactic Dependency Network(in Chinese)," Complex Systems and Complexity Science, vol 18.Jun 2008, pp.37-45.
- [11] Z. Xiao, "Corpus-Based Studies of Translational Chinese in English-Chinese Translation(In Chinese)," Shanghai,CN: Shanghai Jiao Tong University Press, 2012.
- [12] L. Wang and G. Dai, "On Degree Distribution of Complex Network(in Chinese)," Journal of Northwestern Polytechnical University, vol. 24, Aug 2006, pp.405-409.
- [13] M. Kurant, P. Thiran and P. Hagmann, "Error and attack tolerance of layered complex networks," Nature, vol. 76, Jan. 2007, pp. 388-394.

Diachronic Synonymy and Polysemy:

Exploring Dynamic Relation Between Forms and Meanings of Words Based on Word Embeddings

Shichen Liang § †, Jianyu Zheng‡, Xuemei Tang § †, Renfen Hu § †, Zhiying Liu § †

§ Institute of Chinese Information Processing, Beijing Normal University, Beijing China

†UltraPower-BNU Joint Laboratory for Artificial Intelligence, Beijing China

{shichen, tangxuemei, irishu}@mail.bnu.edu.cn, liuzhy@bnu.edu.cn

‡Advanced Innovation Center for Future Education, Beijing Normal University, Beijing China

zheng_jianyu@126.com

Abstract—In recent years, there has been a large number of publications that use distributed methods to track temporal changes in lexical semantics. However, most current researches only state the simple fact that the meaning of words has changed, lacking more detailed and in-depth analysis. We combine linguistic theory and word embedding model to study Chinese diachronic semantics. Specifically, two methods of word analogy and word similarity are associated with diachronic synonymy and diachronic polysemy respectively, and the aligned diachronic word embeddings are used to detect the changes of relationship between forms and meanings of words. Through experiments and case studies, our method achieves the ideal result. We also find that the evolution of Chinese vocabulary is closely related to social development, and there is a certain correlation between the polysemy and synonymy of the word meaning.

Keywords—*Diachronic Synonymy, Diachronic Polysemy, Lexical Semantic, Word Embedding*

I. INTRODUCTION

In various subsystems of the language, vocabulary system holds the fastest changes. As the most basic unit of vocabulary system, a word is a relative unity of form and meaning. With the development of society and cultural communication, the connotation and denotation of a word's meaning is changing. On the one hand, the same word will hold different meanings. That is, old words are with new meanings; On the other hand, the same concept will be also expressed by different words. Faced with that situation, we combined word embeddings and some theories in linguistics, and explored the evolution of lexical semantics from a diachronic perspective.

In recent years, with the boom of deep learning, the word embeddings which can vectorize vocabularies are also used extensively. As each word is represented distributively, the semantic relationship between words can be examined by distances between vectors. For example, as to the following two pairs of words: "Dad" - "Mom", "Dad" - "Apple", by calculation, the similarity of the previous word pair is much higher than the later, which is consistent with people's cognition. In addition, word embeddings can also be used for semantic analogy between words. That is, given the first three words, the fourth word can be automatically inferred by calculations among their vectors. The most famous example is: king + man - woman = queen. Because word embeddings can describe the words' meanings to some extent, they are used in many tasks of natural language processing.

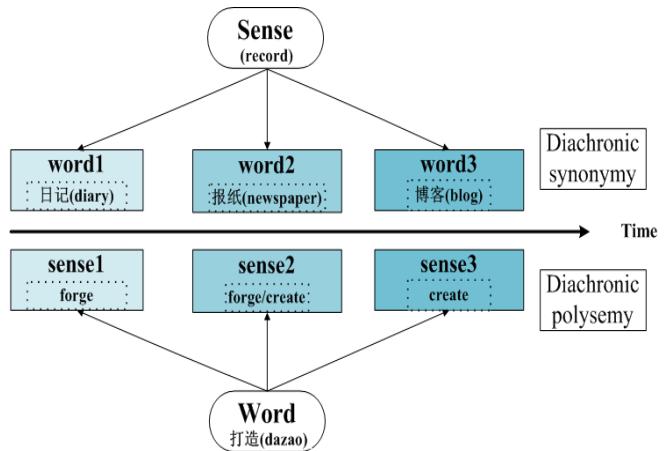


Fig. 1. A sketch map about dynamic relation between forms and meanings of words. As shown in the upper half of the figure, the word “blog” does not exist in the past. The meaning contained in the word “blog” was represented by “diary” in the past period, so the two words hold the semantic equivalence from the diachronic perspective.

In this paper, we study the relation between forms and meanings of words from a diachronic perspective based on the word embeddings. Firstly, according to the diachronic corpus, word embeddings during different periods are respectively trained, and the vector spaces are aligned by a certain alignment method. Then, from the perspectives of diachronic synonymy and diachronic polysemy, this paper explores the dynamic relation between forms and meanings of words. As shown in figure1: from the perspective of diachronic synonymy, it is possible to observe whether the same meaning is represented by different word forms in different periods by the diachronic word analogy; And from the perspective of diachronic polysemy, whether the same form has different meanings in different periods by the diachronic word similarity. The main contributions of this paper are as follows:①the word analogy and word similarity are applied to the study of diachronic semantics, which is different from the previous work, using these two methods to evaluate the quality of word embeddings;②combining the two perspectives of diachronic synonymy and diachronic polysemy, we study the dynamic relation between forms and meanings of words.

II. RELATED WORKS

The evolution of word meaning is a common phenomenon in all languages, and has always been one of the central topics concerned by linguists^[1]. Initially, researchers inferred the changes of word meaning by subjective language usage preferences. Later, with the establishment of large-scale corpus, scholars carried out the

research by the "data-driven" approach from a quantitative perspective^[2]. Specifically, by using the corpus to query how a word is used in the context, the evolution of a word's meaning can be summarized^{[3][4][5][6]}. Among those, counting the usage frequency of a certain word under the diachronic condition is the most typical method^{[7][8][9]}. However, it also has many shortcomings, such as the failure in characterizing and measurement of the relative semantic relationship with other words.

With the deep learning being widely used in natural language processing, word embedding has become an effective way to represent words quantitatively. The word embedding model was originally proposed by Mikolov^[10]. This model can learn unsupervised low-dimensional dense representations of each word from large-scale texts. Later, researchers applied this technique to explore the evolution of word meaning^{[11][12][13][14][15]}. Kim et al. ^[16] investigated the semantic variation of certain words by calculating the cosine similarity between word embeddings, such as the words "cell" and "gay". It is worth to mention that they used incremental updates and Continuous Skip-gram with negative sampling (SGNS) to improve the representation of the word. By calculating the similarity between word embeddings, Kulkarni et al.^[17] observed the neighbor words of a given word in different periods, and judged whether the meaning of word has evolved. Later, from the perspective of word analogy, Szymanski^[18] first proposed Temporal Word Analogies to induce the corresponding analogical relation between words during different periods, such as "Ronald Reagan in 1987 is like Bill Clinton in 1997".

In summary, the study of diachronic meaning by word similarity has gained some achievements. But there is relatively little work in word analogy, especially in Chinese. At the same time, there is no study to analyze lexical semantics macroscopically by combining forms and meanings of words. Inspired by structural linguistics^[19], this paper takes roots in the evolution of lexical semantic systems from two perspectives: diachronic synonymy and diachronic polysemy. At the same time, we apply the word analogy and word similarity of word embedding to observe the two linguistic phenomena, and analyze the dynamic changes of lexical semantics.

III. TRAINING DIACHRONIC WORD EMBEDDINGS

Before carrying out our experiment and analysis, we need to train diachronic word embeddings based on the corpus from different periods, so that we can observe the evolution relation between word meaning and word form in a diachronic level. Essentially, diachronic word embeddings are a set of aligned synchronic word embeddings. In this section, we constructed word embeddings in each decade from 1947 to 2017 with word2vec, and then aligned those vector spaces to get comparable diachronic word embeddings.

A. Word2vec

Word embeddings are low-dimension word representations trained from unlabeled corpora through neural models. According to Firth (1957)^[20], distributional hypothesis is a theory claims that words appearing in similar contexts must have similar meanings and representations. This hypothesis is one of the theoretical bases of word embedding model.

At present, the word representation methods most commonly used include PPMI, SVD, Word2Vec, Glove and so on. The embeddings trained by Word2vec are optimized to predict co-occurrence relationships. Especially, given the input word, skip-gram model can predict the context information which is more suitable for studying semantic change. For this reason, we finally adopted the skip-gram model to construct word embeddings...

B. Datasets, pre-processing, and hyper-parameters

We collected the news in People's Daily from 1948 to 2017 to construct a corpus, as this newspaper is a national print media with the largest scale and the highest authority in China. All news spans 70 years, whose total size is 4.43GB. Taking social realities and experimental operability into account, we divided the data by 10 years, so there were 7 different vector space models.

Years	Tokens	Size(KB)
1948-1957	116382	430,786
1958-1967	157822	274,846
1968-1977	138431	451,674
1978-1987	150195	449,358
1988-1997	175085	354,597

TABLE I. STATISTICS OF 7 VECTOR SPACE MODELS.

Those embeddings in each period were trained by skip-gram with vector size 300, window side 5, min-count 5. As Table I shows, each model contains more than 100k tokens.

C. Alignment of vector space models

During training, those vectors are usually randomly initialized. Although the mutual relations between vectors may be consistent, due to the stochastic processing, vectors not trained in the same run are not comparable with each other. For this reason, It requires us to align each independent vector space model. To address this problem, previous studies have provided three approaches: Non-random initialization, Local linear regression and Orthogonal Procrustes.

In this work, we followed Terrence Szymanski (2017)^[18]'s method: we assumed there was always a linear transformation can align two vector space models, and the meanings of most words did not change. First, we found the common words of two vector spaces and sampled them in proportion; Then, the results of sampling were respectively used as independent variables and dependent variables for linear regression, and the linear regression model was obtained by minimizing the distance (mean square error) of the two sampling spaces; Last, we applied the linear regression model to the entire vector space models.

A small change was adapted during random sampling. What we did was not randomizing completely. According to Hamilton (2016)^[21], words with higher word frequency have less change in meaning. Therefore, we made use of the HSK (an examination for testing the Chinese language level of foreigners) vocabularies issued by HanBan of China, which contains 8650 common words in Chinese. In order to get a more reasonable initial sampling, we reduced the proportion of random samples, and added 8650 words mentioned above

to the set of random words. Through this way, we constructed Chinese diachronic word embeddings.

IV. RESULTS AND ANALYSIS

We explored the evolution relation between form and meaning of Chinese vocabularies from two perspectives: diachronic polysemy and diachronic synonymy. The specific measures are diachronic word analogy and diachronic word similarity. Previously, word similarity and analogy were methods for evaluating the synchronic word embeddings. Meanwhile, they are also useful tools to reveal relation of lexical semantic. In this section, we capture word sets which convey the same meaning in different periods by diachronic word analogy. At the same time, we detect words which hold different meanings in different periods by diachronic word similarity.

A. Analogy and diachronic synonymy

With replacement of things and concepts, the same concept may be expressed in different word forms in the diachronic level. Diachronic word analogy is a technical measure to establish a synonymous relationship between those words which have unrelatedness with each other at the synchronic level. In general, analogy is to infer the unfilled word based on the relationship between two word pairs, i.e. word W1 to word W2 is equivalent to word W3 and word W4. When it comes to diachronic word analogy, that is word

$w\alpha$ at time $t\alpha$ is equivalent to word $w\beta$ at time $t\beta$, which has a slight difference with traditional word analogy. For example, “微博” (Weibo, microblog) in years of 2008-2017 is equivalent to “墙报” (qiangbao, wall newspaper) in years of 1948-1957.

In this section, we select the vector space of a specific period as the basic vector space, then align the vector spaces of the other six periods with it through the alignment method mentioned in section III.C. When given a word in the basic vector space, we carry out word analogy to find diachronic synonyms in other periods. What we need to do is to find vectors closest to the word in other six vector spaces, and get the words represented by the corresponding vectors as a result of analogy. At last, there are seven words in each set of analogy. Ideally, if the correspondence between words and meanings does not change, there should be seven identical words in each set. Therefore, if different words are used for the same meaning at different times, then the diachronic word analogy will be able to discover a semantic consistency between these words. It is worth noting that the meaning here is a relatively broad definition. For example, Xi and MAO are different names which do not point to the same person, but they share a common concept, namely Chinese President. In cases like this, we also view them as two words that convey the same meaning.

TABLE II. EXAMPLES OF WORDS FROM 2008-2017 AND THEIR ANALOGOUS OVER TIME*

1948-1957	1958-1967	1968-1977	1978-1987	1988-1997	1998-2007	2008-2017
王府井大街 WangFujing Street	餐馆 restaurant	书摊 bookstall	赌场 casino	柜台 counter	专卖店 specialty store	淘宝 Taobao
书信 Letter	信件 letter	传真 fax	信函 letter	传真 fax	电子邮件 e-mail	电子邮件 e-mail
黑板 blackboard	卖报 newspaper	名片 business card	电话 phone	手机短信 SMS	电子邮件 e-mail	微信 WeChat
墙报 wall newspaper	副刊 supplement	栏目 column	新闻媒体 media	人民网 People's Network	网站 website	微博 microblog
日记 diary	日记 diary	日记 diary	书评 book review	黑板报 blackboard	报纸 newspaper	博客 blog
顾客 customer	代售 sale	烟酒 tobacco& alcohol	小卖部 commissary	送货上门 delivery	售货员 salesperson	外卖 take-out
航线 air line	电气化铁路 railway	高速公路 highway	高速公路 highway	高速公路 highway	高速公路 highway	高铁 high-speed railway

*The last column (2008-2017) is the words in the reference period, and the vector spaces of the other six periods are respectively aligned with this period to find the words that is closest to words of 2008-2017. This table shows the result of word analogy, for example, “博客”(blog)+time α (2008-2017) = “日记”(diary)+time β (1948-1957)

We selected 50 neologism which are commonly used in 2008-2017 to constitute a small vocabulary for analogy. Those words include political words, life words, ideology, famous figures, institution, and we eventually form 300 pairs of word analogies. These new words from 2008 to 2017 are respectively compared in six other periods to get the synonyms in those periods. The results show that political

words and life words have achieved a better performance. As table II shows, the analogy of life vocabularies roughly shows the development of Chinese society, indicating that the diachronic word analogy is effective to some extent. For example, the term "blog" appeared in the Internet era, also known as "network diary". As the result shows, in 1948-1977,

before the Internet became popular, the corresponding word of this concept is exactly "diary".

B. Similarity and Diachronic polysemy

In this section, we observe old words with changeable meanings and study how those changes happen by diachronic word similarity. This issue can be discussed from two aspects. When a word holds different meanings in the diachronic level. On the one hand, the word's similarity with itself in previous period will decrease; and on the other hand, its similarity with other words will also change. The details are as follows:

1) Diachronic similarity of individual words.

The similarity between a word and itself in previous period is an important reference standard to measure the degree of meaning change. If the lower the diachronic similarity of a word is, the more remarkable its diachronic polysemy.

By the vector space model after alignment, we calculate the cosine similarity of the vector (V_i, V_{i+1}) of each word (W) in every two periods (t_i, t_{i+1}), then sum and average the results of similarities. In this way, we get diachronic similarities of words from 1948 to 2017:

$$\text{similarity} = \frac{1}{n} \sum_{i=1}^{n-1} \frac{V_i \cdot V_{i+1}}{\|V_i\| \cdot \|V_{i+1}\|} \quad (1)$$

n indicates the number of periods, here $n=7$.

TABLE III. SOME EXAMPLES OF DIACHRONIC WORD SIMILARITY.

words	diachronic word similarity
打造(forge; create)	0.132075
曝光(expose; make sth. public)	0.205325
登录(land; log in)	0.325444
家教(domestic or family education ; private teacher)	0.363201
绿色(green; environmentally friendly)	0.460422
变脸(face changing; turn hostile suddenly)	0.482308
赞(praise; favor)	0.493929
强暴(violent; rape)	0.549233
充电(charge; study)	0.549825
漫游(travel around; roam)	0.596138
阳光(sunlight ; cheerful ; transparent)	0.647266

In addition, we have collected some words from previous linguistic studies that did show diachronic polysemy. Unlike traditional linguistic research methods, word similarity can quantify the degree of change in meaning. Some examples are shown in Table III. All of these words are diachronic polysemes, but the competitiveness of the new meaning leads to the content of similarity. For example, the word "打造" (forge; create) with relatively low similarity originally

refers to "forge", it now becomes similar to "create". Two meanings of this word are so different and the original meaning was almost lost; the similarity is slightly higher for the words "家教" (domestic or family education ; private teacher) and "透明" (transparent ; non-secrecy). Those words have two meanings in parallel, and the two meanings are both used at the same period, so the degree of change is moderate; The word "阳光" (sunlight ; cheerful ; transparent) has three different meanings, and the most primitive meaning (sun's rays) is still the most frequently used. So it has relatively high similarity.

2) Diachronic similarity of multiple words

The phenomenon of diachronic polysemy can not only be reflected from the diachronic similarity of words but also from the change of synchronic synonymy of words. When the a word's meaning changes, it means that its synonyms of the word in previous period are broken, while the synonyms in the new period are established. Therefore, we can examine the diachronic polysemy and the movement direction of a word by examining the similarity of words to their neighbors at different times.

According to previous research and Modern Chinese Dictionary, we obtained 20 pairs of words: as the time passing, the meaning of one or two in those word pairs changes, and the relative distance between the pair is also altered. In other words, the semantic relationship between them could be changed. Semantic shifts can be observed as the target word moves closer to or further apart from its neighbors. We measure the movement of words by calculating the cosine similarity between word pairs over time.

If the similarity between W_1 and W_2 increases from time i to time j($j > i$), it indicates that the meanings of W_1 and W_2 are getting closer and closer. Vice versa, the gap between the two meanings is getting narrower. We assume that the meaning of W_2 has not changed, then the meaning of W_1 must have changed. That is, W_1 is a diachronic polysemy. For example, according to the first column in Table IV, there are words whose meaning have changed, and the words in the second and third columns hold stable meanings. Through the changes of similarity of multiple sets of word pairs, we can determine the direction of meaning change. For example, the word "破产" (bankrupt; fail) listed in the Table IV originally refers to the property problem of an individual or a company, after then it captures a metaphorical usage means a failure.

Finally, we find that the polysemy and synonymy of words actually change synchronously. If a word in word pairs changes its meaning, its temporal similarity to other words will also be changed. Conversely, if the diachronic similarity between the word pair changes greatly, it is likely that the meaning of one word in the pair has changed. The change of the meaning is reflected in the relationship between words. The existence of diachronic polysemy often reflects the change of word's synonymous relation in the diachronic level. That is to say, from the diachronic perspective, the polysemous relation and the synonymy relation of words are interrelated.

TABLE IV. SEMANTIC SHIFTS CAPTURED BY HISTORICAL WORD EMBEDDINGS

Words	Moving closer	Moving away	Shift time
钩钩(hook)	骗局(fraud)	钓鱼(fishing)	1998-2007

阶级(steps; class)	阶层(stratum)	台阶(steps)	1948-1957
破产(bankrupt; fail)	失败(fail)	欠债(owe a debt)	1988-1997
包袱(bundle; joke)	笑料(joke)		1978-1987
程序(procedural; programme)	电脑(computer)		1988-1997
纠结(intertwine; confused)	烦闷(depressed)		1978-1987
萌(sprout; lovely)	可爱(lovely)		1998-2007
粉(powder; fans)	粉丝(fans)		1998-2007
霹雳(thunderbolt; unexpected event)	坏消息(bad news)		1978-1987
晒(bask; share)	秀(show off)		1998-2007
山寨(village; copycatting)	假(fake)		2008-2017
小姐(miss; prostitute)	妓女(hooker)		1998-2007
宅(house; stay at home)	呆(stay)		2008-2017
总裁(director-general; chairman)	总经理(GM)		1978-1987

V. CONCLUSION

Based on the diachronic word embedding model, this paper uses word similarity and word analogy to investigate the dynamic relation between word form and semantics over time. According to the experiment results from some typical cases, our method has achieved ideal performance. In diachronic word analogy, we can find the corresponding words of the new words in the past, which can realize the semantic continuity between new and old vocabularies; In diachronic word similarity, we can detect the generation and development of the new meaning of the old words, which can achieve the semantic continuity of the same words. In addition, we also find that the lexical semantic change of Chinese vocabularies is closely related to the development of Chinese society.

However, as to the same word form, there is only a vector representation trained by word2vec generally, so the multiple meanings of the words could not be distinguished, and we could not explore the semantic changes thoroughly. In addition, the BERT model risen recently can also better deal with the problem of polysemous words, and we expect the idea in this paper can achieve more performance on BERT.

REFERENCES

- [1] Baugh, A. C. & T. Cable. A History of the English Language. London: Routledge, 1993.
- [2] Elizabeth Traugott. Semantic change. Oxford Research Encyclopedias: Linguistics, 2017.
- [3] M. Hilpert. Germanic future constructions: A usage-based approach to language change. Benjamins, Amsterdam, Netherlands, 2008.
- [4] Stefan Th. Gries. 1999. Particle movement: a cognitive and functional approach. Cognitive Linguistics, 2008, pp. 105–145.
- [5] D. Kerremans, S. Stegmayr, and H.-J. Schmid. The NeoCrawler: Identifying and retrieving neologisms from the internet and monitoring ongoing change. In K. Allan and J. A. Robinson, editors, Current methods in historical semantics, De Gruyter Mouton, 2010, pp. 130–160.
- [6] Kristina Gulordava and Marco Baroni. A distributional similarity approach to the detection of semantic change in the Google Books Ngram corpus. In Proceedings of the GEMS 2011 Workshop on Geometrical Models of Natural Language Semantics, Edinburgh, UK, 2011, pp. 67–71.
- [7] Patrick Juola. The time course of language change. Computers and the Humanities, 2003, pp. 37(1):77–96.
- [8] Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K. Gray, Joseph P. Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, Steven Pinker, Martin A. Nowak, and Erez Lieberman Aiden. Quantitative analysis of culture using millions of digitized books. Science, 2011, pp. 331(6014):176–182.
- [9] Hyunyoung Choi and Hal Varian. Predicting the present with Google trends. Economic Record, 2012, pp. 88(s1):2–9.
- [10] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems, 2013, pp. 26:3111–3119.
- [11] Peter Turney, Patrick Pantel, et al. From frequency to meaning: Vector space models of semantics. Journal of artificial intelligence research, 2010, pp. 37(1):141–188.
- [12] Hosein Azarbonyad, Mostafa Dehghani, Kaspar Beelen, Alexandra Arkut, Maarten Marx, and Jaap Kamps. Words are malleable: Computing semantic shifts in political and media discourse. In Proceedings of the ACM on Conference on Information and Knowledge Management, Singapore, 2017, pp. 1509–1518.
- [13] Hao Peng, Jianxin Li, Yangqiu Song, and Yaopeng Liu. Incrementally learning the hierarchical softmax function for neural language models. In Proceedings of the 31st AAAI Conference on Artificial Intelligence, San Francisco, California USA, 2017, pp. 3267–3274.
- [14] Yating Zhang, Adam Jatowt, Sourav Bhowmick, and Katsumi Tanaka. Omnia mutantur, nihil interit: Connecting past with present by finding corresponding terms across time. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 2015, pp. 645–655.
- [15] Steffen Eger and Alexander Mehler. On the linearity of semantic change: Investigating meaning variation via dynamic graph models. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 2016, pp. 52–58.
- [16] Yoon Kim, Yi-I Chiu, Kentaro Hanaki, Darshan Hegde, and Slav Petrov. Temporal analysis of language through neural language models. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014, Baltimore, USA, pp. 61–65.
- [17] Vivek Kulkarni, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. Statistically significant detection of linguistic change. In Proceedings of the 24th International Conference on World Wide Web, 2015, Florence, Italy, pp. 625–635.
- [18] Terrence Szymanski. Temporal word analogies: Identifying lexical replacement with diachronic word embeddings. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2018, Vancouver, Canada, pp. 448–453.
- [19] De Saussure F. Course in general linguistics. Columbia University Press, Irvington, 2011.
- [20] John Firth. A synopsis of linguistic theory, 1930-1955. Blackwell, 1957.
- [21] William L. Hamilton, Jure Leskovec, and Dan Jurafsky. Diachronic word embeddings reveal statistical laws of semantic change. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 2016, Berlin, Germany, pp. 1489–1501.

Automatic answer ranking based on sememe vector in KBQA¹

Yadi Li, Lingling Mu, Hao Li, Hongying Zan

*School of Information Engineering, Zhengzhou University, Zhengzhou, P.R.China
1837186472@qq.com; iellmu@zzu.edu.cn*

Abstract—This paper proposes an answer ranking method used in Knowledge Base Question Answering (KBQA) system. This method first extracts the features of predicate sequence similarity based on sememe vector, predicates' edit distances, predicates' word co-occurrences and classification. Then the above features are used as inputs of the ranking learning algorithm Ranking SVM to rank the candidate answers. In this paper, the experimental results on the data set of KBQA system evaluation task in the 2016 Natural Language Processing & Chinese Computing (NLPCC 2016) show that, the method of word similarity calculation based on sememe vector has better results than the method based on word2vec. Its accuracy, recall rate and average F1 value respectively are 73.88%, 82.29% and 75.88%. The above results show that the word representation with knowledge has import effect on natural language processing.

Keywords-KBQA; word similarity; Ranking SVM; sememe vector

I. INTRODUCTION

The automatic Question Answering (QA) system is the basic form of the next generation search engine^[1], and its main features are as follows: (1)Users' inputs are natural language questions. (2)The returned answer is no longer the form of sorted documents, but a direct answer.

The basement of the automatic QA is to deeply understand the natural language questions of the users and extract the meanings from them, and then the corresponding answers are obtained according to the users' questions. KBQA is one of the forms of automatic QA, which generates answers from knowledge base. An example of question mapping and candidate answers extraction is shown in Table I.

TABLE I. The example of question mapping and candidate answers extraction

The “entity-predicate” mapping of the question	轴承(bearing)–直径(diameter) 多少(how many)
The set of corresponding candidate answers	轴承 OD (外径 outside diameter) 170mm
	轴承 d (内径 inside diameter) 80mm
	轴承 外文名(English name) bearing

At present, there are some available large-scale knowledge bases, such as DBpedia^[2], Freebase^[3] and YAGO^[4], etc. Most of these knowledge bases are graph structures composed of “entity-relationship-entity” triples as the basic units. Therefore, transforming natural language questions into structured queries becomes the basic work of searching answers in the knowledge bases.

With the development of natural language processing technology, there are mainly two forms of solutions to the

KBQA. One is the method based on semantic analysis, which uses the method of semantic analysis to analyze questions and convert natural language questions into more advanced expressions or standard question statements. The other is based on the method of information extraction, which firstly extracts the key topic entities and relationships in the questions. Then it searches the candidate answer entities in the knowledge bases that are related to the key entities. Finally, it chooses the most possible answer by calculating the similarities or correlations between the candidate answers and the questions.

Word similarity is often used in KBQA. When the predicates sequences in the questions are different from that in the knowledge base, it is necessary to sort the candidate answers in the knowledge base by calculating the similarities between questions' predicates and candidate answers' predicates. The word similarity can be simply calculated by the word vectors that can be trained by Word2vec^[5] and Glove^[6] and so on. Word2vec and Glove use unsupervised learning method to train word vectors based on a huge amount of unlabeled data. However, they ignore the knowledge information and sometimes cannot correctly distinguish the semantic difference.

The method proposed by this paper is based on the method of information extraction. Firstly, we extract the entities and relationships in the questions. Then, we extract all candidate answers that contain the questions' entities in the knowledge base. Finally, we use the Ranking SVM algorithm with four features to select the most possible answer. This paper uses sememe vector to calculate the word similarity and get a better result on limited training data.

II. RELATED WORK

The research of KBQA has a long history in natural language processing. The common techniques for KBQA are the methods based on semantic analysis, feature-driven and representation learning^[7].

The method based on semantic analysis maps questions in natural language forms to semantically equivalent logical expressions through certain grammars^[8]. Cai et al.^[9] used the purely supervised learning method to train a semantic analyzer. They developed a matching algorithm to find the word-related labels in the knowledge base and established a lexicon extender. The lexicon extender linked the words to the relevant labels in the knowledge base to complete the semantic analyzer learning. However, this method still had some problems. It cannot get rid of the dependence on manual annotation, and it requires higher

¹ The authors were supported financially by the National Social Science Fund of China (18ZDA315), Programs for Science and Technology Development in Henan province (No.192102210260) and the Key Scientific Research Program of Higher Education of Henan (No.20A520038).

accuracy of word-related labels in the lexicon extender. Kwiatkowski et al.^[10] proposed a solution based on ontology matching, which is independent of artificial word triggers. Yao et al.^[11] used question words, question intentions and entity types to construct the questions' feature graphs. For each edge $e(s, t)$ in the graphs, “ $e, s, s|t$ ” are extracted as the features of questions to search for answers. Lai et al.^[12] combined the predicate sequence similarity with the entities' lengths and the frequency features of the answers' templates to rank the candidate answers and select the most possible answer. Kun Xu et al.^[13] selected three types of features in the QA system based on the freebase: the entities' scores based on entities linking technology, the predicates' scores based on Convolutional Neural Network (CNN), the co-occurrences of the answer types and the question words. They used the machine learning algorithm to find answers with three types of features.

This paper proposes an answer ranking method used in KBQA system, which combines the features of predicates' word similarity, edit distances, word co-occurrences and classification with the ranking SVM algorithm to rank the candidate answers.

III. WORD SIMILARITY BASED ON SEMEME VECTOR

In HowNet², there are various relationships among sememes. The upper and lower relationships organize all sememes in a hierarchical graph. We can use the upper and lower position of sememes and combine the PageRank^[14] algorithm to represent the sememes as vectors. Then the sememe similarity is calculated by the sememe vector, the concept similarity is calculated by the sememe similarity. Finally, the word similarity is calculated by concept similarity.

A. Sememe Information Content (SIC)

The SIC refers to the size of information contained in a sememe itself. It is an important feature to distinguish the degree of difference between sememes. If SIC is very close, the similarity between sememes is higher. In the sememe structure graphs, the structural information content of the sememe nodes is mainly considered. The deep structure information of the sememe nodes is an important feature of SIC. In order to further refine the differences among sememes, the layers in which the descendants' nodes are located should be considered. Therefore, we can calculate the SIC by using the sememe structures according to the method of reference[15]. The calculation formula is the following formula (1).

$$SIC_s = \frac{\log(deep(s)+1)}{\log(deep_{max}+1)} \times \left(1 - \frac{\log\left(\sum_{\alpha \in hypo(s)} \frac{1}{deep(\alpha)} + 1\right)}{\log(node_{max})}\right) \quad (1)$$

In the sememe structure graph, $deep(s)$ represents the layers of the sememe node s . The root node $ROOT$ is the first layer, $deep_{max}$ is the depth of the layer where the last layer is located, α is the descendant node of the sememe s , $hypo(s)$ is the set of all the descendants of sememe s , $node_{max}$ is the total number of sememe nodes.

²<http://www.keenage.com/>

B. The Representation of Sememe Vector

In the structure graph of sememes, the transfer probability matrix M is constructed by SIC.

M is defined as a matrix of $N \times N$. N is the total number of sememe nodes. All nodes are traversed in the sememe structure graph, and each element in M is assigned a value according to the formula (2).

$$M_{ji} = \begin{cases} \frac{SIC_j}{\sum_{k \in Out(i)} SIC_k}, & \text{If } i \text{ and } j \text{ have a connection} \\ 0, & \text{else} \end{cases} \quad (2)$$

Where $Out(i)$ is a set of all nodes connected to the sememe node i , SIC represents the information content of the sememe nodes.

Each sememe is represented as a vector based on the transition probability matrix M and PageRank algorithm. The sememe vectors are calculated by the following formulas:

$$\vec{p}_s = cM \cdot \vec{p}_s + (1-c)\vec{v}_s \quad (3)$$

$$\vec{v}_s(j) = \begin{cases} \frac{SIC_j}{\sum_{k \in Out(s)} SIC_k}, & \text{If } s \text{ and } j \text{ have a connection} \\ 0, & \text{else} \end{cases} \quad (4)$$

Where c is the damping coefficient between 0 and 1, M is the transition probability matrix that represents the SIC of each sememes. The vector \vec{p}_s is the vector representation of the sememe s , its dimension is N and initial value is $\{\frac{1}{N}, \frac{1}{N}, \dots, \frac{1}{N}\}$. The value of \vec{p}_s is changed during the iterations until it converges. The vector \vec{v}_s has N dimensions.

C. The Calculation of Word Similarity

After representing the sememes as vectors, the similarities between the sememes are calculated by the cosine similarity of the vectors. According to the description structures of word concept based on sememes in HowNet, the four parts of the sememe set similarity are calculated by the sememe similarity, and then the concept similarity is calculated. We take the maximum similarity among two concepts as the word similarity^[15]. The specific calculation is described as follows:

Formulas (5)-(6) are used to calculate the similarity of sememe i and j .

$$dis(i, j) = \cos(\vec{P}_i, \vec{P}_j) \quad (5)$$

$$sim_{s(i,j)} = \frac{dis(i,j)}{\alpha + dis(i,j)} \quad (6)$$

Where α is an adjustable parameter.

The different semantics of words are described by various concepts, so sometimes there are many similarities between the concepts. It is necessary to calculate the similarities among two concepts separately.

The concept similarity of concepts C1 and C2 consists of four parts: the first basic sememe similarity $sim_1(s_1, s_2)$,

the other basic sememe similarity $\text{sim}_2(s_1, s_2)$, the relational sememe similarity $\text{sim}_3(s_1, s_2)$, the symbolic sememe similarity $\text{sim}_4(s_1, s_2)$. Therefore, the concept similarity between the two concepts is calculated by the formula (7):

$$\text{sim}_c(c_1, c_2) = \sum_{i=1}^4 \beta_i \prod_{j=1}^i \text{sim}_j(s_1, s_2) \quad (7)$$

Where β_i is an adjustable parameter.

Suppose the word w_1 has m concept descriptions: $c_{11}, c_{12}, \dots, c_{1m}$, and the word w_2 has n concept descriptions: $c_{21}, c_{22}, \dots, c_{2n}$. Then the maximum value is taken as the word similarity by the reference[16]

$$\text{sim}_w(w_1, w_2) = \max_{i=1, \dots, m, j=1, \dots, n} (\text{sim}_c(c_{1i}, c_{2j})) \quad (8)$$

IV. ANSWER RANKING METHODS IN KBQA

The answers of KBQA are from a structured knowledge base, which usually exist in the form of triples. The structures of the knowledge base have various triple forms: “entity-relationship-entity” and “entity-attribute-attribute value”, etc. In the knowledge base, this paper takes the “relationship” and “attribute” as “predicate”, and each triple is called “assertion”.

The process of the KBQA system is divided into three steps: questions analysis, candidate answers extraction, and answers sorting. The questions analysis phase identifies the entities in the questions and maps the questions to the structured forms of “entities-predicates”. The candidate answers extraction phase searches the triple of “entity-predicate-entity” in the knowledge base according to the entities in the questions, and extracts the structural assertions of the corresponding entities as the set of candidate answers. After completing the questions analysis and candidate answers extraction, the candidate answers are sorted according to the predicate features of “entity-predicate” and “entity-predicate-entity”, and the highest score answer is returned.

The process of KBQA system is shown in Figure 1.

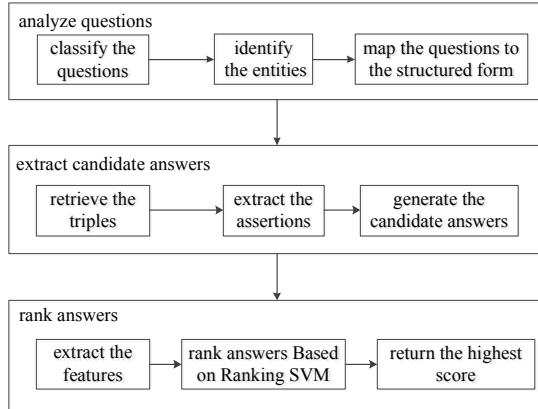


Figure 1. The process of KBQA system.

A. Features Extraction

In this paper, the four features that extracted to rank the candidate answers are the predicate sequence similarity based on sememe vector, predicates’ edit distances, predicates’ word co-occurrences and

classification.

1) Predicate sequence similarity based on sememe vector.

The words in the candidate answers’ predicates mostly have fixed collocations, such as “出品 公司(Produce Company)”, “效力 球队(Serve Team)”, etc. When these words are used in sentences, their positions are usually adjacent, such as “《卧虎藏龙》是由哪个公司出品的? (Which company produced ‘Crouching Tiger, Hidden Dragon’?)”, “李明曾经效力于什么球队? (What team did Li Ming ever serve for?)”. Therefore, after removing the stop words in the questions, these words are usually in adjacent positions. When calculating the similarities between the questions’ predicates and the candidate answers’ predicates, this paper uses the local contiguous subsequences of questions’ predicates and the sequences of candidate answers’ predicates to calculate the similarity. The word similarity calculation is based on the method of sememe vector described in section III.

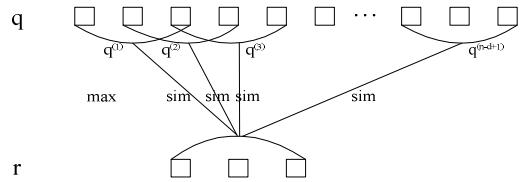


Figure 2. The model of predicate similarity calculation.

In this paper, the set of predicates’ word sequences in questions is called q . The number of words in q is n . The set of predicates’ word sequences in the candidate answers is called r . The number of words in r is d . $q^{(i)}$ is the local word sequence in q , it has a continuous length d and starts with the i -th word. The model of predicate similarity calculation is shown in Figure 2. We can calculate the similarity between all local word sequences and r , and then we take the maximum value as the predicate sequence similarity of q and r , such as the formula (9).

$$\text{sim}(q, r) = \max_{i=1, \dots, n-d+1} (\text{sim}(q^{(i)}, r)) \quad (9)$$

When calculating the similarity between the local word sequences $q^{(i)}$ and r , the similarity of each word in $q^{(i)}$ and all words in r is calculated separately, then the maximum values are taken. The sum of the maximum values is averaged as the similarity of $q^{(i)}$ and r , such as the formula(10).

$$\text{sim}(q^{(i)}, r) = \frac{\sum_{j=1}^d \max_m \text{sim}(q_j^{(i)}, r_m)}{d} \quad (10)$$

Where $q_j^{(i)}$ is the j -th word in the subsequence $q^{(i)}$, r_m is the m -th word in r , $\max_m \text{sim}(q_j^{(i)}, r_m)$ is the maximum value of all words’ similarities in $q_j^{(i)}$ and r , and d is the number of words in $q^{(i)}$.

When the similarity is calculated based on the local subsequences, the length of the questions’ predicate q may be smaller than the length of the candidate answers’ predicate r . At this time, the similarity of the entire q and r is directly calculated, such as the formula (11).

$$\text{sim}(q, r) = \frac{\sum_{k=1}^n \max_m \text{sim}(q_k, r_m)}{n} \quad (11)$$

Where q_k is the k-th word in q , r_m is the m-th word in r , $\max_m \text{sim}(q_k, r_m)$ is the maximum value of all words' similarities in q_k and r , and n is the number of words in q .

2) Predicates' edit distances.

By analyzing the questions' predicates and the candidate answers' predicates, the smaller the edit distances between them, the smaller the differences between the predicates. Therefore, the edit distances can also be considered as a feature of the candidate answers ranking. The edit distances represent the minimum number of times that a string needs to be edited to another string. This process includes insertion, deletion, replacement and other operations. The edit distance is calculated as follows:

For the two strings of “发行公司 (Publishing Company)” and “发行商(Publisher)”, we can align the string “发行(Publish)”. Then the string “公(Gong)” is replaced with the string “商(Shang)” and the string “司(Si)” is deleted. After the above operations, the “发行公司 (Publishing Company)” is converted into “发行商(Publisher)”. Therefore, the edit distance between them is 2.

In this paper, the reciprocal of the editing distance d is selected as a ranking feature, such as the formula (12).

$$d' = \begin{cases} \frac{1}{d}, & \text{若 } d \neq 0 \\ 1, & \text{若 } d=0 \end{cases} \quad (12)$$

3) Predicates' word co-occurrences.

Based on statistical thoughts, the co-occurrences of words in questions and answers are also an important factor in the answers matching. If the words in the questions and the candidate answers often co-occur in the question-answer pairs, the probability that the candidate answers are the correct answers is higher.

In the training data, we need to count the number of co-occurrences of each word in the questions' predicates and the candidate answers' predicates. If the co-occurrences are higher, it means that when some words appear in the questions, the corresponding words tend to appear in the candidate answers' predicates. For example: “时候(moment)” and “时间(time)”, “多少(how many)” and “数(number)”.

Suppose the words sequence of the questions' predicates is q and the words sequence of the candidate answers' predicates is r , l_1 and l_2 are the number of words they contained respectively, then the co-occurrences feature of the two predicates is shown as formula (13).

$$f = \frac{\sum_{i=1}^{l_1} \sum_{j=1}^{l_2} co(q_i, r_j)}{l_1 \cdot co_{\max}} \quad (13)$$

Where $co(q_i, r_j)$ is the number of co-occurrences of the words q_i and r_j in the questions and candidate answers

of the training data, and co_{\max} is the maximum number of co-occurrences of all words.

4) Classification feature.

The feature of classification refers to the probability of consistency between the category of each candidate answer and question. Classifying the questions can narrow the search range of the candidate answers. At the same time, it can improve the efficiency and accuracy of returning answers. If the types of candidate answers and the questions are consistent, the probability of being the correct answers is higher.

This paper respectively trains two Maximum Entropy (ME) models for questions and candidate answers to obtain the types of questions and candidate answers. The classification categories include “description, person, place, number, institution name, entity, time and others”. If a question description is “where is XXX?”, the categories of this question and answer are all “place category”. Similarly, if a question description is “when is XXX?”, the categories of this question and answer are all “time category”. The process of obtaining candidate answers' classification features is shown as Figure 3.

Suppose there are m questions and answers. Where a_i is the i-th candidate answer; q_j is the j-th question; C is the most probable category for the j-th question. $P(a_i|C_k)$ is the probability of the candidate answer a_i under each category C_k ($k=1,2,\dots,m$). $P(a_i|C)$ is the probability of the candidate answer a_i under the category C .

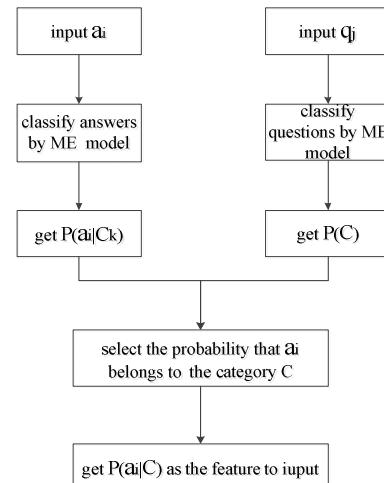


Figure 3. The acquisition process of candidate answers' classification features.

As is shown in Figure 3, the ME model of questions' classification is used to get the category C . The ME model of answers' classification is used to get the probability $P(a_i|C_k)$. Finally, we select the set of $P(a_i|C)$ as the classification features of the candidate answers.

B. Ranking Answers Based on Ranking SVM

The Ranking SVM³ algorithm is a ranking algorithm proposed by Herbrich et al. The four types of features extracted above in this paper are used as inputs of the Ranking SVM. The training data is used to train the model parameters, and the ranking learning model is used to give

³http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

each candidate answers a score, then we can rank answers according to the scores.

V. EXPERIMENTS AND RESULTS ANALYSIS

A. Dataset

This paper selects the KBQA's evaluation data in NLPCC 2016 as the experimental data. It contains 14,609 QA pairs as training data and 9,870 questions as test data. It also contains a structured knowledge base with a total of 47,943,429 “entity-predicate-entity” triples, and there are 8,721,640 entities.

B. Experimental Data Preprocessing

By observing the training data, phrases such as “我想知道(I want to know)”, “我很好奇(I'm curious)”, “谁能告诉我(Who can tell me)”, “有谁知道(Who knows)”, “什么是(What is)” appear in most questions, the original meaning of the questions is not affected after these phrases are removed. Therefore, data preprocessing needs to remove these phrases from questions.

Entity recognition is the primary task of question analysis, so we need to find out the entities contained in the questions. This paper uses pattern matching to identify the entities in the questions according to the structural features of knowledge base and the syntactic features of simple questions. After identifying the entities, the questions' remaining words are used to perform words segmentation and stop words removal. Then the questions can be mapped to the form of “entity-predicate”.

Here is an example, for the question “龙泉镇在中国的哪个地方?(Where is Longquan Town in China?)”, the “龙泉镇(Longquan Town)” and “中国(China)” are entities that successfully match the entities sets. After identifying “龙泉镇(Longquan Town)”, the remaining words “在中国的哪个地方? (Where is it in China?)” are used to perform words segmentation and stop words removal. Therefore, the question can be mapped to the structured form of “龙泉镇(Longquan Town)— 中国(China) 地方(Place)”. In the same way, after identifying “中国(China)”, the question can be mapped to the form of “中国(China)— 龙泉镇(Longquan Town) 地方(Place)”.

After mapping the questions to the form of “entity-predicate”, the entities are searched in the knowledge base. Then the corresponding assertions of the entities are extracted as the candidate answers of the questions.

This paper uses the pattern matching to identify the entities in the questions. The candidate answer triples are extracted from the knowledge base according to the entities.

C. Evaluation Indicators

In this paper, the average accuracy rate (Precision), average recall rate (Recall), average MRR and average F1 value are used as evaluation indicators (formulas 14-18).

$$\text{Precision} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{\#(C_i, A_i)}{|C_i|} \times 100\% \quad (14)$$

$$\text{Recall} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{\#(C_i, A_i)}{|A_i|} \times 100\% \quad (15)$$

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i} \quad (16)$$

$$\text{AveragedF1} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} F_i \times 100\% \quad (17)$$

For the i-th question, C_i represents the set of answers generated by the experimental system, $|C_i|$ is the number of answers in C_i , A_i is the standard answer set, $|A_i|$ is the number of answers in A_i , $\#(C_i, A_i)$ is the number of the same answers in C_i and A_i , $|Q|$ is the total number of test questions, rank_i is the position of the correct answer in the candidate answers set C_i of the i-th question. If C_i does not contain the correct answer, the value of MRR is 0.

$$F_i = \frac{2 \cdot \frac{\#(C_i, A_i)}{|C_i|} \cdot \frac{\#(C_i, A_i)}{|A_i|}}{\frac{\#(C_i, A_i)}{|C_i|} + \frac{\#(C_i, A_i)}{|A_i|}} \quad (18)$$

F_i is the F1 value of the i-th question.

D. Experimental Design

HowNet is a finite semantic dictionary and there are OOV words. We used the following methods to calculate the word similarity.

1) The method based on sememe vector.

For words that do not exist in HowNet, the word similarity is defined as 0; otherwise, the calculation of word similarity is described in section III.

2) The method based on word2vec.

The word2vec⁴ is used as word vector model. The corpus is the People's Daily standard corpus in 1998. According to the experiment, the context window is 5, the word vector dimension is 200, and the word similarity is calculated by the cosine distances. For words do not exist in the corpus, the similarity is defined as 0.

3) The method based on the combination of sememe vector and word2vec.

For the words that exist in HowNet, we use the sememe vector to calculate the word similarity. Otherwise, we use the word2vec to calculate. For words not exist in HowNet and the corpus, the similarity is defined as 0.

This paper conducts two groups of experiments to verify the effect of word similarity. The experiments in the first group use the feature of predicate sequence similarity to rank the candidate answers, which is called the KBQA based on single feature. The experiments in the second group combine the Ranking SVM algorithm with the features of predicate sequence similarity, predicates' edit distances, predicates' word co-occurrences and classification to rank the candidate answers, which is called the KBQA based on multi-features.

E. Experimental Results and Analysis

1) Results and Analysis of KBQA Based on Single Feature.

In this group of experiments, only the feature of predicate sequence similarity is used to rank the candidate answers. The candidate answers with the highest ranking

⁴ <http://word2vec.googlecode.com/svn/trunk/>

are returned. This paper uses three methods to calculate the predicate sequence similarity. The process is described in section V.D. The results of these three methods applied in the KBQA are shown in TABLE II.

TABLE II. The experimental results based on the feature of predicate sequence similarity

Method	AveF1	Precision	Recall	MRR
sememe vector	71.77%	69.64%	78.80%	0.7311
word2vec	70.60%	68.70%	76.55%	0.7175
sememe vector +word2vec	71.81%	69.70%	78.83%	0.7314

As can be seen from Table II, only the predicate similarity is used to sort the candidate answers, the F1 values of the obtained results are all over 70%, and the recall rate is also more than 76%. The above results show that the word similarity calculation has a better effect on the answer retrieval of the KBQA. In addition, the word similarity calculated by the combination of the sememe vector and word2vec is the most effective, the method based on sememe vector is the middle, and the method based on word2vec is the lowest. Therefore, the KBQA experiment with the combination of the two methods is the best; the result is higher than the other methods in four evaluation indicators.

2) Results and Analysis of KBQA Based on Multi-features.

In this group of experiments, when ranking the candidate answers, we use the ranking learning algorithm Ranking SVM combined with predicate similarity, edit distances, word co-occurrences, and classification features to select the top answers. The three calculation methods of word similarity are applied to the KBQA experiment, and the results are shown in Table III.

TABLE III. The experimental results based on multi-features

Method	AveF1	Precision	Recall	MRR
sememe vector	75.88%	73.88%	82.29%	0.7749
word2vec	74.12%	72.35%	79.52%	0.7518
sememe vector +word2vec	75.91%	73.93%	82.32%	0.7723

As can be seen from Table III, the method based on the sememe vector is higher than the method based on the word2vec in the four evaluation indicators. The features of edit distances, word co-occurrences, and classification are the same. Only the feature of predicate sequence similarity has impact on evaluation indicators. The differences of the predicate sequence similarity feature are mainly due to the different methods to calculate the word similarity. It can be seen from the experimental results that the method based on the sememe vector has achieved good results in the application of the KBQA system.

In the QA system experiments, this paper only selects the candidate answers with the highest score to return. The recall rate based on the method of sememe vector is 82.29%, which shows that 8,123 questions get the correct answers among the 9,870 questions.

VI. CONCLUSION

This paper proposes a method of ranking candidate answers based on sememe vector. This method combines the features of predicates' word similarity in the the questions and the candidate answers, edit distances, word

co-occurrences and classification with the ranking SVM algorithm to rank the candidate answers of the questions and select the answers with the highest score.

The experimental results show that the features of the candidate answers ranking obtained in this paper have good results. The recall rate of word representation based on sememe vector is 82% in the KBQA system. This shows that the language knowledge base plays an important role in the word representation.

REFERENCES

- [1] Etzioni O. Search needs a shake-up. *Nature*, 2011, 476(7358): 25–26.
- [2] Lehmann J, Isele R, Jakob M, et al. DBpedia—a large-scale, multilingual knowledge base extracted from Wikipedia [J]. *Semantic Web*, 2015, 6(2): 167–195.
- [3] Bollacker K, Evans C, Paritosh P, Sturge T, Taylor J. Freebase:a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data.Vancouver, Canada: ACM, 2008. 1247–1250.
- [4] Suchanek F M, Kasneci G, Weikum G. Yago: a core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web. Alberta, Canada: ACM, 2007. 697–706.
- [5] Church K W. Word2Vec[J]. *Natural Language Engineering*, 2017, 23(1): 155–162.
- [6] Pennington J, Socher R, Manning C. Glove: Global vectors for word representation[C]//Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014: 1532–1543.
- [7] Liu K, Zhang Y Z, Guoliang J I, et al. Representation Learning for Question Answering over Knowledge Base: An Overview[J]. *Acta Automatica Sinica*, 2016.
- [8] Liang P, Jordan M I, Klein D. Learning dependency-based compositional semantics[J]. *Computational Linguistics*, 2013, 39(2): 389–446.
- [9] Berant J, Chou A, Frostig R, et al. Semantic parsing on freebase from question-answer pairs[C]//Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. 2013: 1533–1544.
- [10] Kwiatkowski T, Choi E, Artzi Y, et al. Scaling semantic parsers with on-the-fly ontology matching[C]//Proceedings of the 2013 conference on empirical methods in natural language processing. 2013: 1545–1556.
- [11] Yao X, Van Durme B. Information extraction over structured data: Question answering with freebase[C]//Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2014: 956–966.
- [12] Lai Y, Lin Y, Chen J, et al. Open domain question answering system based on knowledge base[M]//Natural Language Understanding and Intelligent Applications. Springer, Cham, 2016: 722–733.
- [13] Xu K, Reddy S, Feng Y, et al. Question answering on freebase via relation extraction and textual evidence[J]. arXiv preprint arXiv:1603.00957, 2016.
- [14] Page L, Brin S, Motwani R, et al. The PageRank Citation Ranking: Bringing Order to the Web[J]. Technical Report, Stanford Digital Libraries, 1998.
- [15] Li H, Mu L, Zan H. Computation of Word Similarity Based on the Information Content of Sememes and PageRank Algorithm[M]// Chinese Lexical Semantics. Springer International Publishing, 2016.
- [16] Liu Q, Li S. Vocabulary semantic similarity calculation based on HowNet[C]. Taipei: The 3rd Chinese Lexical Semantics Seminar. 2002: 59–76.

Improving Japanese-English Bilingual Mapping of Word Embeddings based on Language Specificity

Yuting Song*, Biligsaikhan Batjargal† and Akira Maeda‡

* Research Organization of Science and Technology, Ritsumeikan University, Kusatsu, Japan 525-8577

Email: ytsong@gst.ritsumei.ac.jp

† Kinugasa Research Organization, Ritsumeikan University, Kyoto, Japan 603-8577

Email: bilige@fc.ritsumei.ac.jp

‡ College of Information Science and Engineering, Ritsumeikan University, Kusatsu, Japan 525-8577

Email: amaeda@is.ritsumei.ac.jp

Abstract—Recently, cross-lingual word embeddings have attracted a lot of attention, because they can capture semantic meaning of words across languages, which can be applied to cross-lingual tasks. Most methods learn a single mapping (e.g., a linear mapping) to transform word embeddings space from one language to another. In this paper, we propose an advanced method for improving bilingual word embeddings by adding a language-specific mapping. We focus on learning Japanese-English bilingual word embedding mapping by considering the specificity of Japanese language. On a benchmark data set of Japanese-English bilingual lexicon induction, the proposed method achieved competitive performance compared to the method using a single mapping, with better results being found on original Japanese words.

Keywords-Cross-lingual word embeddings; Japanese; Word translation

I. INTRODUCTION

Distributed representations of words, so-called word embeddings [1], [2], [3], have achieved impressive results in many Natural Language Processing (NLP) tasks and applications [4], [5], [6]. While it is possible to obtain monolingual word embeddings for most languages, the monolingual word embeddings in different languages are not comparable, hence, cannot be used in cross-lingual NLP applications.

With the needs of representing words in cross-lingual settings, several models for learning cross-lingual word embeddings have been proposed [7], [8], [9], [10]. These models can be divided mainly into two types: “online” and “offline”. “Online” approaches learn word embeddings of two languages simultaneously by exploiting monolingual texts and some parallel corpora [11], [12]. “Offline” approaches firstly obtain the word embeddings in two languages independently, and then learn a mapping function (e.g., a linear mapping) from pre-trained monolingual word embeddings [7], [9], [13], [14], [15]. In this paper, we focus on this offline approach.

The early works of “offline” approaches learn bilingual word embedding mappings using some bilingual signals, such as bilingual lexicons [7], [9] and parallel or comparable data [8], [16], [17], as supervision to minimize distance between two word embedding spaces. These methods rely on bilingual lexicons or parallel corpora. To mitigate the needs of bilingual data, some recent methods focused on

an unsupervised way, which attempts to learn cross-lingual word embeddings by mapping word embedding spaces to each other based on the distribution information of word embeddings [14], [18] or generative adversarial networks [19].

While both of supervised and unsupervised methods have shown impressive results, the shortcoming of these methods is that they have an assumption that entire word embeddings in one language are mapped to other language by following a same mapping function (e.g., a linear mapping). In this paper, we introduce a method to relax the above assumption by adding a language-specific mapping. Our method aims at learning more precise mapping across languages by considering the specificity of languages. Specifically, we focus on learning Japanese-English bilingual word embedding mapping by considering the specificity of Japanese.

The remainder of this paper is organized as follows. Section II introduces the previous work of learning bilingual word embedding mappings. Section III explains the proposed method. Section IV and Section V present the experimental settings and results analysis. Section VI concludes this paper and outlines future work.

II. BILINGUAL WORD EMBEDDING MAPPING

In this section, we first present notations that are used in this paper. Then, we introduce the previous methods of learning bilingual word embedding mapping using bilingual lexicons, which is the basis of our method for improving Japanese-English bilingual word embedding mapping.

A. Notation

Given pre-trained monolingual word embeddings of two languages, the goal of bilingual word embedding mapping is to learn a mapping that aligns the two pre-trained monolingual word embeddings. Let $x \in \mathbb{R}^d$, $y \in \mathbb{R}^d$ denote word vectors in the source and target language embedding spaces respectively. Let x_i and y_i denote word vectors of an actual word in the source and target language vocabularies respectively.

B. Linear mapping

Learning a linear mapping between two monolingual word embedding spaces was first introduced in [7]. They

use a bilingual dictionary (usually most frequent words) of $n = 5000$ pairs of words with their associated vectors $\{x_i, y_i\}_{i \in \{1, n\}}$ to learn a linear mapping W between two monolingual word embeddings spaces by employing stochastic gradient descent to minimize the squared reconstruction error:

$$\min_W \sum_{i=1}^n \|y_i - Wx_i\|^2 \quad (1)$$

With the learned mapping W , any word embedding in the source language can be mapped to the target language embedding space by $y = Wx$.

Based on the linear mapping model that is introduced above, [9], [20] demonstrated that quality of bilingual word embeddings is improved by adding an orthogonality constraint on W in (1). Our method learns bilingual word embedding mapping based on the method in [9].

III. THE PROPOSED METHOD

In Section II, we introduced previous methods of learning a linear mapping between two language embedding spaces, which are based on an assumption that all word embeddings in one language follow the same mapping function. This single mapping can be seen as a global mapping, which is obviously a simplified method. While the global mapping performs good on the words that represent common concepts across languages (e.g., 犬 - dog), we noticed that the performance drops off for language-specific words (reported in Table II), such as some original Japanese words that their corresponding English words are their romanizations (e.g., 石田, a Japanese surname, its corresponding word in English is its romanization “ishida”). This shows the limitation of mapping all the word embeddings using a single global mapping, since global mapping does not consider the language specificity.

To solve the above problem, we propose an advanced method, in which we learn a language-specific mapping and combine it with global mapping. We believe that the combination of global mapping and language-specific mapping should achieve better performance.

In our method, we first learn a global mapping based on the method in [9], which is introduced in Section II. Then, we use the learned global mapping to extract a set of original Japanese words and their romanizations pairs (e.g., 石田 - ishida), which are used as a bilingual dictionary to learn a language-specific mapping. Finally, we transform word embeddings from one language to another by employing the combination of global mapping and language-specific mapping.

A. Global mapping

To learn the global mapping W_g between Japanese and English word embedding spaces, we follow the method in [9], which is a supervised method that needs some bilingual lexicons. While recent unsupervised methods achieve competitive or superior performance for some language pairs compared to supervised methods, their performance degrades for distant language pairs, such as

Japanese-English, which was experimentally proved by [13], [15]. Thus, we choose a supervised method and use a set of Japanese words and their English translations as a bilingual dictionary to learn a global linear mapping between Japanese and English word embedding spaces.

B. Language-specific mapping

As introduced at the beginning of Section III, we are going to learn a language-specific mapping by employing language-specific words. We aim to utilize Japanese language-specific words, which are original Japanese words whose corresponding English words are their romanizations.

We extract original Japanese words in the following steps:

1) *Identify original Japanese words candidates*: All the Japanese words can be romanized and represented in Latin scripts. However, only the original Japanese words’ corresponding English words should be their romanizations. Thus, we first roughly identify original Japanese words candidates by checking whether their romanizations appear in English vocabulary.

2) *Identify Japanese origins of transliterated words in English*: For some original Japanese words candidates that are obtained in the previous step, even their romanizations appear in English vocabulary, but their true corresponding English words might be not their romanizations. In order to filter out these words, we firstly utilize the global mapping to map all the candidates of original Japanese words to an English embedding space. Then, the mapped Japanese word vectors are compared with all the English word vectors to find the closest words in English. For a given Japanese word, if most of its closest words in English are transliterated words from Japanese words, it has more possibility to be an original Japanese word. Based on this observation, for each candidate of original Japanese word, we find its 5 closest words in English. Then, we identify transliterated words in these 5 closest English words by using the method in [21], which distinguishes a transliterated word from English words (non-transliterated words) by judging whether it can be segmented by Japanese syllabaries. In the experiments, for a given original Japanese word candidate, we empirically determine it as an original Japanese word if at least 3 of its 5 closest English words are transliterated words.

After this step, we can obtain a set of original Japanese words. We use the romanizations of these original Japanese words as their corresponding English words. In this way, we extract a set of pairs of original Japanese words and their romanizations, which are used as a bilingual dictionary to learn a language-specific mapping W_s based on the method in [9] that is introduced in Section II.

C. Combination of global mapping and language-specific mapping

For a given Japanese word, its word vector x is mapped to English word embedding space by using both global

mapping W_g and language-specific mapping W_s . When comparing x with a English word vector y , it obtains two similarity scores, which are represented as $S_{sim}(W_gx, y)$ and $S_{sim}(W_sx, y)$. The details of similarity calculation metrics we used are introduced in the implementation details in Section IV.

To combine the two similarity scores, we propose to either select the maximum score as the final similarity degree, or the weighted summation of two similarity scores:

$$S_{sim}(x, y) = \max\{S_{sim}(W_gx, y), S_{sim}(W_sx, y)\} \quad (2)$$

$$S_{sim}(x, y) = \alpha \cdot S_{sim}(W_gx, y) + (1 - \alpha) \cdot S_{sim}(W_sx, y) \quad (3)$$

α ($\alpha \in [0, 1]$) is the weight that are used to balance the two similarity scores.

IV. EXPERIMENTS

In this section, we experimentally evaluate our method in bilingual lexicon induction task, which measures the word translation accuracy in comparison to a gold standard.

A. Experimental dataset

We evaluated our method on the widely used MUSE dataset [19], which consists of dictionaries for many language pairs divided into training and test sets. We evaluated our method by inducing lexicons between Japanese and English. The training data set is composed of 5000 Japanese words with their English translations. The test data set is composed of 1500 Japanese words with their English translations.

The monolingual word embeddings were trained using *word2vec* with the skip-gram model [2]. The Japanese word embeddings were trained on Japanese Wikipedia corpus, and the English word embeddings were trained on English Wikipedia corpus.

B. Implementation details

1) *Embedding normalization*: Reference [9] proved that pre-processing of monolingual word embeddings with length normalization and dimension-wise mean centering can improve the performance of liner bilingual word embedding mapping. In our experiments, we follow the recommended settings in [9] to pre-process the monolingual word embeddings by applying firstly length normalization, then dimension-wise mean centering, and then length normalization again to ensure that the final embeddings have a unit length.

2) *Retrieval method in bilingual lexicon induction*: In bilingual lexicon induction task, given the words in the source language, it needs to have a retrieval metric that is used to select corresponding translations. Reference [22] demonstrated that nearest neighbors retrieval suffers from the hubness problem, which is that a few words (known as hubs) dominates as becoming nearest neighbors over many

other words. Among the existing solutions to mitigate hubness [10], [19], we utilize the Cross-domain Similarity Local Scaling (CSLS) [19].

To measure the similarity between a mapped source word vector Wx and a word vector in the target language y , the CSLS considers the average similarity of Wx and y for their k nearest neighbors in another language, respectively. The CSLS similarity measure is defined as:

$$\text{CSLS}(Wx, y) = 2\cos(Wx, y)$$

$$- \frac{1}{k} \sum_{y' \in \mathcal{N}_T(Wx)} \cos(Wx, y') - \frac{1}{k} \sum_{Wx' \in \mathcal{N}_S(y)} \cos(Wx', y) \quad (4)$$

where $\mathcal{N}_T(Wx)$ is the set of k nearest neighbors of Wx from the target language. Similarly, $\mathcal{N}_S(y)$ is the set of k nearest neighbors of y . Following the parameter settings in [19], we set $k = 10$ in our experiments.

C. Baseline methods

We compare the proposed method with following baselines:

1) *Global mapping (GM)*: This type of methods aims at transforming words in one language vector space to another by using a single global mapping. We use GM as a baseline to verify the effectiveness of our method, which combines a language-specific mapping with GM.

2) *Language-specific mapping (LM)*: This method only use language-specific mapping to map Japanese word embedding to English. We compare our method with LM in order to examine the effect of only considering language specificity.

D. Our method

As introduced in the last subsection of Section III, our method combines similarity scores (CM) from global mapping and language-specific mapping in two ways: one (i.e., (2)) is to select the maximum score, which is represented as CM_{max} . The other (i.e., (3)) utilizes the weighted summation of two similarity scores, which is represented as CM_{sum} .

V. RESULTS AND ANALYSIS

A. Evaluation metrics

We evaluate the experimental results by using precision @1, @3, and @5. The precisions are equal to the rates of Japanese words whose correct corresponding English words are found in the top 1, 3, 5 results, respectively.

B. Overall experimental results

Table I shows the experimental results. For CM_{sum} , we show the results when $\alpha = 0.9$ as it has better performance than other weights. In addition, we report some sample results from the experiments of GM and CM_{sum} ($\alpha = 0.9$) in Table III.

From Table I, we can that the overall performance of CM_{max} and CM_{sum} are on par with GM, which is the most competitive baseline. Comparing LM with other two methods, LM achieves poor performance. It indicates that

Table I
PRECISION@N FOR JAPANESE-ENGLISH BILINGUAL LEXICON INDUCTION TASK ON MUSE DATASET COMPARED WITH BASELINE METHODS

	P@1 (%)	P@3 (%)	P@5 (%)
GM	36.38	48.08	52.58
LM	12.06	16.85	19.68
CM_{max}	36.75	48.37	52.87
$CM_{sum} (\alpha = 0.9)$	36.67	48.51	53.09

only using original Japanese words to learn a language-specific mapping is not enough for obtaining high-quality bilingual word embeddings.

C. Performance on original Japanese words

In order to evaluate the performance of our method on original Japanese words, we manually extract all the original Japanese words in the test data set. There are totally 499 word pairs that their Japanese words are original Japanese words.

Table II shows the experimental results over original Japanese words. $CM_{sum} (\alpha = 0.3)$ significantly outperforms the two baselines GM and LM. The statistical significance was confirmed in a two-sided t-test at a significance level of 0.1. Comparing the results of GM with other methods, the lowest performance is observed in GM over the original Japanese words, which indicates that using a single global mapping is a simplified method that cannot achieve good performance on language-specific words as discussed in the beginning of Section III.

D. Analysis of different weights for combination of GM and LM

In our method, when combining the similarity scores of global mapping and language-specific mapping, the weighted summation method CM_{sum} (i.e., (3)) needs to set the weight α to balance two similarity scores. To investigate how the weight α affects the performance of CM_{sum} , we conducted experiments on all test data set and original Japanese data set using different values of α .

Fig. 1 presents the results of using different weights to combine GM and LM. The higher values of α achieve better performance on the all test data set. However, the lower values of α obtain better performance on the data set of original Japanese words. This indicates that the impact of LM is more obvious on original Japanese words. In addition, this reveals the necessity of determining α based on data characteristics. For example, when data set contains more original Japanese words, the similarity score of LM should have a higher weight.

Table II

EXPERIMENTAL RESULTS ON THE DATA SET OF ORIGINAL JAPANESE WORDS

	P@1 (%)	P@3 (%)	P@5 (%)
GM	22.12	29.88	33.65
LM	29.18	36.94	40.71
CM_{max}	23.29	30.82	34.59
$CM_{sum} (\alpha = 0.3)$	34.12	44.00	48.24

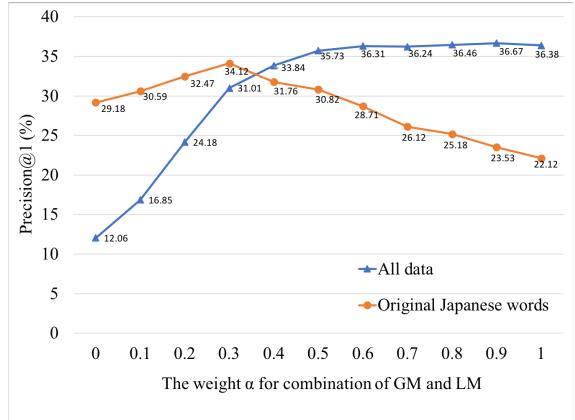


Figure 1. Experimental results of different weights for combining GM and LM

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a method to learn Japanese-English bilingual word embedding mapping. Our method exploits the specificity of Japanese to learn a language-specific mapping, which is combined with global mapping for improving Japanese-English bilingual word embedding mapping. The experimental results on the MUSE benchmark data set of bilingual lexicon induction task proved the effectiveness of the proposed method.

In the future, we will apply our method to other language pairs including Japanese and European languages. In addition, we plan to extend our method beyond the word level to longer text units such as phrases or sentences.

ACKNOWLEDGMENT

This work was supported in part by JSPS KAKENHI Grant Number JP16K00452, and MEXT-Supported Program for the Strategic Research Foundation at Private Universities (S1511026).

REFERENCES

- [1] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching word vectors with subword information,” *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.
- [2] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv preprint arXiv:1301.3781*, 2013.
- [3] J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global Vectors for Word Representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1532–1543.
- [4] Q. V. Le and T. Mikolov, “Distributed representations of sentences and documents,” in *Proceedings of the 31st International Conference on Machine Learning*, vol. 32, 2014, pp. 1188–1196.
- [5] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, “Neural architectures for named entity recognition,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 260–270.

Table III

SAMPLE RESULTS OF GM AND CM_{sum} ($\alpha = 0.9$) . “O” REPRESENTS ORIGINAL JAPANESE WORDS. BOLD TEXT INDICATES THE GROUND TRUTH, AND THE NUMBERS IN PARENTHESIS ARE THE RANKS OF THE GROUND TRUTH ENGLISH WORDS WITHIN THE RETURNED RESULTS.

Japanese word	Ranked English words returned by GM	Ranked English words returned by CM _{sum} ($\alpha = 0.9$)
知り合い	acquaintance (1), reacquainted, befriended, friend, colleague, acquaintances (6), friends, romantically	reacquainted, acquaintance (2), befriended, friend, colleague, acquaintances (6), romantically, acquainted
毎時	headways, hourly (2), half-hourly, hour, off-peak, kmph, round-trips, daytimes	headways, hourly (2), half-hourly, hour, off-peak, kmph, round-trips, daytimes
勝者	winner (1), losers, victors, winners (4), contenders, match-up, loser, loser's	winner (1), losers, victors, winners (4), match-up, contenders, loser, loser's
屋外	outdoor (1), open-air, outdoors (3), indoors, indoor, unheated, well-lit, floodlights	outdoor (1), outdoors (2), open-air, indoors, indoor, unheated, well-lit, tents
栄一 (O)	bronisław, tatsuo, isao, yasuo, kazuo, hellmut, andrzej, eiichi (8)	bronisław, isao, tatsuo, eiichi (4), kazuo, yasuo, ichirō, hellmut
彦根 (O)	kawagoe, karatsu, kumamoto, hikone (4), kurashiki, kakegawa, hirosaki, shizuoka	hikone (1), karatsu, kawagoe, kumamoto, kakegawa, hirosaki, kurashiki, shizuoka
岸田 (O)	yukio, naoto, masahiko, kishida (4), ichirō, fumio, shota, makiko	yukio, kishida (2), naoto, masahiko, ichirō, fumio, makiko, yasushi
熱海 (O)	atami (1), abashiri, otaru, karuizawa, hakone, kusatsu, enoshima, onsen	atami (1), abashiri, karuizawa, otaru, hakone, kusatsu, onsen, enoshima

- [6] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng, “Parsing with Compositional Vector Grammars,” in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, vol. 1, 2013, pp. 455–465.
- [7] T. Mikolov, Q. V. Le, and I. Sutskever, “Exploiting similarities among languages for machine translation,” *arXiv preprint arXiv:1309.4168v1*, 2013.
- [8] I. Vulić and M. F. Moens, “Bilingual distributed word representations from document-aligned comparable data,” *Journal of Artificial Intelligence Research*, vol. 55, pp. 953–994, 2016.
- [9] M. Artetxe, G. Labaka, and E. Agirre, “Learning principled bilingual mappings of word embeddings while preserving monolingual invariance,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 2289–2294.
- [10] S. L. Smith, D. H. P. Turban, S. Hamblin, and N. Y. Hammerla, “Offline bilingual word vectors, orthogonal transformations and the inverted softmax,” in *Proceedings of the 5th International Conference for Learning Representations (ICLR2017)*, 2017.
- [11] S. C. AP, S. Lauly, H. Larochelle, M. M. Khapra, B. Ravindran, V. Raykar, and A. Saha, “An autoencoder approach to learning bilingual word representations,” *Advances in Neural Information Processing Systems*, pp. 1853–1861, 2014.
- [12] K. M. Hermann and P. Blunsom, “Multilingual models for compositional distributed semantics,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014, pp. 58–68.
- [13] A. Joulin, P. Bojanowski, T. Mikolov, H. Jegou, and E. Grave, “Loss in translation: Learning bilingual word mapping with a retrieval criterion,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 2979–2984.
- [14] M. Artetxe, G. Labaka, and E. Agirre, “A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018, pp. 1–10.
- [15] C. Zhou, X. Ma, D. Wang, and G. Neubig, “Density matching for bilingual word embedding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 1588–1598.
- [16] M. T. Luong, H. Pham, and C. D. Manning, “Bilingual word representations with monolingual quality in mind,” in *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, 2015, pp. 151–159.
- [17] J. Guo, W. Che, D. Yarowsky, H. Wang, and T. Liu, “Cross-lingual dependency parsing based on distributed representations,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 2015, pp. 1234–1244.
- [18] H. Cao, T. Zhao, S. Zhang, and Y. Meng, “A distribution-based model to learn bilingual word embeddings,” in *Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016)*, 2016, pp. 1818–1827.
- [19] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jégou, “Word translation without parallel data,” in *Proceedings of the Sixth International Conference on Learning Representations (ICLR 2018)*, 2018.
- [20] C. Xing, D. Wang, C. Liu, and Y. Lin, “Normalized word embedding and orthogonal transform for bilingual word translation,” in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015, pp. 1006–1011.
- [21] Y. Song, B. Batjargal, and A. Maeda, “Recognition and transliteration of proper nouns in cross-language record linkage by constructing transliterated word pairs,” *International Journal of Asian Language Processing*, vol. 27, no. 2, pp. 111–125, 2017.
- [22] G. Dinu, A. Lazaridou, and M. Baroni, “Improving zero-shot learning by mitigating the hubness problem,” *arXiv:1412.6568*, pp. 1–10, 2015.

The Initial Research of Mongolian Literary Corpus-Take the Text of Da.Nachugdorji's Work for Instance

YinhuaHai

School of Mongolian Studies

Inner Mongolia University

Hohhot, China

haiyh2008@163.com

Abstract—Today, the Mongolian corpus is gradually developed from the basic resource construction stage to an in-depth research covering multi-level processing or author-corpus-based quantitative analysis, and multi-functional electronic dictionary's development. However, there are still many shortcomings and deficiencies in the collection, development and processing of literary corpus. In this paper, the author will introduce the corpus of Da.Nachugdorji's Literature and will discuss its profound significance, and fulfill multi-level processing such as lexical, syntactic and semantic annotation, as well as dissertates the preliminary processing research of Mongolian literary corpus from the perspective of statistics on the POS, word and phrase frequency and computation of lexical richness.

Keywords-Mongolian literary corpus; processing; statistical research

I. LITERARY CORPUS AND THE DEVELOPING SIGNIFICANCE

As everyone knows, Da.Nachugdorji is one of the founders of Mongolian contemporary literature and a famous writer and poet. He began his literary creations in the 1920s and produced many poems, essays, comedy works and novels in his short life. His exquisite writing skills, unique language descriptions, rich and refined vocabularies provide readers with an aesthetic enjoyment for reading and appreciating his literary works, affecting a large number of Mongolian literary researchers and enthusiasts throughout the history (mainly from 1920s to 1940s). As it's commented, "the greatest contribution of his literary works is that he created the classic framework of the Mongolian contemporary literature. 'Classic' means a framework that can profoundly affect the hearts of Mongolian people" [1]. Therefore, the development of his work corpus is not only an integral part of the construction of all Mongolian literary corpus, but also the best way to scientifically protect and efficiently make use of his works, and provide valuable resources for many research fields in Mongolian studies.

The development of the corpus of "Da.Nachugdorji's Work" began in January 2016. In fact, it is a balanced corpus of two different texts of traditional Mongolian and Cyrillic Mongolian. The former has been completed; its Cyrillic version is in the developing stage. In terms of source of corpora, as a raw material, "Da.Nachugdorji" [Mongol] (Inner Mongolian People's Publishing House, the first printing in April 1981, a total of 436 pages) was used

to save 106 works for developing the corpora, including (the style, number of works and the proportion of corpus):

- (1) Political and lyric poems: 31, 19%;
- (2) Relevant health poems: 40, 14%;
- (3) Comedy: 1, 19%;
- (4) Fiction prose: 29, 46.7%;
- (5) Translated poems: 5, 1.3%.

For the related information of above-mentioned works, this paper uses Access database2016 to save that information such as the style, catalog and writing time of the works, as shown in Figure 1. The corpus of "Da.Nachugdorji's Work" was completely accomplished manually. The Mongolian text (Microsoft Word) was saved in the notepad++2016 format, and transformed into Latin text format with "Mongolian Proofreading System Editor4.0"¹, and finally achieve two different formats of preservation (Mongolian and Latin). The scale of corpus has reached about 628,000 words at present, whose sample is shown in Figure 2. For the text type, there are three formats, including the traditional Mongolian, Latin and Cyrillic, among which, research group for the Cyrillic texts is cooperating with the Institute of Language and Literature of Mongolian Academy of Sciences.

In fact, although the original intention of constructing a large-scale corpus was mainly to study vocabulary. However, a large number of literary works that have been included in the corpus have invisibly formed a rich field of literary appreciation and literary criticism. Corpus Stylistics is an interdisciplinary subject that combines literary text analysis with literary criticism using corpus linguistics [2]. Chinese scholar Guanghui Ma dissertated that "the method of corpus in linguistics can provide a set of effective methods and tools for literary research, so that people can make a more detailed, deeper and more specific description on literary works" [3]. "The stylistic style of a literary work is of unique charm of the author in the exchange of words, and it is the reflection of the author's creative intentions between the lines" [4]. Therefore, this paper believes that the research of stylistic style must be originated from the text of the works, which refers to the using and expression of language. The corpus had cuts into the research field of literary appreciation and literary criticism from such perspective. Many scholars at home and abroad have made some successful experiments in this field. A variety of retrieval software has been used in corpus stylistics, including WordSmith Tool, AntConc, Readability Analyzer and other open source tools, to have quantitative and qualitative analysis of literary texts from the perspectives of

This research is sponsored by the Project of NSSF of China (18BYY193), the project of Mongolian Language and Character of Inner Mongolia (MW-YB-201701).

word frequency, key words, concordance, collocation, clusters, article plot and so on. At present, the retrieval of novels and translation works has become a hot topic especially. Take Xueqin Cao's *A Dream of Red Mansions* for example, in recent years, the study based on its corpus is very popular in studies of literature and translation. In the database of "CNKI" (www.cnki.net), over hundreds of articles can be retrieved with the topic of "Research on the corpus-based *A Dream of Red Mansions*", which can show that the corpus has an undoubtedly great potential for research of literature, translation, and foreign language teaching.

ID	体裁1	目录1	著作时	体裁2	目录2
1	小说	《红楼梦》	1923	小说	《红楼梦》
2	小说	《红楼梦》	1925	小说	《红楼梦》
3	小说	《红楼梦》	1926	小说	《红楼梦》
4	小说	《红楼梦》	1927	小说	《红楼梦》
5	小说	《红楼梦》	1930	小说	《红楼梦》
6	小说	《红楼梦》	1930	小说	《红楼梦》
7	小说	《红楼梦》	1930	小说	《红楼梦》
8	小说	《红楼梦》	1931	小说	《红楼梦》
9	小说	《红楼梦》	1931	小说	《红楼梦》
10	小说	《红楼梦》	1935	小说	《红楼梦》
11	小说	《红楼梦》	1930	小说	《红楼梦》
12	小说	《红楼梦》	1930	小说	《红楼梦》
13	小说	《红楼梦》	1930	小说	《红楼梦》
14	小说	《红楼梦》	1930	小说	《红楼梦》
15	小说	《红楼梦》	1931	小说	《红楼梦》
16	小说	《红楼梦》	1931	小说	《红楼梦》
17	小说	《红楼梦》	1931	小说	《红楼梦》
18	小说	《红楼梦》	1931	小说	《红楼梦》
19	小说	《红楼梦》	1931	小说	《红楼梦》
20	小说	《红楼梦》	1932	小说	《红楼梦》
21	小说	《红楼梦》	1933	小说	《红楼梦》
22	小说	《红楼梦》	1934	小说	《红楼梦》
23	小说	《红楼梦》	1935	小说	《红楼梦》
24	小说	《红楼梦》	1935	小说	《红楼梦》

Figure 1. Sample of the statistical database of Da.Nachugdorji's Work

达那噶尔作品集.txt	
3217	达那噶尔作品集
3218	达那噶尔作品集
3219	达那噶尔作品集
3220	达那噶尔作品集
3221	达那噶尔作品集
3222	达那噶尔作品集
3223	达那噶尔作品集
3224	达那噶尔作品集
3225	达那噶尔作品集
3226	达那噶尔作品集
3227	达那噶尔作品集
3228	达那噶尔作品集
3229	达那噶尔作品集
3230	达那噶尔作品集
3231	达那噶尔作品集
3232	达那噶尔作品集
3233	达那噶尔作品集
3234	达那噶尔作品集
3235	达那噶尔作品集
1930年	

Figure 2. Sample of the corpus of " Da.Nachugdorji's Work" (Microsoft Mongolian version)

It is believed that corpus stylistics has become an effective supplement to traditional stylistics and one of the indicators of organic integration with modern information processing technology. Like scholar Wikberg of Swedish University of Oslo (1997: 312-325), corpus-based text analysis can complement with traditional text analysis. How to make full use of the corpus, word index in Mongolian language to serve literary studies, foreign language teaching and learning (such as data-driven learning DDL) is lack of necessary development and deeper annotation. Since 1983, a number of Mongolian corpora including *Mongolian Secret History* (Chinese-Mongol version in 1983), HODOM, TOD, DURBELJIN Mongol etc. has been developed in China. There has been accumulation in corpora resources such as monolingual corpus including "*One million words corpus of modern Mongolian*" (that is simply referred to as "*One Million*"), "*10 million words corpus of Mongolian*"; some bilingual or multilingual parallel corpora such as Mongolian-Chinese, Chinese-Mongolian, traditional Mongolian—Cyrillic Mongolian, Chinese-Mongolian-English corpus (its scale is usually calculated as "sentence pairs"); as well as the spoken

corpus. However, due to the slow process of basic resources of research and development, application and sharing, immature network technology and other relevant causes, there was a lag in resources construction progress and low efficiency in its utilization. It is not even expanded to more users for its applicability and practical effect through balanced large-scale development or finishing. It is still necessary to make up for the deficiencies. For the construction type of corpus, the above-mentioned Mongolian monolingual corpus belongs to the heterogeneous² type, which cannot be collected and preserved with the premise of predetermining relevant principles and proportions, resulting in the lack of balance and systematicness of corpus. The results of past research have shown that the types of *One million words corpus of modern Mongolian* include novels, languages, newspapers, and politics, which respectively account for 19.6%, 50.3%, 9.8% and 22.9% of the total corpus [5]. Based on the occupation ratio, it can be known that the novel corpus only accounts for 19.6%. Compared with news, politics, and newspapers and periodicals, the proportion of literary corpus is relatively low. Moreover, the "*One Million*" only contains two articles such as "Xagučin Xüü" and "Sibagun Saral" by Da.Nachugdorji. Therefore, in view of the fact that the "*10 million words corpus of Mongolian*" has not yet been applied, there should be more focus on the development and processing of literary corpus that can fully reflect the richness and diversity of Mongolian vocabulary. This will fill the gaps in literary corpus and can solve some unbalanced problems on the development of Mongolian corpus.

Besides, previous studies have developed special literary corpora such as "HOHE SVDVR", "NIGEN DABHVR ASAR" and "TVNGGALAG TAMIR" for specific applicative needs, with some research from the perspectives of corpus annotation processing, application and dictionary compilation (the basic information about the subject of paper, the name and scale of corpus are shown in Table I). One of the common features of it is the development of applicative software or lexicographic software to provide a convenient tool for users, and the modern corpus linguistics method used has played a role of "taking the lead before and after" on the development and research of Mongolian literary corpus. Drawing on and referring to the experience of above-mentioned special corpus, the research group started to develop the corpus of *Da.Nachugdorji's Work*.

TABLE I. STATISTICS OF CORPUS INFORMATION IN PREVIOUS RESEARCH

subject of paper	name of corpus	scale(ten thousand)
Construction of common noun's section in the electronic dictionary of khokh svdar[6]	khokh svdar	44.3
Construction of function word's section in the electronic dictionary of khokh svdar[7]		
Establishing a corpus of Mongolia-based on Cyrillic Mongolian material[8]	TVNGGALAG TAMIR	16.8
Processing and application of corpus of NIGEN DABHVR ASAR[9]	NIGEN DABHVR ASAR	12.4

II. THE PROCESSING AND STATISTICAL ANALYSIS OF THE LITERARY CORPUS

Constructing Mongolian multi-level tagged corpus through lexical-grammatical-semantic-pragmatic level

annotation and processing of the text on *Da.Nachugdorji's Work*, provide knowledge support for the statistics on word frequency, POS distribution, computation on lexical richness, the context and its adjacent collocation research. Since the mid-1990s, the main feature of the development of corpus linguistics has been the widespread application of corpora at all levels of linguistics. New Zealand's famous corpus researcher G. Kennedy summarized contemporary corpus research into different levels of vocabulary, syntax, text structure and text type [10]. In fact, the corpus of *Da.Nachugdorji's Work* was originally a raw corpus of Mongolian language knowledge carried by computer. This corpus, like G.Kennedy classification, needs to be tagged, analyzed and processed through lexical, syntactical, semantic and other aspects to become a useful knowledge resource-processed corpus. The significance of this research at each level is as follows.

A. Processing of the Corpus

For corpus processing, lexical, syntactical and semantic level, based on the corpus of *Da.Nachugdorji's Work*, can complete tagging the POS, phrase structure, sentence structure and sentence component as well as semantic classification and semantic role information annotation. In this paper, Da.Nachugdorji's representative work *Xagučin Xüü* is discussed as an example to disseminate relevant contents. For the POS tagging, the word segmentation, lexical variation labeling and POS tagging of the text can be completed with the Mglex system developed by the research group.

1) The text is a lexically tagged corpus including the word segmentation, additional components, and POS, for example: HODEGE(NoB21)-YIN BAYIDAL(NuB) \uij \$ALTAR(Aboe2) B0LTAR(Aboe2) ./Wp1 CAG(Ne1)-VN(Fc11) VLARIL(Ne2) 0R0G_A/Ac B0SH_A/Ve1+G_A/Ft22/Wp1 ELESU(Noe2) TEI/Ai \$ANGDA(Noe1+N/Fc8 BVTA/Ne1 TAI/Ai CAYIDAM(Noe1-I/Fc31) DAG_A/Ve1+JV/Fn1./Wp1 HAYA/Dc NIGE/Mu HAR_A/Ac GER/Ne1-ECE/Fc41 VTVG_A/Ne2 SVGVNAGLA/Ve2+N_A/Fs21 ./Wp1

This is a fragment of *Xagučin Xüü* excerpted from the part of “One Million” (pre-processed by manual proofreading) that was a corpus of POS tagging. The original corpus is “Хагучин нийн/ баяцан/ хүчлийн/ эхийн/ салбарын/ төслийн/ зорилтуудын/ ...”

After each word “/” is a code for POS and additional components which consisting of a mixture of uppercase and lowercase letters and numbers. For example, Ne1, Ne2, Ac, Ve1, Ve2, Db, Mu, and Ai are marks of POS which means countable nouns, uncountable nouns, nature adjectives, transitive verbs, intransitive verbs, state adverbs, cardinal numbers, and relational adjectives respectively. Fc, Ft, Fn and Fs represent different configuration additional component tag codes such as grammatical case, shape verb, auxiliary verb, and verb statement of Mongolian language. These marks are derived from the national standard “Information Technology-Mongolian word and expression marks for information processing” (GBT26235-2010)³. So lexical information annotation provides a basic knowledge for lexical analysis and POS statistics.

2) The text is a syntactically parsed corpus, for example: {{{{HAGVCIN/Ac/ HUU/Ne1}}NP2d/sbj {{{{HVLA/Ac/spcM0RI/Ne1+N/Fe8}}NP2dDEGER_E/Oa}OP/adb {HONDELEN/Ac/adbSAGV/Ve2+JV/Fn1}}VP2b}VP5b/abd {HATARI/Ve2+GVL/Fe1

1+V/Zv1+GSAGAR/FnC/adb{HUR/Ve2+CU/Fn1/adbIR_E/Vz1+BE/Fs14} VP2s/prd}VP3b}VP8h}VP10u ./Wp1}S11.

This is a simple sentence based on the syntactic information of Mongolian Phrase Treebank. The original corpus is “Хагучин нийн/ баяцан/ хүчлийн/ эхийн/ салбарын/ төслийн/ зорилтуудын/ ...”. The annotation includes the information of phrase structure and sentence component.

Where { }S denotes the beginning of a sentence, { A N}, {N N} and {V V} denote the start of the NP and VP phrases in a sentence respectively. Similar to the above, followed by each word “/” is a mark of POS, “.” means the period, “Wp1” means the end of a sentence; sbj, prd, obj, spc, adb represent the five sentence components of Mongolian such as subject, predicate, object, attributive and adverbial. Such syntactically parsed corpus can be used to accomplish from the shallow-level study of statistics about the combinational patterns of various POS tags and the sentence number, sentence length or average sentence length to the in-depth study of quantitative analysis on the general sentences, compound sentences or various sentence patterns and so on.

3) The text is a semantically tagged corpus, for example: HODEGE (Nob21) -YIN BAYIDAL (NuB) \uij \$ALTAR (Aboe2) B0LTAR (Aboe2) ./Wp1 CAG (Ne1) -VN VLARIL (Ncg3) \uij 0R0G_A (Abot2) B0SHAG_A (Abot2) ./Wp1 ELESUTEI SANGDA (Nob1) \oro BVTATAI CAYIDAM (Nob1) \oro-I DAGAJV (Vh11) ./Wp1 HAYA NIGE HAR_A (Abbo2GER (Nob22) \irt -ECE VTVG_A (Nbbu14) \uij SVGVNAGLAN_A (Vsb3) ./Wp1

Semantic annotations include semantic categories and semantic case. The former is marked according to the relevant mark of the semantic classification system of “Information Technology Mongolian Word’s Semantic Classification System and Its Mark Set” (MGC/01-05)⁴. The latter is based on the semantic case system as a standard reference to completes the semantic notation that was developed by “The Frame Research on Mongolian Language” [11].

The mark in the brackets () after each word indicates the semantic classification information which it belongs to, such as Nob21 (natural place), NuB (condition), Aboe2 (disordered substance’s attribute), Abot2 (dynamic attribute), Vh11 (moving action), Abbo2 (ordinary), Nbbu14 (natural phenomenon), Vsb3 (diffuse state), etc.; after each word “/” is followed by semantic case, such as uij (subject), oro (placement), irt (source), TVS (objective) and so on.

The core problem of semantic annotation is semantic disambiguation. In the above text, for example, HAR_A is a polysemous word with 11 different meanings such as (1) black ink, black; (2) dirty; (3) ordinary, rough; (4) pure, empty; ... [12]. It is labeled here as the third meaning according to the current context, meaning “ordinary, rough” rather than “color”, which provides a basis for automatic disambiguation.

Through the above annotation processing, a multi-level finishing corpus with lexical-syntactical-semantic information (the sample corpus is shown in Figure 3) can be obtained, providing detailed knowledge resources for other applications such as machine translation, text retrieval, semantic automatic processing, and tree-bank development.

B. Statistical Analysis of the Corpus

Using the preliminary processed corpus with word segmentation and part-of-speech tagging, according to the

principle of statistical linguistics, the paper will discuss a research on word class, word frequency statistics and lexical richness calculation ant its importance by the text of "Xagučin Xüü".

1) For word class, a statistical analysis on all of text of "Xagučin Xüü" and its related word class information was conducted. According to the statistics, "Xagučin Xüü" has 405 words and 34 fixed phrases, which can be classified into 13 word classes. The corresponding number and proportion of each word class are shown in Table II.

Table II only shows the statistical results of the word class information of an article. According to the statistical data on Da.Nachugdorji's works in Figure 1, his literary works include political poetry, lyric poetry, healthy poetry, comedy, novels, prose and translated poetry, and the corpus also has a large scale. Among it, the 439 words of the prose "Xagučin Xüü" are involved in 13 word classes. Because *Xagučin Xüü* is the author's representative work, the study can use its 13 word classes as a basic standard to statistically

purpose is to use a large amount of real language data to analyze the rule and pattern of Mongolian language in real using from researching the distribution frequency of the lexical information.

According to the statistical findings, the word accounted for 92.25%, the fixed phrase accounted for 7.75% in 439 words of the whole text (punctuation excluded; the fixed phrase is counted as a word unit); from the POS distribution, there are 13 word classes. The word frequency of each word class has been counted separately, such as the noun word frequency statistics in Table III and the fixed word frequency statistics in TableIV. The order of the proportion of each word class in mathematical order from high to low is expressed as follows:

a. Word class: noun > verb > adjective > time and position word = pronoun > numeral > statement word> time word > conjunction > postposition > adverb > quantifier > modal word;

b. Fixed phrase class: compound noun > compound

POS	word		fixed phrase		term
	noun	function word	compound word	idiom	
					Personal name
					fixed word
					verbal idiom
					adverb
					compound
					verb
					compound verb
					compound time and position
					compound
					adjective
					adverb
					postposition
					conjunction
					statement word
					adverb
					adjective
					quantifier
					time and position word
					pronoun
					numeral
					fixed phrase
					verb
					noun
					function word
					compound word
					idiom
					term
					Personal name
					fixed word
					verbal idiom
					adverb
					compound
					verb
					compound verb
					compound time and position
					compound
					adjective
					adverb
					postposition
					conjunction
					statement word
					adverb
					adjective
					quantifier
					time and position word
					pronoun
					numeral
					fixed phrase
					verb
					noun
					function word
					compound word
					idiom
					term
					Personal name
					fixed word
					verbal idiom
					adverb
					compound
					verb
					compound verb
					compound time and position
					compound
					adjective
					adverb
					postposition
					conjunction
					statement word
					adverb
					adjective
					quantifier
					time and position word
					pronoun
					numeral
					fixed phrase
					verb
					noun
					function word
					compound word
					idiom
					term
					Personal name
					fixed word
					verbal idiom
					adverb
					compound
					verb
					compound verb
					compound time and position
					compound
					adjective
					adverb
					postposition
					conjunction
					statement word
					adverb
					adjective
					quantifier
					time and position word
					pronoun
					numeral
					fixed phrase
					verb
					noun
					function word
					compound word
					idiom
					term
					Personal name
					fixed word
					verbal idiom
					adverb
					compound
					verb
					compound verb
					compound time and position
					compound
					adjective
					adverb
					postposition
					conjunction
					statement word
					adverb
					adjective
					quantifier
					time and position word
					pronoun
					numeral
					fixed phrase
					verb
					noun
					function word
					compound word
					idiom
					term
					Personal name
					fixed word
					verbal idiom
					adverb
					compound
					verb
					compound verb
					compound time and position
					compound
					adjective
					adverb
					postposition
					conjunction
					statement word
					adverb
					adjective
					quantifier
					time and position word
					pronoun
					numeral
					fixed phrase
					verb
					noun
					function word
					compound word
					idiom
					term
					Personal name
					fixed word
					verbal idiom
					adverb
					compound
					verb
					compound verb
					compound time and position
					compound
					adjective
					adverb
					postposition
					conjunction
					statement word
					adverb
					adjective
					quantifier
					time and position word
					pronoun
					numeral
					fixed phrase
					verb
					noun
					function word
					compound word
					idiom
					term
					Personal name
					fixed word
					verbal idiom
					adverb
					compound
					verb
					compound verb
					compound time and position
					compound
					adjective
					adverb
					postposition
					conjunction
					statement word
					adverb
					adjective
					quantifier
					time and position word
					pronoun
					numeral
					fixed phrase
					verb
					noun
					function word
					compound word
					idiom
					term
					Personal name
					fixed word
					verbal idiom
					adverb
					compound
					verb
					compound verb
					compound time and position
					compound
					adjective
					adverb
					postposition
					conjunction
					statement word
					adverb
					adjective
					quantifier
					time and position word
					pronoun
					numeral
					fixed phrase
					verb
					noun
					function word
					compound word
					idiom
					term
					Personal name
					fixed word
					verbal idiom
					adverb
					compound
					verb
					compound verb
					compound time and position
					compound
					adjective
					adverb
					postposition
					conjunction
					statement word
					adverb
					adjective
					quantifier
					time and position word
					pronoun
					numeral
					fixed phrase
					verb
					noun
					function word
					compound word
					idiom
					term
					Personal name
					fixed word
					verbal idiom
					adverb
					compound
					verb
					compound verb
					compound time and position
					compound
					adjective
					adverb
					postposition
					conjunction
					statement word
					adverb
					adjective
					quantifier
					time and position word
					pronoun
					numeral
					fixed phrase
					verb
					noun
					function word
					compound word
					idiom
					term
					Personal name
					fixed word
					verbal idiom
					adverb
					compound
					verb
					compound verb
					compound time and position
					compound
					adjective
					adverb
					postposition
					conjunction
					statement word
					adverb
					adjective
					quantifier
					time and position word
					pronoun
					numeral
					fixed phrase
					verb
					noun
					function word
					compound word
					idiom
					term

high-frequency words, and its vocabulary consists of many different low-frequency words.

TABLE III. FREQUENCY STATISTICS OF NOUN APPEARED IN THE TEXT OF "XAGUČIN XÜÜ"

It can be seen that the different words used in the author's sentence are very rich. According to Read (2000), lexical richness is the proportion of less common words or advanced vocabulary in the author's text. Therefore, there are 439 words totally except for repeated types. The total number of different words in the text is 431, which indicating that the lexical repetition rate is very low, the author's productive vocabulary is very high, and only about 2% of the words in this article belong to high frequency words. This result can fully indicate that the ratio of low-frequency words in the super 400 vocabularies to the total tokens of the text is more than 98%, mainly because the author's productive vocabulary is too high, unrepeated and advanced vocabulary appear more.

TABLE IV. FREQUENCY STATISTICS OF FIXED PHRASES
APPEARED IN THE TEXT OF "XAGUČIN XÜÜ"

TABLE V. STATISTICS ON THE NUMBER OF OCCURRENCES AND
THE CORRESPONDING WORD'S NUMBER AND ITS PROPORTION

number of occurrences	1	2	3	4	5	6	7	8	9
word's number	248	355	17	7	1	1	2	1	1
proportion	79.23%	11.18%	5.43%	2.24%	0.23%	0.23%	0.46%	0.23%	0.23%

There are many measuring methods of lexical richness in general. Scholars of domestic and foreign had use different measuring methods for different research purposes. For example, Linnarud (1986) measures overall

lexical abilities from four aspects: lexical uniqueness, lexical complexity, lexical diversity, and lexical density [13]. Based on this corpus, many different topics can be accomplished with employing modern technology and methods such as the calculation on the proportion of lexical/content words such as nouns, verbs and adjectives in the composition, the proportion of using different words, and the scope of vocabulary used by the author et all.

III. CONCLUSION

The corpus of *Da.Nachugdorji's Work* is only the starting point for digital research and development of the corpus of Mongolian literary works. Expanding and perfecting the scale and quality of this corpus is the core of the further study. Considering it as a premise, the research group will develop a lot of literary corpora such as "100 pieces of ancient literature of Mongolian" (1-4 volumes) (Mongol, C. DamdinSurung, 1979) and "the Classic Novels of Mongolian" (Mongol, 2009), "the Collected Work of R.Choynom" (Transcoded by B. DamdinRurung, 2014) , laying the foundation for the construction of large-scale literary corpus. On the other hand, one of the gratifying things is that "the Complete Works of Da.Nachugdorji" edited by Y.Temurjin(volumes 1-3) [14], was published by Yuanfang Publishing House in November 2016. In the near future, the research group can reference or use this version to regulate or expand the corpus, so as to achieve the perfect quality indicators of resources. At the same time, the research group will cooperate with the Institute of Language and Literature of the Mongolian Academy of Science to develop its Cyrillic edition. It is expected to achieve the sharing of different cultural resources and provide learners with a large number of real and natural corpus, which will become a teaching resources of learning and application.

REFERENCES

- [1] Reprint from an interview with the starting ceremony of “the Complete Works of Da.Nachugdorji” and the symposium to commemorate the 110th anniversary of the birth of Da.Nachugdorji.Hohhot,China, December 16, 2016.
 - [2] MAHLBERG M : Corpus Stylistics and Dickens’s fiction.New York:Routledge.2013, 5-14.
 - [3] GuanhuiMa, A Corpus-based Study of Novel Stylistics, Journal of Changshu Institute of Technology, No. 5, 2005, 4-6.
 - [4] [10] PinganHe, Corpus Linguistics and English Teaching, Foreign Language Teaching and Research Press, October 2004, 34, 9.
 - [5] HuashaBao,Badam-odsar : A status analysis and improvement strategy of Mongolian corpus, language calculation and content-

- based text processing - Proceedings of the 7th National Conference on Computational Linguistics, Harbin, August 2003, 346-350.
- [6] Hastuya_a, Construction of common noun's section in the electronic dictionary of khokh svdar, Master's degree thesis of Inner Mongolia University, May2008.
 - [7] GuirongLi, Construction of function word's section in the electronic dictionary of khokh svdar,, Master's degree thesis of Inner Mongolia University, May2010.
 - [8] Purubsurung, Establishing a corpus of Mongolia-based on Cyrillic Mongolian material, Doctor's degree thesis of Inner Mongolia University, May 2015.
 - [9] Xiaojuan, Processing and application of corpus of NIGEN DABHVR ASAR, Master's degree thesis of Inner Mongolia University, May 2015.
 - [11] D.Sarn_a, the Frame Research on Mongolian Language, Liaoning Nationalities Publishing House, March 2013.
 - [12] Mongolian Dictionary Compilation Group: "Mongolian Dictionary", Inner Mongolia People's Publishing House, November 1997, 1238
 - [13] JianlinChen, Research on the lexical richness of two genres of college English majors based on CEW corpus, Journal of Tianjin Foreign Studies University, No4, 2011.
 - [14] Tiejun edited, "the Complete Works of Da.Nachugdorji", Yuanfang Publishing House, November 2016.

Towards Robust Neural Machine Reading Comprehension via Question Paraphrases

Ying Li

*National Engineering Laboratory for
Brain-inspired Intelligence Technology and Application (NEL-BITA)
University of Science and Technology of China,
Hefei, China
nicole@baidu.com*

Hongyu Li, Jing Liu

*Baidu Inc.
Beijing, China
lihongyu04@baidu.com, liujing46@baidu.com*

Abstract—In this paper, we focus on addressing the *oversensitivity* issue of neural machine reading comprehension (MRC) models. By *oversensitivity*, we mean that the neural MRC models give different answers to question paraphrases that are semantically equivalent. To address this issue, we first create a large-scale Chinese MRC dataset with high-quality question paraphrases generated by a toolkit used in Baidu Search. Then, we quantitatively analyze the *oversensitivity* issue of the neural MRC models on the dataset. Intuitively, if two questions are paraphrases of each other, a robust model should give the same predictions. Based on this intuition, we propose a regularized BERT-based model to encourage the model give the same predictions to similar inputs by leveraging high-quality question paraphrases. The experimental results show that our approaches can significantly improve the robustness of a strong BERT-based MRC model and achieve improvements over the BERT-based model in terms of held-out accuracy. Specifically, the different prediction ratio (DPR) for question paraphrases of the proposed model decreases more than 10%.

Keywords-machine reading comprehension; oversensitivity; question paraphrases;

I. INTRODUCTION

Machine reading comprehension (MRC) requires machines to understand text and answer questions about the text, and it is an important task in natural language processing. With the increasing availability of large-scale labeled datasets for MRC ([1], [2], [3]) and the development of deep learning techniques ([8], [10], [11], [12], [13], [14], [15]), MRC has achieved remarkable advancements in the last few years.

Although a number of neural models obtains high held-out accuracy on several datasets, previous studies show that most of the complex neural models are not robust: different ways of phrasing the same question can often cause different answers.

Specifically, given a passage and two questions that are paraphrases of each other, a neural MRC model with high held-out accuracy may give different answers. As shown in Table I, *Question 1* and *Question 2* are paraphrases of each other, and we expect that a neural MRC model gives the same answer to these two questions. However, a BERT-based model that is one of the state-of-the-art MRC models, predicts two different answers. Additionally, the difference between *Question 3* and *Question 4* is just one

question mark. Surprisingly, the BERT-based model again gives different predictions to these two questions.

These above examples suggest that the neural MRC models are very sensitive to similar inputs that are semantically equivalent. The *oversensitivity* of the neural MRC models may limit their applications to question answering systems or search engines, which require consistent predictions on various inputs. For example, the search engine users may use different ways to express the same information need. If the system provides different answers to the questions that are paraphrases of each other, it may hurt the user experiences.

Given the great variety of languages for semantically equivalent expressions, it is not surprise that previous work has investigated the use of paraphrases to machine reading comprehension or question answering systems. The previous work can be classified into three categories. The first one uses paraphrases in the context of neural question answering models ([18], [19]), and encourages the models to learn similar representations for the questions that are paraphrases of each other. Another category of work directly generates question paraphrases and applies the question paraphrases to a question answering module by scoring them, because the generated paraphrases often contain low-quality candidates ([17], [21], [22], [23], [24]). The third category of research mines high quality semantically equivalent adversarial rules to generate question paraphrases by involving human-in-the-loop [16].

Although the previous work tried to incorporate question paraphrases to improve the performance of the question answering systems, they did not explicitly address the *oversensitivity* issue. It is not clear to what extent the issue was addressed.

In this paper, we focus on addressing the *oversensitivity* issue of neural MRC models. We first create a large-scale Chinese MRC dataset with high-quality question paraphrases generated by a toolkit used in Baidu Search. Then, we quantitatively analyze the *oversensitivity* issue of a BERT-based MRC model, that is one of the state-of-the-art neural MRC models. Intuitively, if two questions are paraphrases of each other, a robust model should give the same predictions. Based on this intuition, we propose a regularized BERT-based model by incorporating the high-quality question paraphrases. The experimental

Table I
THE EXAMPLES OF THE OVERSENSITIVITY OF A BERT-BASED MRC MODEL.

Passage: 12月24号是平安夜，12月25号是圣诞节；分别相当于中国的大年29和大年30。(December 24th is the Christmas Eve, and December 25th is the Christmas; they are equivalent to the Spring Festival Eve and the Spring Festival in China.)	
Question 1: 12月24日是什么日子 (What special day is December 24)	Predicted Answer: 平安夜 (Christmas Eve)
Question 2: 12月24日是什么节日 (What holiday is December 24)	Predicted Answer: 圣诞节 (Christmas)
Passage: 求一个数的立方根的运算方法，叫做开立方。它是立方的逆运算，最早在我国的九章算术中有对开立方的记载。由于任何实数均有唯一的立方与之对应且不存在两个实数的立方相等，故任何实数都存在且仅存在唯一的立方根。(The method of finding the cube root of a number is called “kai li fang” in Chinese. It is the inverse of the cube operation. It was first recorded in the “Jiuzhang arithmetic” in our country. Since any real number and its cube is a unique pair and no cubes of two real numbers are equal, so the cube root of any real number exists and is unique.)	
Question 3: 任何实数存在多少立方根 (How many cube roots does a real number have)	Predicted Answer: 唯一 (Unique)
Question 4: 任何实数存在多少立方根? (How many cube roots does a real number have?)	Predicted Answer: 开立方 (Cube root)

results show that our approaches can significantly improve the robustness of a strong BERT-based MRC model and achieve improvements over the BERT-based model in terms of held-out accuracy.

The contributions of this paper are three-folds:

- We create a large-scale Chinese MRC dataset with high-quality question paraphrases (see Section II-C). The dataset contains 85K passages and 242K questions, and each of the question has 5 ~ 10 high-quality paraphrases.
- We quantitatively analyze the *oversensitivity* issue of a BERT-based MRC model (see Section III-A). To the best of our knowledge, this is the first study of its kind, indicating a potential direction for future research.
- To address the *oversensitivity* issue, we propose a regularized BERT-based model by incorporating high-quality question paraphrases (see Section III-B). The experimental results show that our approaches can significantly improve the robustness of a strong BERT-based MRC model and achieve improvements over the BERT-based model in terms of held-out accuracy. Specifically, the different prediction ratio (DPR) for question paraphrases of the proposed model decreases more than 10% (see Section IV).

The remainder of this paper is organized as follows. Section II describes a strong BERT-based MRC model, and introduces a large-scale MRC dataset with high-quality question paraphrases generated by a toolkit used in Baidu Search. Section III quantitatively analyzes the *oversensitivity* issue of a BERT-based MRC model, and proposes a regularized BERT-based model by incorporating the high-quality question paraphrases. In Section IV, we give the experimental results. Section V represents the related work. Section VI ends with conclusions.

II. BACKGROUND

A. A BERT-based MRC Model

Recently, the pre-trained language models ([12], [13], [14], [15]) have caused a stir in the MRC community.

Among the pre-trained models, BERT [13], which uses Transformer encoder and trains a bidirectional language model, is one of the most successful models by far, presenting new state-of-the-art results in MRC. In this paper, we choose a BERT-based model as a baseline. Basically, a BERT-based MRC model has an encoding layer and an output layer.

BERT Encoding Layer: This layer uses Transformer encoder to model passages and questions. It takes as input passage $P = \{p_i\}_{i=1}^m$ and question $Q = \{q_j\}_{j=1}^n$, and computes for each token a context-aware representation.

Specifically, given passage $P = \{p_i\}_{i=1}^m$ and question $Q = \{q_j\}_{j=1}^n$, we first pack them into a single sequence of length $m + n + 3$, i.e.,

$$S = [\langle \text{CLS} \rangle, Q, \langle \text{SEP} \rangle, P, \langle \text{SEP} \rangle],$$

where $\langle \text{SEP} \rangle$ is the token separating Q and P , and $\langle \text{CLS} \rangle$ the token for classification (will not be used in this paper). For each token s_i in S , we construct its input representation as

$$\mathbf{h}_i^0 = \mathbf{s}_i^{\text{tok}} + \mathbf{s}_i^{\text{pos}} + \mathbf{s}_i^{\text{seg}},$$

where $\mathbf{s}_i^{\text{tok}}$, $\mathbf{s}_i^{\text{pos}}$, and $\mathbf{s}_i^{\text{seg}}$ are the token, position, and segment embeddings for s_i , respectively. Tokens in Q share a same segment embedding \mathbf{q}^{seg} , and tokens in P share a same segment embedding \mathbf{p}^{seg} . Such input representations are then fed into L successive Transformer encoder blocks, i.e.,

$$\mathbf{h}_i^\ell = \text{Transformer}(\mathbf{h}_i^{\ell-1}), \quad \ell = 1, 2, \dots, L,$$

so as to generate deep, context-aware token representations for passages and questions. We refer readers to [13] for details. The final hidden states $\{\mathbf{h}_i^L\}_{i=1}^{m+n+3} \in \mathbf{R}^{d_1}$ are taken as the output of this layer.

Output Layer: We follow BERT and simply use a linear output layer, followed by a standard softmax operation, to predict answer boundaries. The probability of each token s_i to be the start or end position of the answer span is calculated as:

$$p_i^1 = \frac{\exp(\mathbf{w}_1^\top \mathbf{h}_i)}{\sum_j \exp(\mathbf{w}_1^\top \mathbf{h}_j)}, \quad p_i^2 = \frac{\exp(\mathbf{w}_2^\top \mathbf{h}_i)}{\sum_j \exp(\mathbf{w}_2^\top \mathbf{h}_j)},$$

where \mathbf{h}_i is the output of token s_i by the encoding layer, and $\mathbf{w}_1, \mathbf{w}_2 \in \mathbf{R}^{6d_1+6d_2}$ are trainable parameters. The training objective is the negative log-likelihood of the true start and end positions:

$$\mathcal{L}_1 = -\frac{1}{N} \sum_{j=1}^N (\log p_{y_j^1}^1 + \log p_{y_j^2}^2),$$

where N is the number of examples in the dataset, and y_j^1, y_j^2 are the true start and end positions of the j -th example, respectively. At inference time, the span (a, b) where $a \leq b$ with maximum $p_a^1 p_b^2$ is chosen as the predicted answer.

B. A Question Paraphrase Generation Toolkit

There are various resources (e.g. PPDB [25]) and approaches (e.g. neural paraphrase generation [26], [28], [27]) can be used to generate paraphrases. A common problem with the generated paraphrases is that they often contain inappropriate or low-quality candidates. Hence, a paraphrase scoring module is usually employed in the downstream tasks.

In this paper, we employ a question paraphrase generation toolkit used in Baidu Search. The major advantage of this toolkit is that it can generate high-quality question paraphrases. Given 100 sampled questions, the toolkit can generate around 10 paraphrases for each question on average. We further manually evaluate the generated question paraphrases, the accuracy is around 98%. Basically, there are two modules in the toolkit.

Paraphrase Candidates Retrieval: Given a question, this module will retrieve similar questions as candidates from an inverted index of search logs in Baidu Search.

Paraphrase Similarity Model: Given each pair of a question and its retrieved paraphrase candidate, this module will estimate their semantic similarity and determines if they are paraphrases of each other based on a pre-defined threshold. The similarity model is a BERT-based model that was fine-tuned a large-scale labeled datasets containing more than 500K question pairs.

This toolkit benefits from both the real search engine logs and a well-tuned BERT-based similarity model. Hence, it can generate high-quality paraphrase candidates. We create a large-scale MRC dataset with the high-quality question paraphrases generated by the toolkit (see the next Section).

C. A Chinese MRC Dataset with Question Paraphrases

In this paper, we construct a large-scale Chinese MRC dataset. The way we construct this dataset is similar to SQuAD [1]. We first collect passages from Baidu Baike¹. Then, we ask crowd sourcing workers to ask questions and annotate the corresponding answers to each passage.

¹ <http://baike.baidu.com>

Datasets	#Passages	#Questions	avg. #Que _{paraphrases}
Train	77.2K	217.3K	4.7
Dev	4.3K	12.1K	9.2
Test	4.3K	12.1K	9.2

Table II
THE STATISTICS OF THE MRC DATASET.

Question Class	Percentage
Entity	66.1%
Number	23.9%
Description	9.8%
Other	0.2%

Table III
THE FREQUENCY DISTRIBUTION OF DIFFERENT QUESTION CLASSES.

There are 85K passages and 242K questions in total. Additionally, the average length of questions and passages are 8.8 and 151.7 Chinese characters, respectively. The average length of answers is 6.4 characters. Table III shows the frequency distribution of different question classes. We can observe that the most frequent questions are factoid questions. Hence, the average length of answers is relatively short. As shown in Table II, we randomly divide the dataset into training, development and test sets.

We further use the question paraphrase generation toolkit to generate paraphrases for each question in each sample. On average, there are 4.7 question paraphrases for each question in training set, and 9.2 question paraphrases for each question in the development and test sets. Note that we limit the maximal number of question paraphrases in the training set to reduce the training time.

III. APPROACH

In this section, we first analyze the oversensitivity issue of neural MRC models. Then, we propose a regularized approach to improve the robustness of the neural models.

A. The Oversensitivity of a BERT-based Model

In this paper, we use different prediction ratio (DPR) to measure the *oversensitivity* of a neural MRC model. We define the *DPR* of a neural MRC model $f(\theta)$ on a dataset D as follows.

$$DPR_D(f(\theta)) = \frac{\|Q\| - \Gamma(f(\theta))}{\|Q\|},$$

where Q represents the set of original questions in dataset D , and $\|Q\|$ represents the total number of original questions without paraphrasing. Γ indicates the number of original questions whose all paraphrases make the neural model $f(\theta)$ predict the same answers. Formally, Γ is further defined as follows.

$$\Gamma = \sum_{q \in Q} \prod_{q_k \in \tilde{q}} \mathbb{1}[f(\theta; q_k, p) = f(\theta; q, p)],$$

where q_k is a paraphrase of the original question q , \tilde{q} indicates the set of all the paraphrases of q , and p is a

Models	DEV DPR %	TEST DPR %
BERT Baseline	28.89	29.46
+ Question Paraphrases	18.93	19.63
+ Question Paraphrases, + Regularization	18.57	19.13

Table IV
THE OVERSENSITIVITY OF THE MRC MODELS.

Models	DEV		TEST	
	Rouge-L%	EM%	Rouge-L%	EM%
BERT Baseline	89.47	78.67	89.84	78.44
+ Question Paraphrases	90.28	79.57	90.54	79.33
+ Question Paraphrases, + Regularization	90.38	79.90	90.84	79.90

Table V
THE HELD-OUT ACCURACY OF THE MRC MODELS.

passage. A low *DPR* score means the MRC model is robust with respect to question paraphrases.

We train a BERT-based MRC model without question paraphrasing, and estimate the *DPR* (i.e. *oversensitivity*) of this model on both dev set and test set that contain paraphrases. The DPRs are 28.89% and 29.46% on dev set and test set, respectively. We can see that the strong BERT-based model is very sensitive to the similar inputs that are semantically equivalent.

B. A Regularized BERT-based Model

In the previous section, we can observe that the BERT-based model is not robust. To address the *oversensitivity* issue, we propose a regularized BERT-based model. Intuitively, if two questions are paraphrases of each other, a robust model should give the same answer. That is to say, the probability distributions of the answer start or end positions for question paraphrases should be closed to each other. Based on this intuition, we introduce a regularization loss by leveraging the question paraphrases, to encourage the model give closed predictions to question paraphrases. The regularization loss is defined as follows.

$$\mathcal{L}_2 = \frac{1}{N} \sum_{j=1}^N \frac{1}{2(K+1)^2} \sum_{k_1, k_2=1}^{k_1, k_2=K+1} \text{cross_entropy}(\mathbf{p}_{j, k_1}^1, \mathbf{p}_{j, k_2}^1) + \text{cross_entropy}(\mathbf{p}_{j, k_1}^2, \mathbf{p}_{j, k_2}^2),$$

where N is the total number of the original questions. K is the number of paraphrases of the j -th question. \mathbf{p}_{j, k_1}^1 is the probability distribution of the answer start position for the k_1 -th paraphrase of j -th question. Similarly, \mathbf{p}_{j, k_1}^2 is the probability distribution of the answer end position for the k_1 -th paraphrase of j -th question. If the neural model is robust, the regularization loss \mathcal{L}_2 should be small. Otherwise, the model might be *oversensitivity*.

Finally, by combining the answer prediction loss \mathcal{L}_1 (described in Section II-A) and the regularization loss \mathcal{L}_2 , we expect that the model can simultaneously achieve high held-out accuracy through minimizing the answer prediction loss, and obtain high robustness through minimizing the regularization loss. Formally, the combined objective will be as follows.

$$\mathcal{L} = \mathcal{L}_1 + \lambda \mathcal{L}_2,$$

where λ is a hyper-parameter for the linear combination. In the experiment, we set λ as 1.

IV. EXPERIMENTS

In this section, we first introduce the evaluation metrics. Then, we give the comparison settings of different models and the training details. Last, we present the experimental results.

A. Evaluation Metrics

To evaluate the held-out accuracy of an MRC model, we use the following two metrics, i.e. ROUGE-L [29] and exact match (EM). ROUGE-L can be viewed as a metric to measure the partial correctness of the predicted answers. To calculate these two metrics, we first normalize the predicted and reference answers by removing spaces and punctuations. We then do the calculation in Chinese character-level.

To evaluate the *oversensitivity* of an MRC model, we use different prediction ratio (DPR) that is defined in Section III-A. The lower the DPR is, the more robust the model is.

B. Comparison Settings

In this section, we compare three models: (i) a BERT-based MRC model, that is trained on the dataset without paraphrasing, (ii) a BERT-based MRC model, that is trained on the dataset augmented by question paraphrases, (iii) a regularized BERT-based model, that is trained on the dataset augmented by question paraphrases.

C. Training Details

For all the settings of BERT-based models, we initialize parameters of the BERT encoding layer with pre-trained models officially released by Google². These models were pre-trained on the corpus of Chinese Wikipedia pages, using the tasks of masked language model and next sentence prediction [13]. The pre-trained model contains 12 Transformer encoding blocks, each with 12 self-attention heads and 768 hidden units. There are 110M

² <https://github.com/google-research/bert>

parameters in the model. Throughout our experiments, we use this setting unless specified otherwise. Other trainable parameters are randomly initialized.

We use the Adam optimizer [30] with a learning rate of 3e-5 and a batch size of 32. The number of training epochs is 2, according to the best EM and Rouge-L scores on the dev set. During training, the pre-trained BERT parameters will be fine-tuned with other trainable parameters.

D. Experimental Results

The main experimental results have been shown in Table IV and Table V.

Table IV shows the *oversensitivity* of the MRC models. We can observe that the BERT baseline is very sensitive to the question paraphrases that are semantically equivalent. By directly training the model on the dataset augmented by question paraphrases, the robustness of the model has been significantly improved. The regularized BERT-based model obtains the best robustness by leveraging the question paraphrases to encourage the model give closed predictions to question paraphrases. Comparing to the BERT baseline, the *DPR* decreases more than 10%.

Table V shows the held-out accuracy of the MRC models. We can observe that the BERT baseline obtains good performance on the MRC dataset. Our proposed regularized BERT-based model shows the best performance in terms of Rouge-L and EM.

In a summary, the experimental results show that our approach can significantly improve the robustness of a strong BERT-based MRC model and achieve improvements over the BERT baseline in terms of Rouge-L and EM through optimizing both the regularization loss and answer prediction loss.

V. RELATED WORK

A. Neural Machine Reading Comprehension

In recent years, a number of large-scale datasets have been created for MRC, e.g., CNN/DM [6], SQuAD [1], SearchQA [4], TriviaQA [5], MS-MARCO [2], and DuReader [3].

These datasets have led to the advanced neural MRC models like Match-LSTM [7], BiDAF [8], DCN [9], R-Net [10], and QANet [11]. These end-to-end neural models have similar architectures, including an encoding layer, an attention-based interaction layer and a prediction layer.

Recently, the pre-trained language models such as ELMo [12], BERT [13], ERNIE [14] and XL-NET [15] have been proposed. These language models are deep neural networks, that are pre-trained on large-scale unlabeled text corpus to obtain contextual representations of text. When used in downstream tasks including MRC, the pre-trained contextual representations greatly improve the performance.

Although these neural MRC models achieves high held-out accuracy on particular datasets, they are often not robust: different ways of phrasing the same question can

often cause different answers. In this paper, we focus on addressing the *oversensitivity* issue of MRC models.

B. Paraphrasing for Question Answering

A number of previous work has investigated the use of paraphrases to machine reading comprehension or question answering systems. The previous work can be classified into three categories. The first one uses question paraphrases and encourages the models to learn similar representations for the question paraphrases ([18], [19]). The second category of research directly incorporates generated question paraphrases to a question answering module by scoring them, because the generated paraphrases often contain inappropriate candidates ([17], [21], [22], [23], [24]). The third category of research mines high quality semantically equivalent adversarial rules to generate question paraphrases by involving human-in-the-loop [16].

Although the previous work tried to incorporate question paraphrases to improve the performance of the question answering systems, they did not explicitly address the *oversensitivity* issue. It is not clear to what extent the issue was addressed.

In contrast, we quantitatively analyze the *oversensitivity* issue of a BERT-based MRC model and propose a regularized BERT-based model to improve the robustness of the model.

VI. CONCLUSION

In this paper, we focus on addressing the *oversensitivity* issue of neural machine reading comprehension (MRC) models. To address this issue, we first create a large-scale Chinese MRC dataset with high-quality question paraphrases generated by a toolkit used in Baidu Search. Then, we quantitatively analyze the *oversensitivity* issue of the neural MRC models. Intuitively, if two questions are paraphrases of each other, a robust model should give the same predictions. Based on this intuition, we propose a regularized BERT-based model to encourage the model give the same predictions to similar inputs by leveraging high-quality question paraphrases. The experimental results show that our approach can significantly improve the robustness of a strong BERT-based MRC model and achieve improvements over the BERT baseline in terms of Rouge-L and EM through optimizing both the regularization loss and answer prediction loss.

In the future work, we will investigate this idea with adversarial training to further improve both the robustness and held-out accuracy of the neural MRC models.

REFERENCES

- [1] Rajpurkar P, Zhang J, Lopyrev K, Liang P. *Squad: 100,000+ questions for machine comprehension of text*. In Proceedings of EMNLP. 2016.
- [2] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. *Ms marco: A human generated machine reading comprehension dataset*. In Proceedings of the Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches at NIPS. 2017.

- [3] W He, K Liu, J Liu, Y Lyu, S Zhao, X Xiao, Y Liu, Y Wang, H Wu, Q She. *DuReader: a Chinese Machine Reading Comprehension Dataset from Real-world Applications*. In Proceedings of Machine Reading for Question Answering (MRQA) Workshop at ACL. 2018.
- [4] Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. *Searchqa: A new q&a dataset augmented with context from a search engine*. arXiv:1704.05179. 2017.
- [5] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. *Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension*. In Proceedings of ACL. 2017.
- [6] Karl Moritz Hermann, Tomas Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. *Teaching machines to read and comprehend*. In Proceedings of NIPS. 2015.
- [7] Shuohang Wang and Jing Jiang. *Machine comprehension using match-lstm and answer pointer*. In Proceedings of ICLR. 2017.
- [8] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. *Bidirectional attention flow for machine comprehension*. In Proceedings of ICLR. 2017.
- [9] Caiming Xiong, Victor Zhong, and Richard Socher. *Dynamic coattention networks for question answering*. In Proceedings of ICLR. 2017.
- [10] Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. *Gated self-matching networks for reading comprehension and question answering*. In Proceedings of ACL. 2017.
- [11] AdamsWei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. *Qanet: Combining local convolution with global self-attention for reading comprehension*. In Proceedings of ICLR. 2018.
- [12] Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L. *Deep contextualized word representations*. In Proceedings of NAACL. 2018.
- [13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. *Bert: Pre-training of deep bidirectional transformers for language understanding*. In Proceedings of NAACL. 2019.
- [14] Sun Y, Wang S, Li Y, Feng S, Chen X, Zhang H, Tian X, Zhu D, Tian H, Wu H. *ERNIE: Enhanced Representation through Knowledge Integration*. arXiv preprint arXiv:1904.09223. 2019.
- [15] Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov R, Le QV. *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. arXiv preprint arXiv:1906.08237. 2019.
- [16] Ribeiro MT, Singh S, Guestrin C. *Semantically equivalent adversarial rules for debugging nlp models*. In Proceedings of ACL. 2018.
- [17] Jonathan Berant and Percy Liang. *Semantic parsing via paraphrasing*. In Proceedings of ACL. 2014.
- [18] Antoine Bordes, Sumit Chopra, and Jason Weston. *Question answering with subgraph embeddings*. In Proceedings of EMNLP. 2014.
- [19] Li Dong, Furu Wei, Ming Zhou, and Ke Xu. *Question answering over freebase with multicolumn convolutional neural networks*. In Proceedings of ACL. 2015.
- [20] Pablo Duboue and Jennifer Chu-Carroll. *Answering the question you wish they had asked: The impact of paraphrasing for question answering*. In Proceedings of NAACL. 2006.
- [21] Shashi Narayan, Siva Reddy, and Shay B Cohen. *Paraphrase generation from Latent-Variable PCFGs for semantic parsing*. In Proceedings of NLG. 2016.
- [22] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. *Paraphrase-driven learning for open question answering*. In Proceedings of ACL. 2013.
- [23] Bo Chen, Le Sun, Xianpei Han, and Bo An. *Sentence rewriting for semantic parsing*. In Proceedings of ACL. 2016.
- [24] Dong L, Mallinson J, Reddy S, Lapata M. *Learning to paraphrase for question answering*. In Proceedings of ACL. 2017.
- [25] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. *PPDB: the paraphrase database*. In Proceedings of NAACL. 2013.
- [26] Hasan SA, Lee K, Datla V, Qadir A, Liu J, Farri O. *Neural Paraphrase Generation with Stacked Residual LSTM Networks*. In Proceedings of COLING. 2016.
- [27] Li Z, Jiang X, Shang L, Li H. *Paraphrase Generation with Deep Reinforcement Learning*. In Proceedings of EMNLP. 2018.
- [28] Mallinson J, Sennrich R, Lapata M. *Paraphrasing revisited with neural machine translation*. In Proceedings of EACL. 2017.
- [29] Chin-Yew Lin *Rouge: A package for automatic evaluation of summaries*. In Text Summarization Branches Out: Proceedings of the ACL-04 Workshop, pages 74–81.
- [30] Diederik P. Kingma and Jimmy Ba. *Adam: A method for stochastic optimization..* CoRR,abs/1412.6980. 2014.

Employing Gated Attention and Multi-similarities to Resolve Document-level Chinese Event Coreference

Haoyi Cheng, Peifeng Li, Qiaoming Zhu

Natural Language Processing Lab, School of Computer Science and Technology
Soochow University
Suzhou, China
29029323@qq.com, {pfli, qmzhu}@suda.edu.cn

Abstract—Event coreference resolution is a challenging task. To address the issues of the influence on event-independent information in event mentions and the flexible and diverse sentence structure in Chinese language, this paper introduces a GANN (Gated Attention Neural Networks) model to document-level Chinese event coreference resolution. GANN introduces a gated attention mechanism to select event-related information from event mentions and then filter noisy information. Moreover, GANN not only uses a single Cosine distance to calculate the linear distance between two event mentions, but also introduces multi-mechanisms, i.e., Bilinear distance and Single Layer Network, to further calculate the linear and nonlinear distances. The experimental results on the ACE 2005 Chinese corpus illustrate that our model GANN outperforms the state-of-the-art baselines.

Keywords-gated attention; multi-similarities; chinese event coreference

I. INTRODUCTION

Event coreference resolution is an important task in NLP, which is the foundation of many NLP tasks, such as topic detection [1], information extraction [2], and reading comprehension [3]. In recent years, most studies focused on entity coreference resolution, and only a few concern event coreference resolution because it is a challenging task. The task of event coreference resolution is to determine which event mentions (a phrase or sentence within which an event is described) in texts refer to the same real-world event and then cluster them to a unique coreferential event chain. Take the following two event mentions as examples:

S1: NHK报道日本检方1号针对发生在东海村的油燃料加工厂不慎外泄事故起诉 6 名工厂的员工。(1st, Japanese prosecutors **sued** six employees of the oil fuel processing plant for the leakage accidents in Donghai. - from NHK.)

S2: 因为东海村油燃料加工厂去年发生事故而被起诉的 6 名员工，包括…。(Six employees were **sued** for the accident at the oil fuel processing plant in Donghai last year, including).

The event mention in S1 whose event trigger (the main word that most clearly expresses the occurrence of an event) is “起诉” (sue) and the mention in S2 with the trigger “起诉” (sue) refer to the same real-word event, a *Justice* event, and they are coreferential event mentions. Event mentions that refer to the same event can appear both within a document and across multiple documents. Hence, event coreference resolution is usually divided into document-level and cross-document level. This paper focuses on document-

level Chinese event coreference resolution, which is critical to further cross-document event coreference resolution.

Most previous studies on Chinese document-level event coreference resolution were based on feature engineering, which used lots of manual features and could not capture the semantics hiding in event mentions. Currently, neural network models were introduced to English event coreference resolution and achieved success. Krause [4] and Fang [5] employed the CNN model and the multiple decomposable attention networks to resolve document-level event coreference. However, there are two issues in above neural network models. The first is that they cannot eliminate the influence of noises derived from those event-independent information (e.g., “NHK” in S1) event mentions. The second is that they are not suitable for Chinese event coreference resolution for the language-independent. Compared with English, the event mentions in Chinese are more complex and have the Chinese characteristics, such as discourse-driven, pro-drop, zero entity coreference, etc.

To address the above two issues, this paper introduces a novel GANN (Gated Attention Neural Network) model to the task of document-level Chinese event coreference resolution. Firstly, GANN introduces a gated attention mechanism to select the event-related information from event mentions and then filter noisy information. Secondly, GANN not only uses a single Cosine distance to calculate the linear distance between two event mentions, but also introduces multi-mechanisms, i.e., Bilinear distance and Single Layer Network, to further calculate the linear and nonlinear distance. Hence, linear distance and nonlinear distance can complement each other. The experimental results on the ACE 2005 Chinese corpus illustrate that our model GANN outperforms the state-of-the-art baselines.

II. RELATED WORK

Event coreference is much less studied in comparison to the large number of studies on entity coreference [6]. Early work on document-level event coreference resolution used traditional methods, such as probability-based models [7] and graph-based models [8], most of them derived from the entity coreference literature and focused on English. Chen [9] proposed a pairwise event coreference model on various kinds of annotating event attributes. Especially, it took the four basic attributes of events (tense, polarity, molarity and genericity) as its features. Liu [10] also used a SVM-based pairwise model on annotating information and manual features (e.g., semantic information of trigger words, distance between event mentions). Since the above

pairwise classifiers do not take into account the document-level global information, there are many conflicts when pairwise results are transformed into coreferential event clusters. Lu and Ng [11] proposed a joint inference model based on Markov logic networks to correct the mistakes from the pairwise event coreference resolver. Liu [12] presented a unified graph framework to conduct event coreference by using many features (e.g., frame features, argument features and event mention distances). As for the task of Chinses event coreference resolution, there is only one literature concerned it. Teng [13] trained a Maximum Entropy Model on a large number of features extracted by the external tools.

Recently, neural networks have been widely-used in various English NLP applications. Krause introduced the Convolutional Neural Network (CNN) to event coreference and they used many annotating entity and event information. Fang introduced a multiple decomposable attention network from different views, i.e., event mention, event arguments and trigger context. Moreover, it applied document-level global inference mechanism to further resolve the coreference chains. This is the first paper to apply a neural

network to the task of document-level Chinses event coreference resolution.

III. GANN FOR EVENT COREFERENCE RESOLUTION

In this section, we first introduce the framework of our gated attention neural network called GANN and then describe its components, i.e., input layer, gated attention layer, similarity calculation layer and output layer.

A. Overview

Krause used a CNN to dig out the contextual information of words and it only considered local information between words regardless of the relationship between event mention pairs. Fang used an attention mechanism to extract relatively important features, but there are noises in those extracted features. To address the above issues and the characteristics of Chinese language, this paper introduces a novel neural network GANN to resolve document-level Chinese event coreference and its architecture is shown in Figure 1.

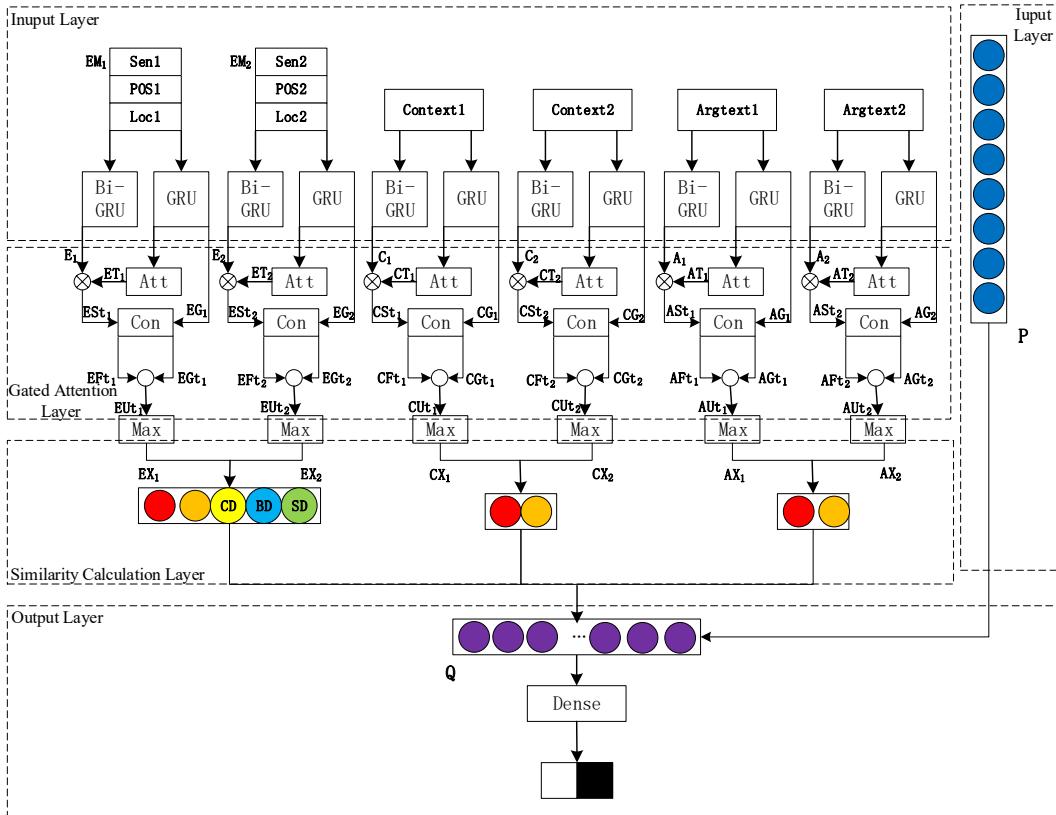


Figure1. The structure of GANN

GANN mainly introduces an attention mechanism to extract event-related features and then uses a gated mechanism to filters those event-independent information. Finally, it uses multiple similarity distances and pairwise features to resolve document-level Chinese event coreference. GANN is mainly divided into the following four parts: 1) input layer, 2) gated attention layer, 3) similarity calculation layer, and 4) output layer. In the input layer, various kinds of event mention representations (e.g., event mention, POS, location, context of the trigger, arguments and pairwise features) are encoded as vectors by

using Bi-GRU and GRU. In the gated attention layer, a self-attention mechanism is introduced to extract event-related features from event mentions and then a gating mechanism is used to remove those event-independent information to reduce noisy information and simplify computational complexity. In the similarity calculation layer, the Cosine distance, Bilinear distance and Single Layer Network are used to obtain the linear distance and nonlinear distance between two event mentions. In the output layer, GANN determines whether a pair of event mentions is coreferential or not.

B. Input

Following Krause and Fang, the input of GANN is two event mentions e_1 and e_2 with annotated triggers, event types/subtypes, event arguments, and event attributes (e.g., modality), etc. We extract the follows features from these two event mentions as follows.

1) **Sentential features:** the words in event mention (SF1), their POS tagged by NLTK tools (SF2), and their positions (SF3);

2) **Context features:** the context around the trigger (the windows size is set to 3) (CF1) and the argument list of the event mention (CF2);

3) **Pairwise features:** the comparison results of the event trigger (PF1), type (PF2), subtype (PF3), modality (PF4), polarity (PF5), genericity (PF6) and tense (PF7), respectively. If the above attributes of an event mention pair have the same annotated label, its corresponding feature will be assigned 1; otherwise 0.

Pre-trained Wikipedia 300-dimensional word vector matrix \mathbf{M} is used as the training matrix to encode SF1, SF2, CF1 and CF2 to four vectors **Sen**, **POS**, **Context** and **Argtext**, respectively. In Figure 1, **Sen1**, **POS1**, **Context1** and **Argtext1** are the vectors extracted from the event mention e_1 , while **Sen2**, **POS2**, **Context2** and **Argtext2** are the vectors from the event mention e_2 . Besides, the feature SF3 is encoded to 50-dimensions location vectors **Loc** by a random word embedding matrix.

For two event mentions e_1 and e_2 , we merge their **Sen**, **POS** and **Loc** vectors to a vector **EM**, respectively, as follows.

$$\mathbf{EM}_i = \text{Concat}(\mathbf{Sen}_i, \mathbf{Pos}_i, \mathbf{Loc}_i) \quad (i = 1, 2) \quad (1)$$

We use Bi-GRU to encode **EM_i**, **Context_i** and **Argtext_i** to get the new vectors **E_i**, **C_i** and **A_i**, and use GRU to encode **EM_i**, **Context_i** and **Argtext_i** to get the new vectors **EG_i**, **CG_i** and **AG_i**. Since the pairwise features PF1-PF7 are binary numbers, we merge these pairwise features and the distance between two event mentions to a vector **P**.

C. Gated Attention

Attention mechanism [14] is a useful method in many NLP applications and can reweight each word in an event mention when we apply it to event coreference resolution. In this paper, we design a gated attention network to combine the attention mechanism and gating mechanism. This network uses the attention mechanism to reweight all words in the event mention, the context of the trigger and the argument list, and then uses the gating mechanism to reduce noisy information.

The vectors **EM₁**, **EM₂**, **Context₁**, **Context₂**, **Argtext₁** and **Argtext₂** are input into the self-attention layer as follows.

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}(\mathbf{Q}\mathbf{K}^T / \sqrt{d_k})\mathbf{V} \quad (2)$$

$$\begin{aligned} \mathbf{ET}_i &= \text{Attention}(\mathbf{EG}_i, \mathbf{EG}_i, \mathbf{EG}_i) \\ &= \text{Softmax}(\mathbf{EG}_i \mathbf{EG}_i^T / \sqrt{d_k}) \mathbf{EG}_i \quad (i = 1, 2) \end{aligned} \quad (3)$$

$$\begin{aligned} \mathbf{CT}_i &= \text{Attention}(\mathbf{CG}_i, \mathbf{CG}_i, \mathbf{CG}_i) \\ &= \text{Softmax}(\mathbf{CG}_i \mathbf{CG}_i^T / \sqrt{d_k}) \mathbf{CG}_i \quad (i = 1, 2) \end{aligned} \quad (4)$$

$$\begin{aligned} \mathbf{AT}_i &= \text{Attention}(\mathbf{AG}_i, \mathbf{AG}_i, \mathbf{AG}_i) \\ &= \text{Softmax}(\mathbf{AG}_i \mathbf{AG}_i^T / \sqrt{d_k}) \mathbf{AG}_i \quad (i = 1, 2) \end{aligned} \quad (5)$$

where \mathbf{Q} , \mathbf{K} and \mathbf{V} are the input vectors, and $\sqrt{d_k}$ is used to limit on the value of the inner product and prevent it from being too large. The encoding process of the function *Attention()* is to match each word vector in the sequence \mathbf{Q} with each word vector in the sequence \mathbf{K} to obtain a similarity, and then multiply this similarity by the sequence \mathbf{V} to better preserve the original input information and represent the overall information. Hence, we obtain the important information vectors **ET_i**, **CT_i** and **AT_i**, respectively.

The vector **ET_i** is multiplied by the vectors **E_i**, the vector **CT_i** is multiplied by the vectors **C_i**, and the vector **AT_i** is multiplied by the vectors **A_i** as follows. This operation can amplify the vector gaps between words in event mention to make those event-related words more important than those event-independent words.

$$\mathbf{EST}_i = \mathbf{ET}_i \mathbf{E}_i \quad (i = 1, 2) \quad (6)$$

$$\mathbf{CST}_i = \mathbf{CT}_i \mathbf{C}_i \quad (i = 1, 2) \quad (7)$$

$$\mathbf{AST}_i = \mathbf{AT}_i \mathbf{A}_i \quad (i = 1, 2) \quad (8)$$

We concatenate the vector **EG_i**, which encoded by GRU, and the vector **EST_i**, and then input their result to the fully connected layer using the activation functions *tanh* and *sigmoid* to obtain the weighted vectors **EFT_i** and **EGt_i**, respectively, as follows. Using the similar equations, we can also get the weighted vectors **CFt_i** and **CGt_i** from **CG_i** and **CSt_i**, and get the weighted vectors **AFT_i** and **AGt_i** from **AG_i** and **AST_i**, respectively.

$$\mathbf{EFT}_i = \tanh(\mathbf{W}^f[\mathbf{EG}_i, \mathbf{EST}_i]) \quad (i = 1, 2) \quad (9)$$

$$\mathbf{EGt}_i = \text{sigmoid}(\mathbf{W}^g[\mathbf{EG}_i, \mathbf{EST}_i]) \quad (i = 1, 2) \quad (10)$$

where \mathbf{W}^f and \mathbf{W}^g are the parameter matrix and vector, respectively.

We adopt the gating mechanism to obtain the event-related information flow **EUt_i** from the event mention as follows. Using the similar equations, we can also get the event-related information flow **CUT_i** from the context of the trigger and get the event-related information flow **AUT_i** from the argument list.

$$\mathbf{EUt}_i = \mathbf{EGt}_i \mathbf{EFT}_i + (1 - \mathbf{EGt}_i) \mathbf{EG}_i \quad (i = 1, 2) \quad (11)$$

Finally, we use the global maximum pooling to get the final event mention vector **EX_i**, the context vector **CX_i** and the argument list vector **AX_i** as follows.

$$\mathbf{EX}_i = \text{GlobalMax}(\mathbf{EUt}_i) \quad (i = 1, 2) \quad (12)$$

$$\mathbf{CX}_i = \text{GlobalMax}(\mathbf{CUT}_i) \quad (i = 1, 2) \quad (13)$$

$$\mathbf{AX}_i = \text{GlobalMax}(\mathbf{AUT}_i) \quad (i = 1, 2) \quad (14)$$

D. Similarity Calculation

The above operations only processed each event mention itself and extracted its features. The principle of event coreference resolution is to measure the semantic similarity between an event mention pair. Hence, we must combine the information derived from an event mention pair to explore their relationship.

We use three distances to calculate the similarity between two event mentions. The Cosine distance ($Cos()$) calculates the angle between two vectors to measure the degree of similarity. Bilinear distance ($Binlinear()$) is a simple way to incorporate the linear interactions between two vectors. Single Layer Network ($SLN()$) is to calculate the nonlinear distance between two vectors and we choose $tanh$ as the activation function. Because of the existence of nonlinear activation functions, the Single Layer Network can capture nonlinear interactions between two event mentions. Since Bilinear distance focuses on capturing linear interactions and Single Layer Network focuses on capturing non-linear interactions, Single Layer Network can make up for the lack of Bilinear to some extent.

With Cosine distance, Bilinear distance and Single Layer Network, we measure the similarities and capture the linear and non-linear interactions between two event mentions as follows. Finally, we obtain the Cosine distance **CD**, the Bilinear distance **BD** and Single Layer Network distance **SD**.

$$CD = \text{Cos}(EX_1, EX_2) = \frac{EX_1^T EX_2}{\|EX_1\| \|EX_2\|} \quad (15)$$

$$BD = \text{Binlinear}(EX_1, EX_2) = EX_1^T W EX_2 \quad (16)$$

$$SD = SLN(EX_1, EX_2) = \text{Tanh}(W[EX_1, EX_2] + b) \quad (17)$$

where **W** and **b** are the parameter matrix and vector, respectively.

E. Output

We merge the vectors using the global maximum pooling and three distances as follows.

$$Q = \text{Concat}(EX_1, EX_2, CX_1, CX_2, AX_1, AX_2, CD, BD, SD) \quad (18)$$

Then we merge the vector **Q** with the pairwise feature vector **P** to get the combined vector **V_f** as follows.

$$V_f = \text{Concat}(Q, P) \quad (19)$$

The vector **V_f** is placed in a fully connected classifier which uses the activation function *Relu* as follows

$$V_h = \text{Relu}(W^h V_f + b) \quad (20)$$

We get the confidence by the function *sigmoid* as follows.

$$score = \text{sigmoid}(W^0 V_h + b_0) \quad (21)$$

We use the dropout in the fully connected layer to prevent over-fitting, which also improves robustness of the model, making the model easy to converge. Finally, following Fang, we use a closure to form an event chain that is determined as coreferential by the above classifier.

IV. EXPERIMENTS

In this section, we first introduce the dataset and experimental settings. Then, we report and analyze the experimental results on the ACE 2005 Chinese corpus.

A. Experimental Setup

The ACE 2005 Chinese corpus is the only available dataset for the task of document-level Chinese event coreference resolution. Since a portion of the documents in this corpus do not have coreferential event pairs, following Teng, we remove these documents from the corpus and obtain a new corpus containing 445 documents. We enumerate all event mention pairs of any two event mentions in the same document, excluding those event pairs with different event subtypes. Finally, a total of 14394 event mentions pairs are extracted and the positive-negative ratio is about 1:5.

For fair comparison, we perform 5-fold cross-validation on the ACE corpus. In each turn, the whole corpus is divided into training set, validation set and test set on the ratio of 3:1:1. MUC [15], B³ [16], CEAFe [17] and BLANC [18] are used to evaluate the performance of event coreference resolution. Among them, MUC is the most important metric and it is a score based on event links. B³ is a score based on event nodes, which makes up for the MUC's ignorance of the evaluation of non-coreferential events. CEAFe is similar to B³, but it adds entities to evaluate the performance of event coreference resolution. BLANC measures the average performance between non-coreferential events and coreferential events. The evaluation of the above four metrics can comprehensively reflect the model performance in event coreference resolution. Following the previous work, we also use the average score (AVG) of the above four metrics as comparison metric.

Follow the previous work [19], we pre-trained the word embeddings with *Word2Vec* on the Wikipedia Chinese corpus. The dimension of word vector is set to 300 and the dimension of the position vector is set to 50. To prevent overfitting, the value of Dropout is set to 0.2. Besides, GRU neurons are set to 100 dimensions, and Bi-GRU neurons are set to 50 dimensions. Finally, the model training round is set to 20.

B. Experimental Results

To verify the effectiveness of our model GANN, we introduce three state-of-the-art systems as baselines: 1) **Teng**: the only available system on document-level Chinese event coreference resolution using a traditional model; 2) **Krause**: a CNN model; and 3) **Fang**: a multiple decomposable attention network model. Since Krause and Fang are two systems on English, we obtain their codes and modify their system to fit Chinese. Table 1 shows the performance comparison of four systems on the ACE 2005 Chinese corpus.

TABLE I. PERFORMANCE COMPARISON OF FOUR MODELS

System	MUC	B ³	BLANC	CEAFe	AVG
Teng	73.5	/	/	/	/
Krause	66.52	86.08	75.31	77.34	76.31
Fang	69.87	87.77	77.86	80.53	79
GANN	74.59	89.18	80.32	82.56	81.66

From Table 1, we can find out that:

1) Compared with the traditional model Teng, our GANN improves the metric MUC by 1.09. Teng used the external tools to extract a large number of features, and we only use a few features provided the corpus. This result verifies that the neural network models are more suitable for Chinese event coreference resolution than the traditional models.

2) Compared with the CNN model Krause, our GANN gains an improvement of 5.35 in AVG with all improvements in four metrics. The reason is that Krause only performed convolution operations to extract the local information in event mentions, and they did not reweight the important information in event mentions. Moreover, they also did not consider the relationships between an event mention pair. Our GANN not only introduces the gated attention mechanism to reweight and select event-related information, but also calculates the semantic similarity between an event mention pair on three distances. This also illustrates the usefulness of our gated attention mechanism and three distances.

3) Compared with another neural network model Fang, our GANN also outperforms it on all metrics from 1.41 to 4.72. Although Fang used the attention mechanism, but they did not filter the noisy information. Besides, Fang only used the Cosine distance to obtain the linear relation between an event mention pair, while GANN uses three distances to capture the linear and nonlinear relations.

Among all metrics, the largest improvement of GANN comes from MUC. The metrics MUC is the most important metric in event coreference resolution and it scores on the edges of undirected graphs, which usually related to the coreferential events. This result derives from the ability of the gated attention mechanism to reweight and select event-related information from event mentions. Hence, GANN can identify more positive examples in which have all kinds of noisy information.

C. Results Analysis

To analyze the effectiveness of each component in GANN, we implement four simplified versions for comparison. The results are showed in Figure 2, where 1) -GA is a model whose gated attention mechanism is replaced by Bi-GRU and CNN, 2) -B&S is a model only using Cosine distance, 3) -C&S is a model only using Bilinear distance, and 4) -C&B is a model only using Single Layer Network.

TABLE II. COMPARISON OF GANN AND ITS FOUR SIMPLIFIED VERSIONS.

System	MUC	B ³	BLANC	CEAF _e	AVG
GANN	74.59	89.18	80.32	82.56	81.66
-GA	-5.53	-1.12	-2.61	-1.62	-2.72
-B&S	-3.73	-1.18	-2.73	-1.56	-2.3
-C&S	-4.38	-1.2	-2.29	-1.73	-1.11
-C&B	-3.27	-1.13	-2.52	-1.6	-0.7

Compared with GANN, the simplified version -GA reduces the AVG by 2.72 and this further ensures that the gated attention mechanism is helpful to resolve document-level Chinese event coreference.

The event-related information is often valuable for event coreference resolution. The attention mechanism plays a core role in GANN, which can reweight the features

and then highlight the event-related information. Moreover, the gating mechanism can further filter event-independent information to enhance the representation of event mention. On the contrary, the Bi-GRU and CNN cannot enlarge the difference between event-related and event-independent information, and some noisy information will affect the discrimination of the model. Take the following two event mentions as examples.

S5: 号称是最先进、设备最齐全的台北市 119 勤务中心今天正式落成启用。(119 service center of Taipei, which is the most advanced and fully equipped center, was officially **opened** today.)

S6: 重新改建的勤务中心是在上午落成启用。(The rebuilt service center was **opened** in the morning.)

The gated attention mechanism can extract the event-related information both in S5 and S6 as “勤务中心落成启用”(the service center was opened) and ignore the rest noisy information. Another advantage of the gated attention mechanism is that it can reduce computational complexity and enhance the discriminative accuracy.

When we introduce only one distance to GANN, the results of -C&B, -C&S and -B&S in Figure 2 show that the combination of three linear and nonlinear distances can improve the performance of GANN. Different from the results in English event coreference resolution where Cosine distance plays a core role, nonlinear distance Single Layer Network is more suitable for Chinese event coreference resolution for its low performance drop (AVG: 0.7 vs 2.3). The reason derives from the characteristics of Chinese language in which an event has flexible and diverse expressions. The Cosine and Bilinear distances consider the semantic similarity between two event mentions and they do not consider the influence of a single word on the distance. On the contrary, nonlinear distance Single Layer Network fully considers the influence of a single word on the distance, and is suitable for Chinese for its flexible sentence structure. Furthermore, the results in Figure 2 also show that the combination of three distances can be complementary. If the three distances are used together, it is possible to comprehensively consider the relationships and interactions between event mentions in a multi-faceted manner, which is more helpful for the event coreference resolution.

The disadvantage of our GANN is that it cannot resolve the coreferential event mentions which have different words and structures. Currently, almost all coreference resolution models are similarity model, which relies heavily on the similarity between two event mentions. GANN is also a similarity model and it has the common problem in those similarity models. Take the following two event mentions as examples.

S7: 凌晨 2 点南投分局出动了 90 名警力，一行人浩浩荡荡来到南投司法大厦前吴薇婉的静坐处，在经过分局长沟通说明之后，进行强制拆除。

(At 2 a.m, Nantou Branch dispatched 90 police officers. They went to the front of the Nantou Judicial Building where is the place of the sit-in **protest** for Wu Weiwan. After the director communicated with Wu, they removed all items by force.)

S8: 今天早上她带着将近 30 多位的支持群众向警方抗议。

(This morning, she brought nearly 30 supporters to protest to the police.)

Since the words in event mentions S7 and S8 are almost completely different, this lead to the low similarity between the above two mentions, calculated by GANN. However, if the model can understand the context information according to the full text, it can understand that they are coreferential.

In this paper, we uses a few annotating features (e.g., arguments, tense and modality) to build a strong model. However, annotating these event information are time-consuming and laborious. Hence, how to resolve event coreference from less annotating features is a challenging task and it can be applied to real world applications. To make our neural network model more persuasive, we also evaluate our model on annotated event mentions, event types and triggers, but not use any other annotated information. We compare our GANN with the baseline Fang (the highest performance model in all baselines) and the results are shown in Table 3.

TABLE III. EVALUATION ON LESS ANNOTATED FEATURES

System	MUC	B ³	BLANC	CEAF _e	AVG
GANN	56.45	82.17	67.4	71.77	69.45
Fang	-7.67	+1.37	-1.4	+1.59	-1.53

Table 3 shows that our GANN outperforms Fang by 1.53 in AVG, especially the largest improvement in MUC (7.67). This result also can further verify the effectiveness of GANN for the task of document-level Chinese event coreference resolution. Besides, we can also find that AVG drops from 81.66 to 69.45 (-12.21) without the annotated entity and event attributes.

V. CONCLUSIONS

In this paper, we introduce a novel GANN model to the task of document-level Chinese event coreference resolution. Firstly, GANN introduces a gated attention mechanism to select the event-related information from event mentions and then filter noisy information. Secondly, GANN not only uses a single Cosine distance to calculate the linear distance between two event mentions, but also introduces multi-mechanisms, i.e., Bilinear distance and Single Layer Network, to further calculate the linear and nonlinear distance. Hence, linear distance and nonlinear distance can complement each other. The experimental results on the ACE 2005 Chinese corpus illustrate that our model GANN outperforms the state-of-the-art baselines. In the future, we

will focus on the end-to-end and cross-document event coreference resolution.

REFERENCES

- [1] M. Fouad and M. Atyah, “Efficient Topic Detection System for Online Arabic News,” Proc. IJCA, 2018, pp. 7-12.
- [2] L. Cheng, H. Gao and H. Wang, “A News Event Extraction Method in Chinese and Thai Languages Based on Dependency Tree Elements Combined with Rules,” Software Guide, 2018.
- [3] S. Swayamdipta, A. Parikh and Tom Kwiatkowski, “Multi-Mention Learning for Reading Comprehension with Neural Cascades,” Proc. ICLR, 2018.
- [4] S. Krause, F. Xu, H. Uszkoreit and D. Weissenborn, “Event Linking with Sentential Features from Convolutional Neural Networks,” Proc. CoNLL, 2016, pp. 239–249.
- [5] J. Fang, G. Zhou and P. Li, “Employing Multiple Decomposable Attention Networks to Resolve Event Coreference,” Proc. NLPCC, 2018, pp. 246-256.
- [6] A. Haghghi and D. Klein, “Simple coreference resolution with rich syntactic and semantic features,” Proc. EMNLP, 2009, pp. 1152-1161.
- [7] B. Yang, C. Cardie and P. Frazier, “A Hierarchical Distance-dependent Bayesian Model for Event Coreference Resolution,” Computer Science, 2015.
- [8] Z. Chen and H. Ji, “Graph-based event coreference resolution,” Proc. ACL, 2009, pp. 54–57.
- [9] Z. Chen, H. Ji and R. Haralick, “A pairwise event coreference model, feature impact and evaluation for event coreference resolution,” Proc. ACL, 2009, pp. 17-22.
- [10] Z. Liu, J. Araki, E. Hovy and T. Mitamura, “Supervised with-document-level event coreference using information propagation,” Proc. LREC, 2014, pp. 4539–4544.
- [11] J. Lu and V. Ng, “Joint Learning for Event Coreference Resolution,” Proc. ACL, 2017, pp. 90–101.
- [12] Z. Liu, T. Mitamura and E. Hovy, “Graph-Based Decoding for Event Sequencing and Coreference Resolution,” Computational Natural Language Learning, 2018.
- [13] J. Teng, P. Li and Q. Zhu, “Global Inference for Co-reference Resolution between Chinese Events,” Acta Scientiarum Naturalium Universitatis Pekinensis, pp. 97-103.
- [14] A. Parikh, O. Tackstrom and J. Uszkoreit, “A Decomposable Attention Model for Natural Language Inference,” Proc. EMNLP, 2016, pp. 2249–2255.
- [15] M. Vilain, J. Burger and J. Aberdeen. “A Model-Theoretic Coreference Scoring Scheme,” Proc. ACL, 1995, pp. 45-52.
- [16] A. Bagga and B. Baldwin, “Algorithms for scoring coreference chains,” Proc. LREC, 1998, pp. 563–566.
- [17] X. Luo, “On coreference resolution performance metrics,” Proc. EMNLP, 2005, pp. 25-32.
- [18] M. Recasens and E. Hovy, “BLANC: Implementing the Rand Index for Coreference Evaluation,” Proc. NLE, 2011, pp. 485-510.
- [19] S. Xu, P. Li, G. Zhou and Q. Zhu, “Employing Text Matching Network to Recognise Nuclearity in Chinese Discourse,” Proc. EMNLP, 2018, pp. 525-535.

A Multi-stage Strategy for Chinese Discourse Tree Construction

Tishuang Wang, Peifeng Li, Qiaoming Zhu

Natural Language Processing Lab, School of Computer Science and Technology
Soochow University
Suzhou, China
tswang@stu.suda.edu.cn, {pfli, qmzhu}@suda.edu.cn

Abstract—Building discourse tree is crucial to improve the performance of discourse parsing. There are two issues in previous work on discourse tree construction, i.e., the error accumulation and the influence of connectives in transition-based algorithms. To address above issues, this paper proposes a tensor-based neural network with the multi-stage strategy and connective deletion mechanism. Experimental results on both CDTB and RST-DT show that our model achieves the state-of-the-art performance.

Keywords-discourse parsing; tree construction; chinese discourse treebank

I. INTRODUCTION

According to the Rhetorical Structure Theory (RST) [1], discourse usually composed of a series of Element Discourse Units (EDUs, e.g., words, phrases, sentences or paragraphs), which is an organized, hierarchical whole. Discourse parsing aims to identify the structures and relationships with semantic connection and combines adjacent EDUs with rhetorical relations in a hierarchical way to represent an entire document as a discourse tree. As a subtask of discourse parsing, discourse tree construction can assist to analyze and understand the information of discourse, and it is widely used in many down-stream NLP tasks, such as information extraction [2], summarization [3], and question answering [4].

Discourse (structure) tree construction is to recursively connect EDUs by rhetorical relation to larger text spans until the final tree is built. To make a clearer explanation of the discourse tree, take the chtb_0013 as an example, which is a typical news article from Chinese Treebank 8.0 [5].

Example1: 大运河作为一条水运大动脉，为沿岸企业提供了运输和给排水之便 a ，成为企业发展的生命通道 b 。据江苏苏钢集团公司负责人介绍，苏钢每年要靠大运河运输原料、成品一百五十万吨 c 。其中煤炭从徐州运来，走运河比走陆路运费每吨便宜十五元 d ，仅此一项大运河每年就为厂家节约成本三千多万元 e 。运河整治后，苏钢在运河对岸建了新厂区 f ，并自筹资金建设了一座跨运河大桥 g ，把新厂、老厂连为一体 h 。As the main artery of water transport, the Grand Canal provides transportation and water supply and drainage for coastal enterprises a , become the life channel of enterprise development b . According to the person in charge of Jiangsu Sugang Group Co., Ltd., Sugang relies on the Grand Canal to transport raw materials and finished products up to 1.5 million tons each year c . Among them, coal is transported from Xuzhou, and it's 15 yuan cheaper per ton than land freight by canal d . Only this one, Grand Canal saves more

than 20 million yuan for manufacturers each year e . After the canal was rehabilitated, Sugang built a new factory on the opposite side of the canal f , and they raised funds to build a bridge across the canal g , Connect the new factory and the old factory h .

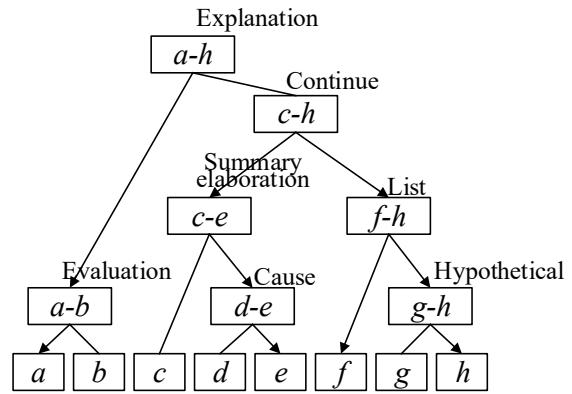


Figure1. Discourse tree of chtb_0013

This whole discourse tree with 15 spans is shown as Figure 1 and there are eight EDUs ($a-h$). The leaf of this tree is EDU and the internal node is relational node. The arrow points to the important span. In this paper, we only focus on how to construct the naked Chinese discourse tree, ignoring the relations (e.g., *Evaluation*) and the nuclearity (e.g., *Nucleus* and *Satellite*).

Although there are many studies on discourse parsing due to its vital role in NLP, only a few address discourse tree construction [6, 7, 8, 9, 10, 11]. Among them, only two studies [10, 11] explored discourse tree construction in Chinese due to the lack of annotated corpus and the abstract nature of Chinese itself. In addition, those studies heavily relied on manual feature engineering.

Those transition-based algorithms are widely used in discourse tree construction and there are two issues in Chinese discourse tree construction using shift-reduce algorithm. The first is the error accumulation. The error of the previous prediction in the shift-reduce algorithm will directly lead to the deviation of the subsequent prediction results. We found that the performance of discourse tree construction drops quickly with the increase of the EDU number. The second is the influence of connectives. The connectives have a greater impact on action prediction in the shift-reduce algorithm. It tends to connect two EDUs to a new span when there is a connective between them. For example, the connective (并 and) between two spans f and $g-h$ in example1 will make the algorithm connect them to a span.

In this paper, to solve the above two issues in Chinese discourse tree construction, we propose a tensor-based neural network with the multi-stage strategy and connective deletion mechanism. Experimental results on CDTB, a Chinese discourse corpus, show that our model achieves the state-of-the-art performance.

II. RELATED WORK

Most of previous work focused on constructing English discourse tree. The algorithms of discourse tree construction on Rhetorical Structure Theory Discourse Treebank (RST-DT) [12], a English corpus, can mainly be categorized as shift-reduce algorithms [6, 7, 13], probabilistic CKY-like algorithms [8, 14, 15] and greedy bottom-up algorithms [9]. Wang et al. [7] used a transition-based system to build discourse trees with nuclearity labels and then used Support Vector Machines (SVM) to determine the discourse relation at different text levels. Joty et al. [8] used the sequence labeling instead of classification, and used Dynamic Conditional Random Field (DCRF) model combining structure recognition with relationship recognition. Li et al. [15] proposed an attention-based hierarchical Bi-LSTM network to learn the representations of the text spans and used a tensor-based transformation function to capture interactions among the features of the text spans. Feng et al. [9] adopts a greedy bottom-up approach, with two linear-chain CRFs applied in cascades as local classifiers. To enhance the accuracy of the pipe line, they add additional constraints in the Viterbi decoding of the first CRF, and used the novel approach of post-editing, which modifies a fully-built tree by considering information from constituents on the upper levels.

As for Chinese discourse tree construction, there are only two studies [10,11] on the Chinese Discourse Treebank (CDTB) [16]. Kong and Zhou [10] proposed a CDT-styled End-to-End discourse parser, which can automatically detect discourse units in a free text, generates the discourse parse tree in a bottom-up way, and determines the sense and centering attributions for all nonterminal nodes by traversing the discourse parse tree. Sun and Kong [11] used shift-reduce algorithm and then used Convolutional Neural Networks (CNN) with different convolution windows, proposed a complete Chinese discourse structure generating framework which can be used to generate the tree-like structure from plain texts. Furthermore, those studies heavily relied on manual feature engineering.

III. CONSTRUCTING CHINESE DISCOURSE TREE WITH MULTI-STAGE STRATEGY AND CONNECTIVE DELETION MECHANISM

In this section, we first introduce the basic model, a tensor-based neural network, to construct Chinese discourse tree, and then apply the multi-stage strategy and connective deletion mechanism to further help the basic model to improve the performance.

A. Basic Model

We also use transition-based (shift-reduce) algorithm to build the Chinese discourse trees and this process is modeled as a sequence of shift and reduce action with a stack and a queue. The stack is initialized to be empty and the queue contains all EDUs in a document. At each step, a tensor-based neural network is to perform either shift or

reduce. The action shift pushes the first EDU in the queue on the top of the stack, while the reduce action pops and merges the top elements in the stack to get a new subtree, which is then pushed back to the top of the stack. Finally, a discourse tree can be constructed until the queue is empty and the stack contains only the root node of the discourse tree.

For the two elements DU_{s2}, DU_{s1} at the top of the stack and the first element DU_{q1} in the queue, a tensor-based neural network, as showed in Figure 2, is introduced to judge whether the relationship between DU_{s1} and DU_{s2} is closer or more closely related to DU_{q1} . If it is closer to DU_{s2} , the reduce action is performed; otherwise, the shift action is performed. Our tensor-based neural network consists of three parts: 1) Input and Coding; 2) Tensor-based Matching; and 3) Classification.

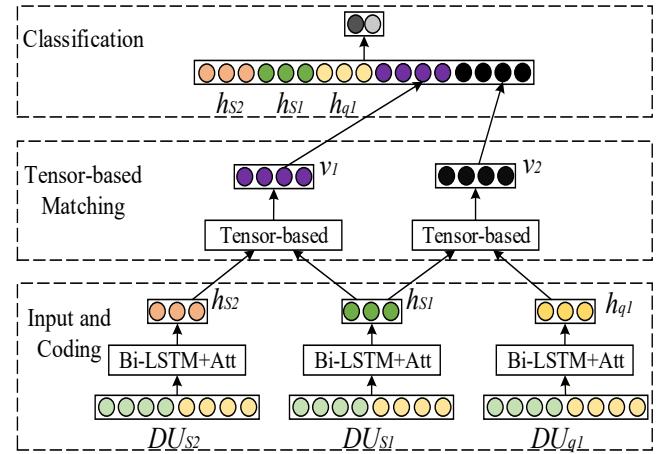


Figure 2. The tensor-based neural network model

Input and Coding

In the module Input and Coding, the input are the word sequences and POSs (Part-Of-Speeches) of the first two elements DU_{s1} and DU_{s2} in the top of the stack and the first element DU_{q1} in the queue, and then this module encodes the units DU_{s2} , DU_{s1} , and DU_{q1} using Bi-LSTM and attention mechanism to obtain their semantic vectors h_{s1} , h_{s2} and h_{q1} .

We combine the last outputs of the Bi-LSTM to be $h_s = [\overrightarrow{h_{e_n}}, \overleftarrow{h_{e_1}}]$. We also combine the outputs of the Bi-LSTM at every step: $h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$ and thus get a matrix $H = [h_1; h_2; \dots; h_n]$. Taking H and h_s as inputs, we get a vector α standing for weights and use it to get representation of the DU r :

$$M = \tanh(\mathbf{W}_y H + \mathbf{W}_l h_s \otimes e_n) \quad (1)$$

$$\alpha = \text{softmax}(\mathbf{W}_\alpha^T M) \quad (2)$$

$$r = H\alpha \quad (3)$$

where \otimes denotes Cartesian product, e_n is a n dimensional vector of all 1s, and $\mathbf{W}_y, \mathbf{W}_l, \mathbf{W}_\alpha$ are parameters.

Finally, we synthesize the information of r and h_s to get the final representation of the DU:

$$w_h = \sigma(\mathbf{W}_{hr} r + \mathbf{W}_{hh} h_s) \quad (4)$$

$$h = w_h \odot h_s + (1 - w_h) \odot r \quad (5)$$

where \odot denotes Hadamard product, $\mathbf{W}_{hr}, \mathbf{W}_{hh}$ are parameters, w_h is the representation of weight vector calculated by r and h_s . h is the final representation of the DU by the Bi-LSTM and Attention.

Tensor-based Matching

Based on the discourse unit DU_{s1} , DU_{s2} and DU_{q1} , we obtain the semantic representation vectors h_{s1}, h_{s2} and h_{q1} from the Input and Coding module. Then these semantic representations are fed into the Tensor-based Matching module, which uses tensor-based transformation function [15] to incorporate the interaction between DU_{s1} and DU_{s2} and the interaction between DU_{s1} and DU_{q1} .

$$v_1 = \text{Relu}(\mathbf{W}_h[h_{s1}, h_{s2}] + [h_{s1}, h_{s2}]^T \mathbf{P}_h \mathbf{Q}_h [h_{s1}, h_{s2}] + \mathbf{b}_h) \quad (6)$$

$$v_2 = \text{Relu}(\mathbf{W}_h[h_{s1}, h_{q1}] + [h_{s1}, h_{q1}]^T \mathbf{P}_h \mathbf{Q}_h [h_{s1}, h_{q1}] + \mathbf{b}_h) \quad (7)$$

where $\mathbf{W}_h, \mathbf{P}_h, \mathbf{Q}_h, \mathbf{b}_h$ are parameters to incorporate the interaction between DU_{s1} and DU_{s2} and the interaction between DU_{s1} and DU_{q1} , and we choose *Relu* as the activation function.

Classification

Finally, three semantic vectors (h_{s1}, h_{s2} and h_{q1}) and two interactive vectors (v_1 and v_2) are combined and then sent to the output layer, i.e., the Classification module, through a nonlinear transformation.

$$c = [h_{s1}, h_{s2}, h_{q1}, v_1, v_2] \quad (8)$$

$$y = \text{softmax}(\mathbf{W}_y c + \mathbf{b}_y) \quad (9)$$

where \mathbf{W}_y and \mathbf{b}_y are the parameters. During the training, we use the Adam optimizer to optimize the network parameters by maximizing the log-likelihood loss function.

B. Multi-stage Strategy

The shift-reduce algorithm will cause error accumulation, and the error of the previous prediction will directly lead to the deviation of the subsequent prediction results. We found that the performance of discourse tree construction drops quickly with the increase of the EDU number. To reduce the cascading errors in shift-reduce algorithm, we propose a multi-stage strategy to construct discourse tree, as shown in Figure 3. It first constructs the sentence-level subtree, and then constructs the paragraph-level subtree based on its sentence-level subtrees. Finally, it constructs the document-level tree on its paragraph-level subtrees if this document has more than one paragraph.

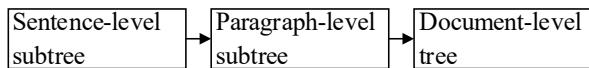


Figure 3. Multi-stage in discourse tree construction.

The article in Figure 1 has four sentences, as showed in Figure 4(a) (each dashed box contains a sentence). It is easy to understand that each sentence can form a subtree in most cases. Therefore, we first construct four sentence-level subtrees for each sentence, as showed in Figure 4(a). Then

we construct the paragraph-level subtree based on four sentence-level subtrees, as showed in Figure 4(b). If this article has more than one paragraph, we will continue to construct its document-level tree.

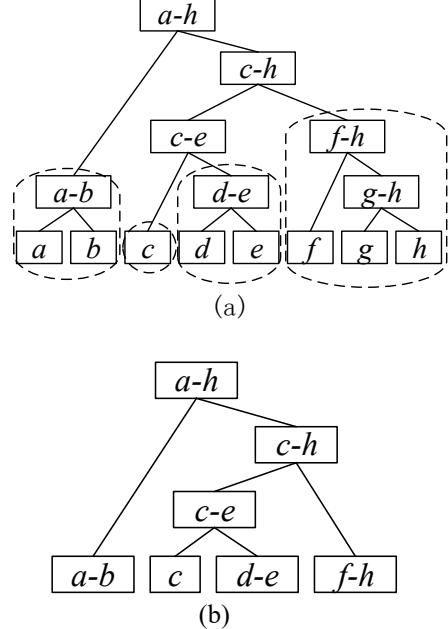


Figure 4. Two-stage to construct a discourse tree.

Feng and Hirst [9] also proposed a two-stage (sentence-level and document-level) bottom-up, greedy parser with linear-chain CRF models. However, their model used the sliding window to get contextual features and heavily relied on manual features engineering. Different from them, our model uses the shift-reduce algorithm to build discourse tree and uses the tensor-based neural network to classify actions. Besides, we do not use any additional manual features.

C. Connective Deletion

Connectives have a greater impact on action prediction in the shift-reduce algorithm. For example in *chtb_0013*, there is a connective (并 and) between two spans *f* and *g-h*. The current state is that the two elements in the stack are *f* and *g* and the first element in the queue is *h*. Our model is more inclined to connect *f* and *g* due to the above connective between them.

There are two types of connective in our model, intra-sentence connectives and inter-sentence connectives. Our multi-stage strategy can solve the impact of inter-sentence connective because each sentence must be a subtree, as the connective (其中 Among them) in *chtb_0013*, our multi-stage strategy to construct discourse tree will builds *c* and *d-e* into a subtree respectively, at this time, the connective (其中 Among them) is connected to the subtrees of *c* and *d-e*, and then our multi-stage strategy to construct discourse tree avoid connecting *c* and *d* directly. However, the intra-sentence connectives still have an adverse effect on constructing sentence-level subtree, as the connective (并 and) in *chtb_0013* connecting is still *f* and *g*. Therefore, we use a simple mechanism to solve this issue, i.e., deleting the intra-sentence connectives and retaining the inter-sentence connectives.

IV. EXPERIMENTS

In this section, we first introduce the dataset and experimental setting, then report the experimental results on CDTB. Finally, we also evaluate our model on the English corpus RST-DT.

A. Experimental Setup

We conducted our experiments on the Chinese Discourse TreeBank (CDTB) [16]. This corpus is built on the Chinese Treebank (CTB) [5] with a connective-driven dependency tree scheme. Each paragraph in CDTB is marked as a tree and CDTB consists of 500 newswire articles, which are further divided into 2342 paragraphs with a tree representation for one paragraph. Hence, we only apply a two-stage strategy for CDTB because this corpus does not provide document-level tree. Besides, CDTB contains 10650 EDUs, and each EDU has 22 Chinese characters on average.

Following Sun and Kong [10], we choose 425 for training, 25 for development and 50 for testing. We follow Morey et al. [17] to report the micro-averaged and macro-averaged F1-scores. In addition, we also report the accuracy of the whole tree structure (Tree Acc). For comparison with previous studies, the experimental result returned on CDTB corpus does not contain leaf nodes of discourse tree, but on RST-DT the leaf nodes are included.

The dimension of the word embeddings is set to 300, and the dimension of the POS embeddings is set to 50. We pre-trained the word embeddings with Word2Vec on the Wikipedia Chinese corpus. For a fair comparison, all of the models in our experiments use the same parameters. The number of LSTM neurons is set to 150, and the number of Attention neurons is set to 50. We adopt the dropout strategy to avoid overfitting and set the dropout rate to 0.5.

B. Experimental Results

We compare our method with the following state-of-the-art Chinese baselines:

- **KZ17** [10]: used contextual features, lexical features and dependency tree features to build discourse tree by a maximum entropy (ME) classifier with a greedy bottom-up algorithm.
- **SK18** [11]: used shift-reduce algorithms and then used a stack-augmented parser-interpreter CNN model with the features of different size windows.
- **LLC16** [15]: A Bi-LSTM model with the attention mechanisms and the tensor-based transformation function and it used probabilistic CKY-like algorithms.

TABLE I. OVERALL PERFORMANCE IN CDTB

Model	Macro-F1	Micro-F1
LLC16	74.8	60.8
KZ17	67.3	57.1
SK18	84.0	-
Ours	86.8	79.9

Table 1 shows the performance of our model and three baselines on CDTB and it illustrates that our model outperforms all the others significantly both on the micro-averaged and macro-averaged F1-score. This result verifies

that our tensor-based neural network and two-stage strategy are beneficial to Chinese discourse tree construction. Our Tensor-based Matching module is similar to LLC16, and the improvement of our model shows that our tensor-based neural network and two-stage strategy are more suitable for Chinese discourse tree construction.

To further explore the different influence of our two-stage and the mechanism of connective deletion in our model, we implement three simplified versions (i.e., *Simp1/2/3*) as show in table 2 where *Stage* means whether the two-stage strategy is adopted and *Del* represents whether the intra-sentence connective deletion mechanism is used.

TABLE II. COMPARISON WITH SIMPLIFIED VERSIONS

Model	Stage	Del	Macro-F1	Micro-F1	Tree Acc
Simp1	No	No	74.5	59.6	44.5
Simp2	No	Yes	77.3	64.0	48.5
Simp3	Yes	No	85.5	77.9	58.1
Ours	Yes	Yes	86.8	79.9	60.3

The simplest model Simp1 is like to Li et al. [15] except that we use the shift-reduce algorithms. Comparing with Simp1, we can find that Simp2 with the connective deletion mechanism achieves the improvements on all three metrics. This result ensures that this mechanism is helpful for discourse tree construction. We can also observe that Simp3 with the two-stage strategy improves the micro-averaged/macro-averaged F1-score and the whole tree accuracy by 11, 18.3 and 13.6, respectively. This result verifies that our two-stage strategy can reduce the complexity of discourse tree construction.

Table 3 shows the performance comparison of one-stage and two-stage strategy both on sentence-level subtree and paragraph-level subtree performance. We can find that our two-stage strategy can improve the Micro-F1 scores of sentence-level and paragraph-level subtree construction simultaneously.

TABLE III. COMPARISON OF THE EFFECT OF TWO-STAGE STRATEGY ON DIFFERENT LEVEL SUBTREE (MICRO-F1)

	one-stage	two-stage
sentence-level	60.7	80.8
paragraph-level	72.2	78.3

Table 4 shows the effectiveness of connective deletion mechanism on those explicit and implicit nodes, in which the explicit nodes mean that the adjacent DUs contain connectives. We can find that the connective deletion mechanism improves the Micro-F1 scores of the explicit and implicit nodes by 2.4% and 1.8%, respectively.

TABLE IV. THE EFFECTIVENESS OF CONNECTIVE DELETION MECHANISM ON EXPLICIT AND EXPLICIT NODES (MICRO-F1)

	w/o deletion	deletion
explicit	75.7	78.1
implicit	78.6	80.4

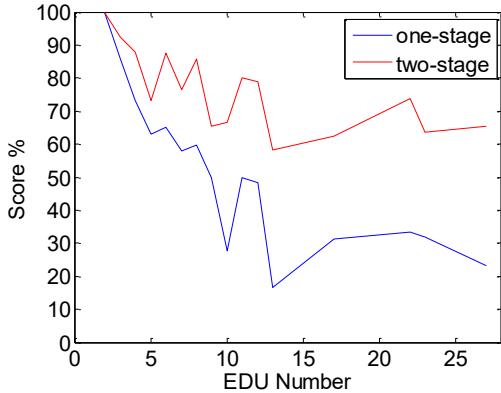


Figure 5. Micro-averaged F1 in different EDUs number.

Figure 5 shows the micro-averaged F1-scores with different EDUs numbers in a discourse tree. We can find that the micro-averaged F1-score of one-stage strategy drop significantly with the increase of the number of EDUs. However, this figure of the two-stage strategy is relatively stable. This result proves that our two-stage strategy is better than the one-stage strategy in all EDUs numbers.

We find that the construction performance of sentence-level subtree is similar to that of paragraph-level tree. The main reason is that the sentence-level subtree and paragraph-level tree are quite complex and they have a similar average number of EDUs (3.0 and 2.9). Besides, we can also find out that the sentence-level subtree and paragraph-level tree are relatively simple in CDTB for their small EDUs numbers.

C. Experimental Result on RST-DT

We also evaluate our model on an English corpus, RST-DT [12], and all parameter settings are the same as on CDTB. In CDTB, each paragraph forms one discourse tree and each paragraph contains 4.5 EDUs on average. In RST-DT, one document forms one discourse tree and each document contains 55.6 EDUs on average. Hence, these two corpora are different in many aspects. Since RST-DT provides a document-level discourse tree, a three-stage strategy is used in our model.

We use the same data split as in Li et al. [14], i.e., 312 for training, 30 for development and 38 for testing. Three state-of-the-art baselines are selected for comparison: 1) BSC17 [13]: a transition-based discourse parser using a feed-forward neural network and a shift-reduce algorithm; 2) LLC16 [15]: a Bi-LSTM model with the attention mechanisms and the tensor-based transformation function and it used probabilistic CKY-like algorithms; 3) FH14 [9]: a two stage (sentence-level and document-level) bottom-up, greedy parser with linear-chain CRF models. Table 5 shows the performance of three baselines and our model.

TABLE V. PERFORMANCE COMPARISON WITH STATE-OF-THE-ART DISCOURSE PARSERS.

Model	Macro-F1	Micro-F1
BSC17	85.1	81.3
LLC16	85.4	82.2
FH14	87.0	84.3
Ours	87.2	83.4

The preliminary experimental results show that our model achieves comparable performance to those state-of-the-art discourse parsers. Especially, our model outperforms the other two neural network models, a transition-based (shift-reduce algorithm) model (BSC17), and a tensor-based model (LLC16). However, compared with FH14, our model achieves the similar performance. It is worthy to note that FH14 uses many additional manual features, while we only use the sentence as the input.

Compared with CDTB, the structure of RST is more complex. The main reason is that RST annotates the structure between paragraphs. In addition, some sentence does not have a well-formed subtree, because some of its units attach to the left and some to the right. Vliet and Redeker [18] called these cases as ‘leaky’ boundaries. This situation accounts for 5% of RST-DT and only 0.1% of CDTB. Therefore, our multi-stage strategy for discourse tree construction works better on CDTB.

In CDTB, each paragraph forms one discourse tree and the EDU number of all discourse tree is less than 30. In RST-DT, one document forms one discourse tree, the largest discourse tree contains 304 EDUs. Figure 6 shows the micro-averaged F1 of discourse tree with less than 30 EDUs in CDTB and RST-DT. We can find that RST-DT performs better than CDTB for discourse tree with less than 30 EDUs, this is because the structure of English is clear and precise, and however, the expression in Chinese is freer. But due to the low performance of the discourse tree with more than 30 EDUs in English, the overall performance of CDTB is better than RST-DT.

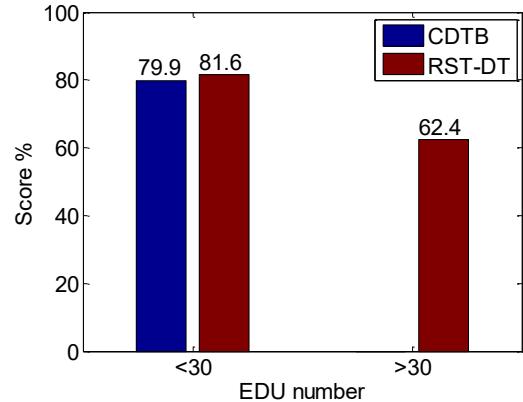


Figure 6. Micro-averaged F1 in different EDUs number on CDTB and RST-DT.

V. CONCLUSIONS

In this paper, we focus on two issues in naked Chinese discourse tree construction, the error accumulation and the influence of connectives in transition-based algorithms, and then propose a tensor-based neural networks with the multi-stage strategy and the connective deletion mechanism. Experimental results on both the Chinese and English corpora (CDTB and RST-DT) show that our model outperforms the state-of-the-art systems. In our future work, we will focus on how to provide more effective and language-independent methods to replace our simple mechanisms in this paper.

REFERENCES

- [1] W. Mann, S. Thompson, “Rhetorical structure theory: Toward a functional theory of text organization.” *Text-Interdisciplinary Journal for the Study of Discourse*, 1988, PP. 243-281.
- [2] B. W. Zou, G. D. Zhou, Q. M. Zhu, “Negation focus identification with contextual discourse information.” Proc. ACL 2014, pp. 522-532.
- [3] A. Cohan, N. Goharian, “Scientific document summarization via citation contextualization and scientific discourse.” *International Journal on Digital Libraries*, 2018, pp.187-303.
- [4] M. Liakata, S. Dobnik, S. Saha, “A discourse-driven content model for summarising scientific articles evaluated in a complex question answering task.” Proc. EMNLP 2013, pp. 747-757.
- [5] N. Xue, F. Xia, F. Chiou, M. Palmer, “The Penn Chinese Treebank: Phrase structure annotation of a large corpus.” *Natural Language Engineering*, 2005, pp. 207-238.
- [6] Y. Ji, J. Eisenstein, “Representation learning for text-level discourse parsing.” Proc. ACL 2014, pp. 13-24.
- [7] Y. Wang, S. Li, H. Wang, “A two-stage parsing method for text-level discourse analysis.” Proc. ACL 2017, pp. 184–188.
- [8] S. Joty, G. Carenini, R. Ng, and Y. Mehdad, “Combining intra- and multi-sentential rhetorical parsing for document-level discourse analysis.” Proc. ACL 2013, pp. 486–496.
- [9] V. W. Feng, G. Hirst, “Text-level discourse parsing with rich linguistic features.” Proc. ACL 2012, pp. 60–68.
- [10] F. Kong, G. D. Zhou, “A CDT-styled end-to-end Chinese discourse parser.” *ACM Transactions on Asian and Low-Resource Language Information Processing*, 2017, pp. 16-26.
- [11] C. Sun, F. Kong, “A Transition-based Framework for Chinese Discourse Structure Parsing.” *Journal of Chinese Information Processing*, 2018, pp. 48-56.
- [12] L. Carlson, D. Marcu, M. Okurowski, “Building a discourse-tagged corpus in the framework of rhetorical structure theory.” *Current and new directions in discourse and dialogue*, 2003, pp. 85-112.
- [13] C. Braud, M. Coavoux, A. Søgaard, “Cross-lingual rst discourse parsing.” Proc. EACL 2017, pp. 292–304.
- [14] J. Li, R. Li, E. Hovy, “Recursive deep models for discourse parsing.” Proc. EMNLP 2014, pp. 2061–2069.
- [15] Q. Li, T. Li, and B. Chang, “Discourse parsing with attention-based hierarchical neural networks.” Proc. EMNLP 2016, pp. 362–371.
- [16] Y. C. Li, F. Kong, and G. D. Zhou, “Building Chinese discourse corpus with connective-driven dependency tree structure.” Proc. EMNLP 2014, pp. 2105-2114.
- [17] M. Morey, P. Muller, N. Asher, “How much progress have we made on RST discourse parsing? A replication study of recent results on the RST-DT.” Proc. EMNLP 2017, pp. 1330–1335.
- [18] V. Nynke, D. Vliet, and R. Gisela. “Complex Sentences as Leaky Units in Discourse Parsing.” In *Proceedings of Constraints in Discourse*, Agay-Saint Raphael, 2011.

Articulatory Features Based TDNN Model for Spoken Language Recognition

Jiawei Yu, Minghao Guo, Yanlu Xie, Jinsong Zhang

Beijing Advanced Innovation Center for Language Resources

Beijing Language and Culture University

Beijing, China

vyujiawei@gmail.com, gmhgsmh8000@163.com, {xieyanlu, jinsong.zhang}@blcu.edu.cn

Abstract—In order to improve the performance of the Spoken Language Recognition (SLR) system, we propose an acoustic modeling framework in which the Time Delay Neural Network (TDNN) models long term dependencies between Articulatory Features (AFs). Several experiments were conducted on APSIPA 2017 Oriental Language Recognition(AP17-OLR) database. We compared the AFs based TDNN approach to the Deep Bottleneck (DBN) features based ivector and xvector systems, and the proposed approach provide a 23.10% and 12.87% relative improvement in Equal Error Rate (EER). These results indicate that the proposed approach is beneficial to the SLR task.

Keywords-spoken language recognition; articulatory features; time delay neural network; deep bottleneck features;

I. INTRODUCTION

Spoken Language Recognition (SLR) technology is to identify or verify the language being spoken in a speech utterance [1]. It can be used as the front-end of the multi-lingual speech recognition systems and the automatic translation systems [2], [3]. The state-of-the-art SLR systems can use a variety of features to distinguish one language from another, such as: acoustic features, prosody, phonotactic structure, lexical knowledge, vocabulary [1].

Generally, the SLR approach can be classified into two types according to features used: spectral-based and token-based. The spectral-based one exploit different distributions in the acoustic space between different languages. State-of-the-art modeling method, such as ivector and xvector, project acoustics of different languages to different places in high dimensional space. The token-based approaches utilize phonotactic information which characterizes how these phonemes are combined in a language. One of the examples is Phone Recognition followed by Language Modeling (PRLM) which converts a speech utterance into a sequence of phones by a phone recognizer, then uses an n-gram language model produces a likelihood score [1].

Compared to the token-based approach, the spectral-based one is weak at modeling the temporal information such as phonotactics for SLR task. On the contrary, the token-based approach can not accurately exploit the acoustic differences between languages. Especially, they heavily rely on the accuracies of recognizers [1] , which is usually hard for cross-language tasks.

In view of this, AFs were introduced to the SLR task [4], [5], [6]. The AFs represent the articulatory specification

in the vocal tract when pronouncing a phone. The combination of a few AFs can determine a specific phone. The finer granularity of AFs have, the better cross-language modeling power are got when compared to phonemes. So the recognition performance of AFs is generally better than phonemes [7], and consequently the AFs based SLR systems perform better. Besides, n-gram LM based on AFs is capable of modeling the phonotactics of different languages, and can bring about further improvements to SLR task. However, it still suffers from the data sparsity problem of n-gram, especially when contextual width is to be lengthened [1].

In this paper, we propose a scheme of AFs plus TDNN for SLR task. Our motivations include utilizing the advantage of cross-language modeling by AFs, and the effectiveness of TDNN in modeling temporal dependencies in the acoustic signal.

The rest of this paper is organized as follows. Section II presents the AFs based TDNN SLR system in detail. The experimental setup is presented in Section III. Finally Section IV and Section V show the experimental results and conclusion.

II. AFs BASED TDNN SLR SYSTEM

The AFs based TDNN SLR system diagram is shown in Fig. 1. The system include two part, the front-end is a feature extractor which processing spoken language utterances into sequence of AFs using ASR DNN model. Once we get the AFs, the TDNN back-end will classify these tokens to the specific language.

A. Articulatory Features

The International Phonetics Association (IPA) classifies the sounds of a language by means of AFs [8]. A sound is described by a bundle of articulatory features, and a unique symbol is used as a shorthand to represent this bundle. The AFs generally used to assist automatic speech recognition (ASR) [9], and several studies have proved that AFs can be recognized more robustly across languages than phonemes [7]. In the token-based and DBN based SLR approaches, the accuracy of phone recognizer is a critical factor. Specifically, if a phoneme of another language to be recognized is always recognized as the one in the phone set designed for the phone recognizer, it is fine to model it in the language model based on the assumption of similarity between them. If some phonemes are very different from the phonemes of the language for phone

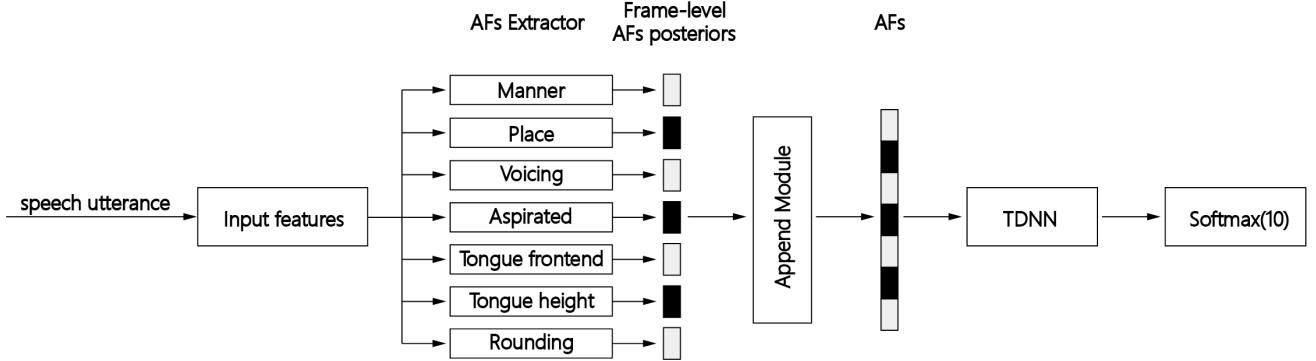


Figure 1. Block diagram of articulatory features based TDNN SLR system.

recognizer, they cannot be represented well in language modeling, which is quite common for spoken languages in different language families. So, we use AFs that are language-universal across all spoken languages as a front-end to obtain a more accurate recognizer and improve the performance of SLR system.

B. Articulatory Categories

In this paper, we choose 39 AFs, which belong to 7 categories listed in Table I according to the scheme of IPA. Each type of AFs having different items, e.g. “manner” has 6 items: affricate, fricative, nasal, liquid, sibilant and final labels. Except these AFs, We also use the “silence” token to represent the soundless segments.

TABLE I. OVERVIEW OF AFs TYPES USED

AFs	Numbers ^a	Description
Manner (MA)	9	<i>Manner of articulation</i>
Place (PA)	6	<i>Place of articulation</i>
Voicing (VO)	2	<i>Voicing</i>
Aspirated (AS)	3	<i>Aspiration</i>
Tongue frontend (TF)	8	<i>Tongue x position</i>
Tongue height (TH)	8	<i>Tongue y position</i>
Rounding (RO)	3	<i>Lips rounded</i>

a. The item number of each AF.

C. Articulatory Features Extraction

Since manual AF annotations of speech signals are rather difficult and costly to produce, one reasonable way of generating training material for the articulatory classifier is to convert phone-based training transcriptions to AFs transcriptions [10]. This can be achieved by using a canonically defined phone and AFs mapping table. In this study, we use Mandarin phone set converting AFs. Our mapping table is based on the [11], and we added three new mapping relationship between phone and AFs, as shown in Table II. In this paper, we used the posterior probabilities of the articulatory categories as the articulatory features. As shown in Fig. 1, the feature extraction module consists series of AFs’ extractors according to 7 attribute categories described in Table I. A context dependent DNN-based AFs’ extractor is separately built for each category. The current frame posteriors are linked to the possible class within that category. Subsequently, a

group of the frame attribute posteriors will be fed into the append module, the append module stacks together with the attribute posteriors and generates a vector, which is AFs.

TABLE II. AFs AND THEIR ASSOCIATED PHONES IN MANDARIN

AFs	Category	Phone set
Tongue Frontend (TF)	Front 2	<i>ii</i>
	Front 1	<i>iii</i>
	Front	<i>i v</i>
	Half F	
	Central	<i>a</i>
	Half B	
	Back	<i>u</i>
Tongue Height (TH)	High	<i>i ii iii v u</i>
	Second H	
	Half H	
	Middle	
	Half L	
	Second L	
Rounding (RO)	Low	<i>a</i>
	Rounded	<i>u v</i>
	Unrounded	<i>a i ii iii</i>

D. TDNN Back-end

The TDNN structure is shown in Fig. 2. The architecture of TDNN is designed to work on sequential data. Specifically, A TDNN is formulated as a feedforward network but it has delays on the layer weights associated with the input weights. The data are represented at different time points by adding a set of delays to the input. This allows the TDNN to have a finite dynamic response to time series input data [12], [13], [14].

In TDNN structure, a narrow temporal context is provided to the first layer and increasingly wide contexts are available to the subsequent hidden layers, i.e., each layer in a TDNN operates at a different temporal resolution, in this way, the higher layers of the network are able to learn longer temporal relationships.

To explain in more detail how TDNN learns long term temporal dependencies between AFs, we use the following example to illustrate. as shown in Fig. 2, Suppose t is current frame, at the input layer (layer1), frames [t-2,t+2] are spliced together. Layers 2, 3 and 4 we splice together frames [t-1, t+2], [t-3, t+3] and [t-7, t+2] respectively. In

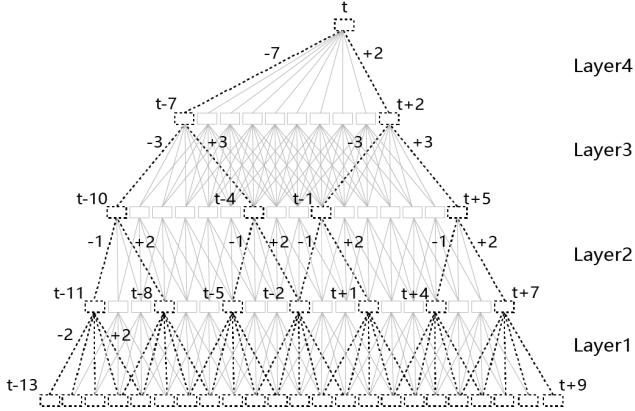


Figure 2. TDNN architecture.

total, the DNN has a left-context of 13 and a right-context of 9.

III. EXPERIMENTAL SETUP

To establish a baseline framework, we built a classical ivector and xvector system, the feature we use include: MFCC, DBN features and AFs. At the same time, we also establish a n-gram LM based back-end system for comparing the performance with TDNN. Baseline systems are summarized below. All the experiments were conducted with Kaldi toolkit [15].

A. Description of Database

The DBN features are extracted from an ASR DNN trained on two mandarin corpus. The first one is from the Chinese National Hi-Tech Project 863 for Mandarin LVCSR system development [16], and the second corpus is an open-source Mandarin speech corpus called AISHELL-1 [17]. A total of 250,000 utterances spoken by 1800 speakers (300 hours) were used for acoustic modeling.

All experiments were evaluated on AP17-OLR databases which used for second oriental language recognition challenge [18]. The database consists of 10 different languages: Kazakh in China (ka-cn), Tibetan in China (ti-cn), Uyghur in China (uy-id), Cantonese in China Mainland and Hong Kong (ct-cn), Mandarin in China (zh-cn), Indonesian in Indonesia (id-id), Japanese in Japan (ja-jp), Russian in Russia (ru-ru), Korean in Korea (ko-kr), and Vietnamese in Vietnam (vi-vn). The duration of training data for each language is about 10 hours and the speeches were recorded with mobile phones, at a sampling rate of 16 kHz and 16 bits resolution. Our systems evaluated on one of databases' condition called "test all".

B. Features Extraction

The acoustic MFCC features are 40-dim without cepstral truncation and with a frame-length of 25ms. These features are equivalent to filter bank coefficients, but are more compressible.

The phonetic DBN features are from an ASR DNN acoustic model. The DNN is a time-delay acoustic model

with p-norm nonlinearities, and the DNN model was trained on the AISHELL-1 and 863 mandarin corpus described on Section III. The DNN has 6 hidden layers, and the dimension of each DNN layer is set to be 650, except for the last hidden layer is replaced with a 100 dimensional linear bottleneck layer. Its input features are 40-dimentional MFCC. Excluding the softmax output layer, which is not needed to compute DBN features. No fMLLR or ivectors are used for speaker adaptation.

The AFs are also from an ASR DNN acoustic model, but they were extracted from the posteriors of softmax output layer instead of the bottleneck layer. The DNN is a chain time-delay acoustic model with p-norm nonlinearities, and the DNN model was trained on the AISHELL-1 and 863 mandarin corpus described on Section III. The DNN has 6 hidden layers, and the dimension of each DNN layer is set to be 625. Its input features are 40-dimentional MFCC. Furthermore, the input features are at the original frame rate of 100 per second and the output frame rate is reduced by 3-fold.

C. I-vector Baseline

The ivector system follows the procedure described in [19]. It is based on the GMM-UBM and the UBM is a 2048 component full-covariance GMM. The system uses a 400 dimensional ivector extractor and cosine for scoring. The input features to ivector system separately incorporates mentioned above three features: MFCC, DBN and AFs.

D. X-vector Baseline

The xvector system follows the procedure described in [20]. The network of xvector is a 5 layer TDNN. The input of each layer is the sliced output of the previous layer. The sliced indices in the consecutive layers were [t-2, t-1, t, t+1, t+2; t-2, t, t+2; t-3, t, t+3; t, t]. The dimension of 1-4 layer is 512, and 5th layer is 1500. The segment-level part is a 2-layer fully connected network with 512-dim per layer and the nonlinearities are rectified linear units (ReLUs). The output is a softmax layer and the size is 10 (the number of languages).

E. TDNN SLR Classifier

The detail of TDNN back-end is described in section II. The configuration of TDNN shows below. The TDNN model was composed of 6 layers and the dimension of each layer is 650. The activation function was p-norm and the spliced indices in the consecutive layers were [t-2, t-1, t, t+1, t+2; t-1, t, t+1; t-1, t, t+1; t-3, t, t+3; t-6, t-3, t]. The output is a softmax layer and the size is 10 (the number of languages).

IV. RESULTS

A. AFs based TDNN vs Baseline System

Table II shows the performance of different features in ivector, xvector and the proposed SLR system. We use All_AFs to denote 7 attributes combined together. As we can see, AFs based systems achieve lower EER than DBN based system and MFCC based system. Overall, AFs

based TDNN outperforms DBN based TDNN by about 46%, AFs based ivector outperform DBN based ivector by about 9% and AFs based xvector outperform DBN based xvector by about 20%. It is evident that AFs are beneficial to improving the performance of SLR task.

Moreover, the result of AFs based TDNN is impressive. AFs based TDNN which has a 15% relative improvement perform better than AFs based ivector. At the same time, the result of AFs based TDNN and AFs based xvector is close. It reveals that a simple TDNN back-end with AFs is effective to SLR task.

TABLE III. SYSTEM PERFORMANCE IN DIFFERENT METHODS IN TERMS OF PERCENTAGE OF EER AND MINCAVG (REPORTED WITHIN PARENTHESIS)

Feature	TDNN	Ivector+cosine	Xvector+cosine
MFCC	11.29(12.09)	6.22(6.87)	5.76(5.13)
DBN	7.17(6.88)	5.02(4.76)	4.43(4.53)
All_AFs	3.86 (3.56)	4.56(4.32)	3.52 (3.22)

B. Fusion

The fusion results shown in table III, we evaluate the fusion system using the Focal toolkit [21], and we fuse AFs based TDNN system to different approach. As we can see, the fusion systems get significant performance gain for SLR. Especially, The AFs_ivector+AFs_xvector+AFs_TDNN fusion system delivered a relative improvement of 45% in EER to the best AFs based xvector approach alone.

TABLE IV. THE PERFORMANCE OF DIFFERENT FUSION SYSTEM IN TERMS OF PERCENTAGE OF EER AND MINCAVG (REPORTED WITHIN PARENTHESIS)

Fusion	EER(minCavg)
MFCC_TDNN + AFs_TDNN	3.46(3.76)
DBN_TDNN + AFs_TDNN	2.95(3.21)
DBN_ivector + AFs_TDNN	2.56(2.32)
DBN_xvector + AFs_TDNN	2.21(2.36)
AFs_ivector + AFs_TDNN	2.27(2.53)
AFs_xvector + AFs_TDNN	2.14(2.01)
AFs_ivector + AFs_xvector + AFs_TDNN	1.92 (1.84)

TABLE V. TDNN BASED SYSTEM PERFORMANCE IN DIFFERENT AFs AND AF's COMBINATION IN TERMS OF PERCENTAGE OF EER AND MINCAVG (REPORTED WITHIN PARENTHESIS)

Feature	EER(minCavg)
MFCC	11.29(12.09)
Manner(MA)	9.51(10.21)
Place(PA)	6.53 (6.14)
Voicing(VO)	10.45(10.23)
Aspirated(AS)	11.66(11.54)
Tonguefrontend(TF)	9.82(10.28)
Tongueheight(TH)	10.15(10.53)
Rounding(RO)	10.98(10.68)
MA + VO + AS	7.12(7.45)
MA + PA + VO + AS	5.25(4.78)
All_AFs	3.86(4.10)

C. Performance of different AFs

To investigate the performance based on the different AFs, we did several experiments to evaluate the system performance for different AFs and AFs' combination. Table V shows the SLR results, the description of AFs is described in Table I of section II. The features shown in Table V were evaluated using the TDNN system (see Section II). The results shows that place of articulatory (PA) can significantly improve the performance of SLR. The PA based system show the highest performance among all single AFs system, and this indicates that different AFs have different effects on the SLR system. Furthermore, the performance will improve when combine different AFs.

V. CONCLUSION

In this paper, we have explored using AFs based TDNN modeling for SLR task. This approach took advantage of the cross-lingual characters of AFs and the capability of TDNN capturing long term dependencies between input features . The experiments were performed on AP17-OLR database revealed effectiveness of the proposed approach. Specifically, The experimental results show that our proposed approach provides a 23.10% and 12.87% relative improvement in EER to DBN features based i-vector and xvector approach. The AFs based i-vector or x-vector approach also achieved performance gain to DBN based approach. Furthermore, we evaluated the fusion system. The fusion of AFs based TDNN approach with different baseline approach got significant performance gain for SLR task. Finally, We evaluated the effectiveness of different AFs. The result shows that the place of articulation is the most effective feature compared to other AFs. These results is a strong support that the AFs based TDNN approach is beneficial to the SLR task.

ACKNOWLEDGMENT

This study was supported by Advanced Innovation Center for Language Resource and Intelligence (KYR17005), the Fundamental Research Funds for the Central Universities (16ZDJ03, 18YJ030006, 19YCX113), and the project of "Intelligent Speech technology International Exchange". Jinsong Zhang is the corresponding author.

REFERENCES

- [1] H. Li, B. Ma, and K. A. Lee, "Spoken language recognition: from fundamentals to practice," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1136–1159, 2013.
- [2] Schultz, Tanja, Waibel, and Alex, "Language-independent and language-adaptive acoustic modeling for speech recognition," *Speech Communication*, vol. 35, no. 1, pp. 31–51, 2001.
- [3] A. Waibel, P. Geutner, L. M. Tomokiyo, T. Schultz, and M. Woszczyna, "Multilinguality in speech and spoken language systems," *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1297–1313, 2000.

- [4] S. M. Siniscalchi, J. Reed, T. Svendsen, and C.-H. Lee, “Exploring universal attribute characterization of spoken languages for spoken language recognition,” in *Tenth Annual Conference of the International Speech Communication Association*, 2009.
- [5] S. M. Siniscalchi, J. Reed, T. Svendsen, and C.-H. Lee, “Exploiting context-dependency and acoustic resolution of universal speech attribute models in spoken language recognition,” in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.
- [6] S. M. Siniscalchi, J. Reed, T. Svendsen, and C.-H. Lee, “Universal attribute characterization of spoken languages for automatic spoken language recognition,” *Computer Speech & Language*, vol. 27, no. 1, pp. 209–227, 2013.
- [7] S. Stuker, T. Schultz, F. Metze, and A. Waibel, “Multilingual articulatory features,” in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03)., vol. 1.* IEEE, 2003, pp. I–I.
- [8] I. P. Association, I. P. A. Staff *et al.*, *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press, 1999.
- [9] F. Metze and A. Waibel, “A flexible stream architecture for asr using articulatory features,” in *Seventh International Conference on Spoken Language Processing*, 2002.
- [10] K. Kirchhoff, G. A. Fink, and G. Sagerer, “Combining acoustic and articulatory feature information for robust speech recognition,” *Speech Communication*, vol. 37, no. 3-4, pp. 303–319, 2002.
- [11] W. Li, S. M. Siniscalchi, N. F. Chen, and C.-H. Lee, “Improving non-native mispronunciation detection and enriching diagnostic feedback with dnn-based speech attribute modeling,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 6135–6139.
- [12] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, “Phoneme recognition using time-delay neural networks,” *Backpropagation: Theory, Architectures and Applications*, pp. 35–61, 1995.
- [13] A. Waibel, “Modular construction of time-delay neural networks for speech recognition,” *Neural computation*, vol. 1, no. 1, pp. 39–46, 1989.
- [14] V. Peddinti, D. Povey, and S. Khudanpur, “A time delay neural network architecture for efficient modeling of long temporal contexts,” in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.
- [15] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, “The kaldi speech recognition toolkit,” in *IEEE 2011 workshop on automatic speech recognition and understanding, no. CONF.* IEEE Signal Processing Society, 2011.
- [16] S. Gao, B. Xu, H. Zhang, B. Zhao, C. Li, and T. Huang, “Update progress of sinohear: advanced mandarin lvcsr system at nlpr,” in *Sixth International Conference on Spoken Language Processing*, 2000.
- [17] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, “Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline,” in *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*. IEEE, 2017, pp. 1–5.
- [18] Z. Tang, D. Wang, Y. Chen, and Q. Chen, “Ap17-olr challenge: Data, plan, and baseline,” in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2017, pp. 749–753.
- [19] A. McCree, G. Sell, and D. Garcia-Romero, “Augmented data training of joint acoustic/phonotactic dnn i-vectors fornist lre15,” *Proc. of IEEE Odyssey*, 2016.
- [20] D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, D. Povey, and S. Khudanpur, “Spoken language recognition using x-vectors,” in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 105–111.
- [21] N. Brummer, “Focal multi-class toolkit,” <https://sites.google.com/site/nikobrummer/focal>, 2014.

A Study on Syntactic Complexity and Text Readability of ASEAN English News

Yusha Zhang

Hunan University of Information Technology
Changsha, China

Abstract—English is the most widely used language in the world. With the spread and evolution of language, there are differences in the English text expression and reading difficulty in different regions. Due to the difference in the content and wording, English news in some countries is easier to understand than in others. Using an accurate and effective method to calculate the difficulty of text is not only beneficial for news writers to write easy-to-understand articles, but also for readers to choose articles that they can understand. In this paper, we study the differences in the text readability between most ASEAN countries, England and America. We compare the textual readability and syntactic complexity of English news texts among England, America and eight ASEAN countries (Indonesia, Malaysia, Philippines, Singapore, Brunei, Thailand, Vietnam, Cambodia). This paper selected the authoritative news media of each country as the research object. We used different indicators including Flesch-Kincaid Grade Level (FKG), Flesch Reading Ease Index (FRE), Gunning Fog Index (GF), Automated Readability Index (AR), Coleman-Liau Index (CL) and Linsear Write Index (LW) to measure the textual readability, and then applied L2SCA to analyze the syntactic complexity of news text. According to the analysis results, we used the hierarchical clustering method to classify the English texts of different countries into six different levels. Moreover, we elucidated the reasons for such readability differences in these countries.

Keywords- Textual Understanding Difficulty; Textual Readability; Syntactic Complexity; Hierarchical Clustering.

I. INTRODUCTION

Language is the carrier of information. English, as a dominant language worldwide, evolves by its nature and adapts to the need of its users, leading to national and regional variation of how English is written and spoken. English news texts mirror a nation's English language level and language habits, reflecting the changes occurred in English. They are easier to understand in some countries than others as a result of different content and wording in respective countries. How to measure the difficulty of English texts in different countries in a scientific and quantitative way has become a research hotspot in linguistics.

Using an accurate and effective method to calculate the difficulty of text would conduce not only to content management for news writers, but to proper materials selection for readers. The study of English text readability is of great significance for reading theory and reading instruction (such as the choice of teaching materials, the choice of reading test materials, the choice of reading psychology research materials, the choice of reading instruction methods, etc.) [1]. Scholars mainly measure the difficulty of English text from the perspectives of text readability and syntactic complexity. Readability, also known as readability or legibility, refers to the degree or nature of text that is easy to read and understand [1].

Nankai Lin[✉] and Shengyi Jiang
School of Information Science and Technology
Guangdong University of Foreign Studies
Guangzhou, China
neakail@outlook.com

Syntactic complexity refers to the range and complexity of language forms in language output [2].

This paper focuses on the quantitative study of news readability and news syntactic complexity in England, America and eight ASEAN countries (Indonesia, Malaysia, Philippines, Singapore, Brunei, Thailand, Vietnam, Cambodia) and compares their differences in the usage of English. We use different indicators to measure the textual readability. They are: Flesch-Kincaid Grade Level (FKG), Flesch Reading Ease Formula (FRE), Gunning Fog Index (GF), Automated Readability Index (AR), Coleman-Liau Index (CL) and Linsear Write Formula (LW). We also use L2SCA to analyze the syntactic complexity.

Based on the analysis results, we use the hierarchical clustering method to classify the English texts of different countries into six different levels.

The remaining part of this paper is organized as follows: Section 2 briefly reviews related studies; Section 3 demonstrates our approach; Section 4 provides information of our data, as well as the results and analysis; Section 5 concludes our work.

II. RELATED WORK

The readability assessment deals with estimating the level of difficulty in reading texts. More and more scholars have carried out text readability analysis on a wide variety of texts. Du Bay and William H introduced the research on readability and the readability formulas [3]. Commonly used text readability evaluation criteria are Flesch Reading Ease Formula (FRE), Automated Readability Index(AR), Gunning Fog Index (GF), Flesch-Kincaid Grade Level (FKG), Coleman-Liau Index (CL) and Linear Write Formula (LW). Rudolph Flesch put forward the simplification of Flesch Reading Ease Formula [4]. Automated Readability Index (ARI) was devised by Smith et al [5]. AR computed the average word length and average sentence length. Appropriate weightings of these factors result in an index reflecting the readability of the passage. Gunning Fog Index (GF) was created by Gunning in 1952 [6]. Kincaid et al recalculated Automated Readability Index, Gunning Fog Index and Flesch Reading Ease Formula for naval purpose and created the Flesch-Kincaid Grade Level [7]. Meri Coleman and Liau proposed a new calculation method Coleman-Liau Index [8]. There is no need to consider syllables since letter count is a better predictor of readability than syllable count. Linsear Write Formula is not actually presented as an index of readability. O'heyre explains that this formula is more about “write ability”, that is, it serves the writer, and not the reader. The formula aims at helping writers to use simple, one-syllable words [9]. Betul Karakus et al research the readability analysis of Turkish elementary school textbooks [10]. Scotti A. Crossley et al assessed text

readability using cognitively based indices [11]. Solnyshkina et al demonstrated the correlations between the narrativity, abstractness and word concreteness of the texts and Flesch-Kincaid Grade Level [12]. Another study was done by Brenda Lynn Hoketo to see if readability levels printed on recreational reading books were as accurate as when the Fry formula and the Flesch-Kincaid Grade Level were applied to them [13].

In addition to text readability, text complexity is an important indicator of measuring the difficulty of text. Lu described a computational system L2SCA for automatic analysis of syntactic complexity in second language writing using fourteen different measures that have been explored or proposed in studies of second language development [14]. Moreover, Xiaofei Lu has conducted more in-depth studies in syntactic complexity analysis [15][16][17][18][19][20]. Chen et al. used the academic texts of computer and library/information science as the research object, and compared the readability and complexity of academic texts from the aspects of disciplines, text structure, writer's position and j impact factors for different journals [21]. Wu used a variety of readability evaluation formulas and syntactic complexity analysis tools L2SCA to observe the syntactic complexity and text readability of Chinese journalists in different disciplines [22].

III. APPROACH

A. Textual readability analysis

We use different indicators to analyze the readability of English in ten countries from different perspectives.

FRE (Flesch Reading Ease Formula) — FRE is a simple approach to assess the grade-level of the reader. This evaluation method mainly measures the readability of the text from the average sentence length and the average syllable number of words in the text. Flesch believes that the longer the sentence is, the harder it is to read and the more complex the pronunciation of the word is, the harder it is to understand. The formula is:

$$FRE = 206.835 - (1.015 * ASL) - (84.6 * ASW)$$

ASL refers to the average sentence length which is obtained by the number of words divided by the number of sentences, *ASW* refers to the average number of syllables per word which is obtained by the number of syllables divided by the number of words. *FRE* means the score of readability. The Flesch Reading Ease formula will output a number from 0 to 100 and a higher score indicates easier reading. The score of 90-100 means "very easy", 80-89 is "easy", 70-79 is "fairly easy", 60-69 is "standard", 50-59 means "fairly difficult", 30-49 means "difficult" and 0-29 is "very confusing". The above scales are assigned to grade level 1-7.

FKG (Flesch-Kincaid Grade Level) — FKG evaluation method is an improvement of Kincaid's FRE method. This method describes the relationship between ASL and ASW on text score more reasonably. This method was originally developed for the purpose of the U.S. Navy and is most suitable for the field of education. FKG outputs

a U.S. school grade level which indicates the average student in that grade level can read the text. The formula is:

$$FKG = (0.39 * ASL) + (11.8 * ASW) - 15.59$$

ASL refers to the average sentence length which is obtained by the number of words divided by the number of sentences, *ASW* refers to the average number of syllables per word which is obtained by the number of syllables divided by the number of words. A score of 9.3 means that a ninth grader would be able to read the document. Because FKG believes that the lowest reading level is third grade level, its output range is between 3-13.

GF (Gunning Fog Index) — GF is similar to the Flesch scale in that it compares syllables and sentence lengths. FRE and FKG use the average number of bytes to express the difficulty of words in the text, while GF directly calculates the proportion of difficult words in the text, and defines "Foggy" words. "Foggy" words are words that contain 3 or more syllables. A Fog score of 5 is readable, 10 is hard, 15 is difficult, and 20 is very difficult. The formula is:

$$GF = 0.4 * (ASL + PHW)$$

ASL refers to the average sentence length which is obtained by the number of words divided by the number of sentences and *PHW* is the percentage of hard words in the context. The ideal score for readability with the Fog index is 7 or 8. Anything above 12 is too hard for most people to read. We divide the score into 14 levels. 14th level represents the hardest.

AR (Automated Readability Index) — The Automated Readability Index (AR) is a readability test designed to assess the understandability of a text. It measures the readability of a text from the ratio of the number of characters to the number of words and the ratio of the number of words to the number of sentences. AR outputs a number which approximates the required grade level. For example, if the AR outputs the number 3, it means students in 3rd grade (ages 8-9 yrs. old) should be able to comprehend the text. The formula is:

$$AR = 4.71 * \frac{\text{characters}}{\text{words}} + 0.5 * \frac{\text{words}}{\text{sentences}} - 21.43$$

Here *characters* is the number of letters and numbers, and *words* is the number of words in the text while the *sentences* means the number of sentences.

CL (Coleman-Liau Index) — CL relies on characters and sentence length instead of syllables per word. It measures the readability of a text in terms of the average number of characters and sentences per hundred words. This formula will output a grade. For example, 10.6 means your text is appropriate for a 10-11th grade high school student. The formula is:

$$CLI = 0.0588 * L - 0.296 * S - 15.8$$

L is the average number of letters per 100 words. *S* is the average number of sentences per 100 words. CL is similar to the Automated Readability Index, but unlike most of the other grade-level predictors that rely on syllables per word.

LW (Linsear Write Formula) — Linsear Write Formula is a readability formula for English text, originally developed for the United States Air Force to help them

calculate the readability of their technical manuals. Linsear Write Formula is specifically designed to calculate the grade level in the US of a text sample based on sentence length and the number of three-plus syllable words. The LW score computing method is show in Table 1.

IV. DATA AND RESULT

A. Data

We selected the news of the authoritative English newspapers of various countries as experimental data. We regarded the official media of official institutions, governments or enterprises and the large media with great influence as the authoritative English newspapers, such as the New York Times, which is the most influential newspaper in the United States, and the Antara News, the official newspaper of Indonesia. The experimental data in this paper is shown in Table 4.

B. Textual readability analysis result

We used six evaluation methods to analyze the text readability of ten countries. The results are shown in Table 5-10. We also present the cumulative results of each evaluation indicator in the form of line charts, as shown in Figures 1-6.

The evaluation results of FRE, FKG, GF and LW are similar. The English news texts are less readable in Cambodia, Indonesia and Singapore than the other seven countries. We regarded the English of America and England as the standard English and we could see that the English news texts of Malay and Thailand are more easy-to-understand than the standard English. Among the AR evaluation results, countries with lowest English readability are Cambodia, Indonesia and Singapore, followed by Vietnam, Philippines and Myanmar, and the other four countries have higher English readability. The CL evaluation indicators show that English texts in Vietnam, Cambodia, Indonesia, and Singapore are more difficult, while British English and American English texts are more readable.

C. Syntactic complexity analysis result

According to the results of the L2SCA syntactic analysis tool, most of the indices in the news texts of Singapore and Cambodia are higher than others. The syntactic complexity of the news texts of these two countries is high, and some indices of Indonesia (C/T, CT/T, CN/T) and Vietnam (MLC, CP/T, CP/C, CN/C) have higher values. Indonesian news texts have higher results when calculating syntactic complexity in units of T while the Vietnamese news text has higher results when calculating the syntactic complexity in units of C. The syntactic complexity of English texts in other countries is low.

TABLE I. LW SCORE COMPUTING FLOW

- (1) Calculate the easy words (defined as two syllables or less) and place a number "1" over each word, even including a, an, the, and other simple words.
- (2) Calculate the hard words (defined as three syllables or more) and place a number "3" over each word as pronounced by the dictionary.
- (3) Multiply the number of easy words times "1".
- (4) Multiply the number of hard words times "3".
- (5) Add the two previous numbers together.
- (6) Divide that total by the number of sentences.
- (7) If the answer of (6) is >20, divide by "2".
- (8) If the answer of (6) is <20 or equal to 20, subtract "2" and then divide by "2".

TABLE II. THE NINE SYNTACTIC COMPLEXITY BASIC INDICATORS

Index	Meaning
W	Length of text
S	Number of sentence
VP	Number of verb phrases
C	Number of clause
CN	Number of Complex nominals
T	T-unit
DC	Dependent clauses
CT	Complex T-unit
CP	Number of complex nominals

TABLE III. THE FOURTEEN SYNTACTIC COMPLEXITY EVALUATION INDICATORS

Index	Meaning
MLS	Mean length of sentence
MLT	Mean length of T-unit
MLC	Mean length of clause
C/S	Sentence complexity ratio
CT/T	Complex T-unit ratio
C/T	T-unit complexity ratio
DC/C	Dependent clause ratio
DC/T	Dependent clauses per T-unit
T/S	Sentence coordination ratio
CP/C	Coordinate phrases per clause
CP/T	Coordinate phrases per T-unit
VP/T	Verb phrases per T-unit
CN/T	Complex nominals per T-unit
CN/C	Complex nominals per clause

TABLE IV. THE DATA THAT WE USED IN THE EXPERIMENTS

Country	Abbreviation	Website	Number of news
Indonesia	ID	http://www.antaranews.com/	36001
Malay	MY	http://www.nst.com.my/	76764
Philippines	PH	http://www.malaya.com.ph/	60858
Singapore	SG	http://www.todayonline.com/	56323
Brunei	BN	http://www.brudirect.com/	38313
Thailand	TH	http://www.bangkokpost.com/	104914
Vietnam	VN	http://vov.vn/	59456
Cambodia	KH	https://www.cambodiadaily.com/	38161
English	UK	http://www.theguardian.com/uk	261591
America	USA	http://www.nytimes.com	300094

TABLE V. FRE CALCULATION RESULT (%)

Country	FRE level						
	1	2	3	4	5	6	7
SG	13.78	20.06	17.42	13.56	9.21	10.80	15.17
MY	91.29	7.93	0.40	0.06	0.01	0.02	0.29
PH	82.05	15.39	1.78	0.26	0.07	0.07	0.38
BN	75.01	11.12	0.78	0.12	0.03	0.01	12.95
TH	94.46	4.86	0.30	0.08	0.01	0.01	0.28
VN	80.61	16.39	1.21	0.16	0.04	0.02	1.56
KH	3.30	10.83	17.10	16.95	14.06	17.22	17.22
USA	89.74	9.06	0.84	0.18	0.06	0.04	0.08
ID	8.53	11.49	13.29	12.56	10.47	14.39	29.28
UK	88.07	10.71	0.90	0.15	0.07	0.03	0.07

TABLE VI. FKG CALCULATION RESULT (%)

Country	FKG level										
	3	4	5	6	7	8	9	10	11	12	13
SG	0.73	0.85	2.16	3.61	2.97	5.32	3.96	6.08	6.20	3.82	64.31
MY	13.13	13.01	24.82	23.29	10.42	8.80	2.89	1.97	0.75	0.20	0.70
PH	9.95	9.39	19.82	21.80	12.04	12.46	4.95	4.45	2.20	0.76	2.18
BN	9.94	9.55	18.76	19.18	10.20	10.10	3.73	3.13	1.29	0.44	13.68
TH	18.58	15.88	26.77	20.64	8.02	5.94	1.74	1.19	0.45	0.21	0.58
VN	9.02	9.76	19.25	21.82	11.96	12.19	5.44	4.88	2.25	0.75	2.69
KH	0.10	0.10	0.34	0.76	0.75	1.92	1.61	3.20	3.95	2.84	84.43
USA	20.03	12.77	22.14	19.43	9.24	8.39	3.04	2.46	1.10	0.40	1.01
ID	0.48	0.52	1.46	2.21	1.86	2.94	2.24	3.44	3.69	2.42	78.74
UK	17.51	12.58	21.79	19.58	9.68	9.49	3.63	2.97	1.30	0.45	1.02

TABLE VII. AR CALCULATION RESULT (%)

Country	AR level													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
SG	0.01	0.02	0.03	0.04	0.06	0.16	0.39	0.71	0.98	1.44	1.97	2.31	2.72	89.16
MY	0.11	0.09	0.15	0.34	1.15	3.08	6.60	10.61	13.09	13.36	12.02	9.21	7.39	22.81
PH	0.11	0.11	0.20	0.42	1.09	2.30	4.60	7.33	9.62	10.98	10.91	9.55	8.26	34.54
BN	0.04	0.04	0.16	0.29	0.96	2.58	4.86	7.58	8.78	9.28	8.32	7.16	6.55	43.39
TH	0.06	0.20	0.44	0.71	1.71	4.08	7.45	10.76	11.45	10.57	8.72	7.59	8.01	28.26
VN	0.04	0.04	0.06	0.21	0.65	1.93	3.57	5.61	6.37	6.37	6.33	6.15	7.12	55.56
KH	0.01	0.01	0.10	0.10	0.10	0.42	0.60	0.10	0.18	0.29	0.57	0.74	1.18	96.78
USA	1.06	0.73	0.69	1.14	2.64	5.22	8.49	11.73	13.68	13.37	11.61	9.02	6.44	14.18
ID	0.04	0.01	0.01	0.02	0.03	0.10	0.18	0.37	0.54	0.77	1.04	1.24	1.40	94.27
UK	1.07	0.63	0.78	1.30	2.15	4.27	7.72	11.08	12.79	12.79	11.38	9.34	7.44	17.26

TABLE VIII. GF CALCULATION RESULT (%)

Country	GF level									
	1	2	3	4	5	6	7	8	9	10
SG	0.03	0.11	0.32	1.12	1.30	3.13	2.84	4.93	20.07	66.13
MY	0.47	2.56	5.65	17.49	16.41	24.84	12.59	11.46	7.97	0.57
PH	0.45	2.21	4.07	12.64	12.68	21.95	13.51	14.77	15.50	2.23
BN	0.50	2.48	4.28	13.79	13.31	22.28	13.24	14.47	14.22	1.44
TH	1.53	4.01	7.45	21.62	18.05	23.79	10.14	8.11	4.87	0.43
VN	0.26	1.78	3.95	12.92	12.72	21.98	13.54	14.63	16.72	1.49
KH	0.01	0.02	0.05	0.13	0.19	0.59	0.61	1.70	10.83	85.87
USA	2.18	5.41	7.42	17.81	14.80	20.98	10.83	10.35	9.06	1.17
ID	0.09	0.10	0.17	0.66	0.91	1.98	1.74	2.97	11.68	79.70
UK	1.76	4.24	6.61	17.47	14.47	20.96	11.16	11.16	10.71	1.21

TABLE IX. LW CALCULATION RESULT (%)

Country	LW level											
	1	2	3	4	5	6	7	8	9	10	11	12
SG	0	0.01	0.04	3.12	41.39	41.94	11.47	1.87	0.17	0.01	0.01	0
MY	0.01	0.02	0.57	43.45	55.67	0.21	0.05	0.02	0.01	0.01	0	0
PH	0	0.01	0.51	31.03	67.18	1.26	0.01	0	0	0	0	0
BN	0	0	0.67	31.53	65.31	2.48	0.01	0	0	0	0	0
TH	0	0.02	1.75	50.08	47.94	0.21	0.01	0	0	0	0	0
VN	0	0.01	0.37	28.34	70.33	0.94	0.02	0.01	0	0	0	0
KH	0	0	0.01	0.64	25.41	56.87	15.73	1.29	0.04	0	0	0
USA	0.01	0.09	2.83	49.64	46.98	0.40	0.06	0.01	0.01	0	0	0
ID	0	0.02	0.06	1.48	23.11	46.26	23.23	5.34	0.49	0.02	0.03	0
UK	0.02	0.09	2.48	47.74	49.23	0.42	0.04	0.01	0.01	0	0.01	0.01

TABLE X. CL CALCULATION RESULT (%)

Country	CL level												
	1	2	3	4	5	6	7	8	9	10	11	12	13
SG	0	0.01	0.01	0.01	0.01	0.03	0.12	0.32	1.07	2.72	5.50	9.12	81.12
MY	0	0	0.01	0.01	0.02	0.09	0.36	1.00	2.30	4.10	6.68	8.15	77.30
PH	0	0	0	0.01	0.03	0.18	0.51	1.15	2.08	3.62	6.40	9.02	77.00
BN	0	0	0	0	0.01	0.05	0.26	0.84	1.69	3.15	5.71	7.94	80.36
TH	0	0	0	0.01	0.01	0.05	0.27	0.85	1.95	3.72	6.25	9.01	77.88
VN	0	0	0	0	0.01	0.03	0.11	0.28	0.78	1.42	2.48	3.74	91.15
KH	0	0	0	0.01	0.01	0.03	0.18	0.67	2.19	4.59	7.85	10.81	73.66
USA	0	0.01	0.02	0.08	0.29	0.83	2.11	4.90	8.73	13.03	15.79	15.05	39.17
ID	0	0.01	0.01	0.01	0.01	0.01	0.01	0.04	0.09	0.45	1.19	2.95	95.26
UK	0	0.01	0.04	0.11	0.34	0.89	2.10	4.14	7.06	10.72	13.61	13.96	47.04

TABLE XI. SYNTACTIC COMPLEXITY INDEX FOR EACH COUNTRY

Index	Country									
	UK	USA	ID	MY	TH	PH	VN	BN	SG	KH
W	822.22	680.47	332.14	341.35	487.96	422.08	265.98	302.15	567.22	455.34
S	39.81	31.42	13.95	15.25	21.93	17.84	10.57	13.42	15.40	13.81
VP	98.32	84.23	35.54	39.60	56.81	45.96	24.85	34.45	65.27	57.94
C	76.20	63.67	27.10	29.50	42.10	34.15	17.05	24.71	47.79	44.49
T	43.37	36.28	12.87	15.77	23.61	18.83	11.08	14.17	18.55	16.89
DC	27.91	24.25	10.61	10.99	15.47	12.70	4.91	9.31	23.96	20.56
CT	19.82	17.44	6.61	7.69	11.06	8.93	3.93	5.99	11.60	9.77
CP	18.73	15.37	7.28	7.47	11.31	10.52	9.66	6.46	11.95	8.87
CN	98.42	82.51	42.04	40.95	59.59	52.57	34.37	34.48	75.37	58.18
MLS	22.06	23.08	24.15	23.34	23.78	25.22	26.46	24.05	46.58	35.98
MLT	20.22	20.18	27.53	23.78	22.49	24.22	25.59	24.10	39.97	29.54
MLC	11.65	11.50	13.01	12.54	12.72	13.71	17.48	15.54	13.80	10.67
C/S	1.94	2.05	1.93	1.93	1.94	1.91	1.59	1.71	3.76	3.40
VP/T	2.29	2.38	2.86	2.63	2.47	2.48	2.26	2.39	4.40	3.66
C/T	1.76	1.78	2.18	1.94	1.82	1.82	1.52	1.66	3.21	2.79
DC/C	0.36	0.37	0.38	0.36	0.35	0.35	0.26	0.34	0.54	0.46
DC/T	0.66	0.69	0.88	0.73	0.68	0.68	0.43	0.61	1.70	1.33
T/S	1.09	1.15	0.90	1.00	1.07	1.04	1.04	1.01	1.32	1.24
CT/T	0.46	0.49	0.52	0.50	0.47	0.48	0.35	0.37	0.72	0.59
CP/T	0.47	0.46	0.60	0.51	0.52	0.63	0.98	0.51	0.84	0.58
CP/C	0.28	0.27	0.29	0.28	0.31	0.37	0.68	0.32	0.30	0.21
CN/T	2.48	2.51	3.50	2.87	2.79	3.04	3.31	2.65	5.39	3.84
CN/C	1.42	1.42	1.64	1.51	1.57	1.71	2.24	1.65	1.83	1.38

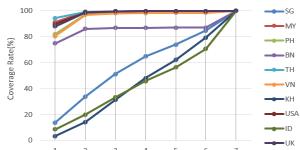


Figure 1. FRE cumulative coverage rate.

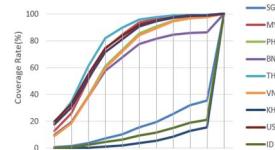


Figure 2. FKG cumulative coverage rate.

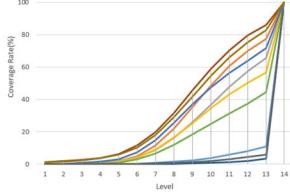


Figure 3. AR cumulative coverage rate.

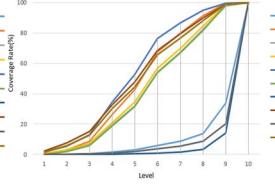


Figure 4. GF cumulative coverage rate.

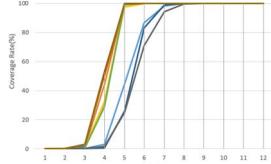


Figure 5. LW cumulative coverage rate.

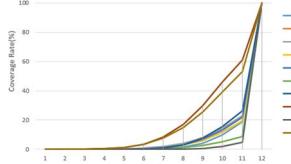


Figure 6. CL cumulative coverage rate.

D. Overall result analysis

There are different indicators in text readability and syntactic complexity. In order to comprehensively consider each indicator, we used all the indicators to cluster the English texts of each country. We utilized the average of each indicator of text readability and the average of the various indicators of syntactic complexity, and applied the hierarchical clustering method to cluster the English texts of each country. We implemented this with the `scipy`¹ tool. The parameters of the model we used are shown in Table 12. The result is shown in Figure 7.

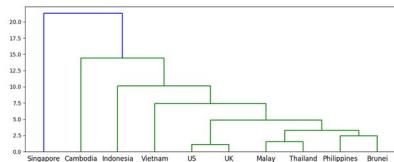


Figure 7. Hierarchical clustering result.

TABLE XII. THE PARAMETERS OF THE METHOD

Parameter	Value
metric	euclidean
optimal ordering	false
method	weighted

¹ <https://github.com/scipy/scipy>

TABLE XIII. THE RESULT OF THE CLASSIFICATION

Level	Country
1	Malay Thailand Philippines Brunei
2	America England
3	Vietnam
4	Indonesia
5	Cambodia
6	Singapore

Based on the results of hierarchical clustering, we divided the difficulty of English texts in ten countries into six levels. Level 1 represents the easiest level and level 6 is the most difficult level. The result of the classification is shown as table 13. We can see that, as standard English, the English texts' difficulty in America and England is at the second level. The results show that Vietnam, Indonesia, Cambodia and Singapore's news are more difficult than the news of America and England.

We conducted an in-depth study of the grading results. English is highly popular in Singapore, but its distinctive "Singapore English" is more difficult for non-Singapore people to read. Cambodia pays attention to the promotion of Khmer in the country. The use of English in Cambodia is not as high as Khmer, so it also has a certain reading difficulty. Due to its high openness and the absorption of multiple languages, Indonesia has reduced the standardization of language use, which makes reading in English more difficult.

V. CONCLUSION

This paper focuses on the quantitative study of the difficulty of English texts in the United Kingdom, the United States and ASEAN countries, and researches the differences in the English news text readability and syntactic complexity between these countries. We used different indicators that including FKG, FRE, GF, AR, CL and LW to measure the textual readability, and then used L2SCA to analyze the syntactic complexity of news text. According to the analysis results, we used the hierarchical clustering method to classify the English texts of different countries into six different levels according to the difficulty of the text. Moreover, we analyzed the reasons for the differences in the difficulty of English texts in these countries. In the future, we will further analyze the text readability from other dimensions.

ACKNOWLEDGEMENT

This work was supported by the 13th Five-Year Plan Project of Educational Science for Hunan Province (No. XJK18CGD044). The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

- [1] S. Shan, "Overview of legibility research," Journal of PLA University of Foreign Languages, vol. 23, pp. 1-5, 2004.

- [2] L. Ortega, "Syntactic Complexity Measures and their Relationship to L2 Proficiency: A Research Synthesis of College-level L2 Writing," *Appl. Linguist.*, 2003.
- [3] W. Dubay, "The Principles of Readability," *CA*, vol. 92627949, pp. 631–3309, 2004.
- [4] R. Flesch, "New Facts about Readability," *Coll. English*, vol. 10, pp. 225-226, 1949.
- [5] E. A. Smith and R. J. Senter, "Automated readability index," *AMRL-TR. Aerosp. Med. Res. Lab.*, 1967.
- [6] R. Gunning, *The Technique of Clear Writing*, McGraw-Hil. New York, 1968.
- [7] J. P. Kincaid, J. Fishburne, R. Robert P., C. Richard L., and Brad S., "Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel," 1975.
- [8] M. Coleman and T. L. Liau, "A computer readability formula designed for machine scoring," *J. Appl. Psychol.*, 1975.
- [9] J. O'Hare, "Gobbledygook Has Gotta Go," D.C.: U.S. Dept. of the Interior, Bureau of Land Management, Washington, 1966.
- [10] B. Karakus, G. Aydin, and I. R. Hallac, "Distributed Readability Analysis of Turkish Elementary School Textbooks," *Int. Conf. Inf. Technol. Comput. Sci.*, 2015.
- [11] S. A. Crossley, J. Greenfield, and D. S. McNamara, "Assessing text readability using cognitively based indices," *TESOL Q.*, 2008.
- [12] M. I. Solnyshkina, R. R. Zamaletdinov, L. A. Gorodetskaya, and A. I. Gabitov, "Evaluating Text Complexity and Flesch-Kincaid Grade Level," www.jsser.org *J. Soc. Stud. Educ. Res. Sos. Bilgiler Eğitimi Araştırmaları Derg.*, 2017.
- [13] B. Lynn Hoke, "Comparison of Recreational Reading Books Levels Using the Fry Readability Graph and the Flesch-Kincaid Grade Level," 1999.
- [14] X. Lu, "Automatic analysis of syntactic complexity in second language writing," *Int. J. Corpus Linguist.*, 2010.
- [15] X. Lu, "A corpus-based evaluation of syntactic complexity measures as indices of college-level ESL writers' language development," *TESOL Q.*, 2011.
- [16] H. Ai and X. Lu, "A corpus-based comparison of syntactic complexity in NNS and NS university students' writing," 2014.
- [17] W. Yang, X. Lu, and S. C. Weigle, "Different topics, different discourse: Relationships among writing topic, measures of syntactic complexity, and judgments of writing quality," *J. Second Lang. Writ.*, 2015.
- [18] X. Lu and H. Ai, "Syntactic complexity in college-level English writing: Differences among writers with diverse L1 backgrounds," *J. Second Lang. Writ.*, 2015.
- [19] X. Lu, Q. Xu, "L2 Syntactic Complexity Analyzer and its applications in L2 writing research," *Foreign Language Teaching and Research*, vol. 48, pp. 409-420, 2016.
- [20] X. Lu, "Automated measurement of syntactic complexity in corpus-based L2 writing research and implications for writing assessment," *Lang. Test.*, vol. 34, no. 4, pp. 493–511, 2017.
- [21] L. Chen, X. Li, C. Zhao, "Analysis of the Readability and Complexity of Academic Texts," *Digital Library Forum*, vol. 168, pp. 64-68, 2018.
- [22] X. Wu, "A Study on Syntactic Complexity and Text Readability of International Journal Articles by Chinese Scholars," *Journal of PLA University of Foreign Languages*, vol 40, pp.11-19, 2017.

Tibetan word segmentation method based on CNN-BiLSTM-CRF model

Lili Wang², Hongwu Yang^{1;2;3}, Xiaotian Xing², Yajing Yan²

¹College of Educational Technology, Northwest Normal University, Lanzhou 730070, China

²College of Physics and Electronic Engineering, Northwest Normal University, Lanzhou 730070, China

³National and Provincial Joint Engineering Laboratory of Learning Analysis Technology in Online Education, Lanzhou 730070, China

Email:yanghw@nwnu.edu.cn

Abstract—We propose a Tibetan word segmentation method based on CNN-BiLSTM-CRF model that merely uses the characters of sentence as the input so that the method does not need large-scale corpus resources and manual features for training. Firstly, we use convolution neural network to train character vectors. Then the character vectors are searched through the character lookup table to form a matrix C by stacking searched results. Then the convolution operation between the matrix C and multiple filter matrices is carried out to obtain the character-level features of each Tibetan word by maximizing the pooling. We input the character vector into the BiLSTM-CRF model, which is suitable for Tibetan word segmentation through the highway network, for getting a Tibetan word segmentation model that is optimized by using the character vector and CRF model. For Tibetan language with rich morphology, fewer parameters and faster training time make this model better than BiLSTM-CRF model in the performance of character level. The experimental results show that character input is sufficient for language modeling. The robustness of Tibetan word segmentation is improved by the model that can achieve 95.17% of the F value.

Keywords-Convolutional Neural Network; recurrent neural network; Conditional random field; Tibetan word segmentation

I. INTRODUCTION

Word segmentation is not only the most basic but also the most important part in natural language processing (NLP). Word segmentation provides important feature information of advanced NLP tasks involving topic recognition, topic tracking, information retrieval, machine translation and public opinion analysis. In the past, the rule-based method, the statistic-based method and the combination of rule-based method with statistic-based method are commonly used for word segmentation. In recent years, researchers have proposed many deep neural network-based methods in Chinese word segmentation and have achieved good results [1]. However, there are few studies on minority languages, such as Tibetan dialects that are usually not easy to get training corpus. Tibetan is a national language with a long history in China. It is widely used in Tibet, Qinghai, Gansu, Sichuan, Yunnan province as well as the part of Nepal, Bhutan, Pakistan and India. Not only the population of Tibetans is very large, but also the geographical location is widely distributed. Tibetan inherits and records the rich and colorful Tibetan culture. Tibetan belongs to the Tibetan-Burmese branch of the Sino-Tibetan Language Family. Tibetan is also a kind of Pinyin character. When sorting the letters, there are strict rules that should be written from left to right and from top to bottom. The basic syllable is the core position of each

character, which is used to determine the central consonant position of the character. Tibetan characters are segregated by syllable point, but there is no segregation mark between words. An example of Tibetan word structure is shown in Figure 1. The word, expressing independent meaning, is the smallest language component. Only at the word level can we improve the certainty of Tibetan processing. In this way, a good foundation for Tibetan intelligent analysis is needed. At present, studies on Tibetan word segmentation are usually based on rule-based method [2], statistic-based method [3], or the combination of these two methods [4]. However, most of them rely on artificial features, which is time-consuming and laborious. Therefore, a hybrid deep learning model based on CNN-BiLSTM-CRF is proposed for Tibetan word segmentation. Firstly, we use the convolutional neural network (CNN) to capture the character-level feature vectors of Tibetan words. Then we input character-level feature vectors into the highway network to train the context-dependent information as the input of bi-directional long short-term memory (BiLSTM) network. In this way, we can acquire the implicit semantic features between sentences and words. Finally, the optimal probability distribution is obtained by conditional random field (CRF) layer.



Figure 1. An example of the structure of Tibetan word.

II. ARCHITECTURE OF CNN-BiLSTM-CRF MODEL

Proposed Tibetan word segmentation based on CNN-BiLSTM-CRF model is mainly composed of BiLSTM module, CNN module and CRF module. The first layer is the input layer. In this module, Tibetan words are used as the current input. The hidden state of the previous moment is utilized to predict the next Tibetan words. In the process of character embedding search, the found character vectors are stacked together to form a matrix C. Then a convolution operation is conducted between the matrix C and multiple filter matrices. In this layer, we use 12 filters, three filters with 2 widths, four filters with 3 widths and five filters with 4 widths. The character level features of each word are obtained by maximizing the pooling for inputting into the highway network. The output of the highway network is the input of BiLSTM that is the

second layer of the neural network module. Finally, the third layer of CRF module is used to decode the output of the second layer into an optimal probability distribution sequence. The architecture of the neural network in this paper is given in Figure 2.

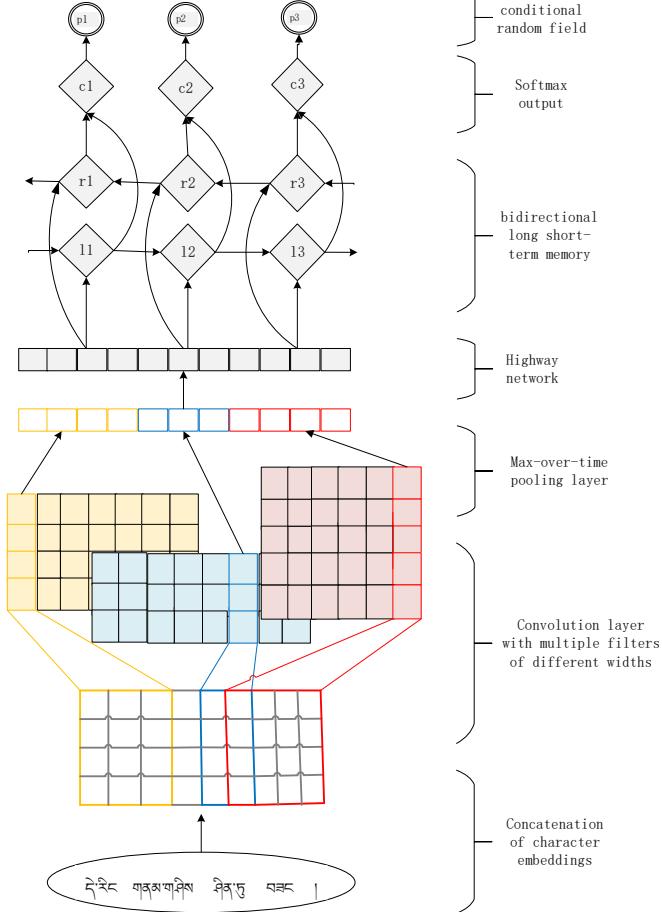


Figure 2. The flow chart of Tibetan word segmentation

A. Character vector feature

There are typical stick-in forms in Tibetan. The stick-in forms are commonly named as stick-in case, contraction word and contraction case. The stick-in forms mainly consist of various case markers, auxiliary words, conjunctions and end words. The syllables without postscript or with a postscript of ང are added to the stick-in forms, which forms a new character. This form seems like a new character. However, it is actually a condensation form of two characters, which is caused by spelling. In the meantime, the complex stick-in form is the most significant reason that affects the accuracy of word segmentation.

Through some changes such as rewriting, combination and auxiliary symbols, there are 41 consonant letters and 15 vowel letters in the contemporary Tibetan language. Besides, some numeric characters, punctuation symbols, textual modification symbols and other symbols have also been included in the Tibetan text. According to the combination of prefix subscribed letter, top-addition, bottom-addition, suffix and superscription letter, the letters forms different Tibetan characters. Tibetan characters appear in different positions such as the beginning, middle

and end of a word, which form distinctive Tibetan words. According to the Chinese-Tibetan Contrast Dictionary, there are approximately more than 80,000 words. In order to better reflect the local information, the CNN model is used to obtain the character feature vector.

B. Convolution Neural Network

In convolution neural network [7], the local feature information of text data can be extracted by the convolution layer. Besides, the most representative part of the local feature information can be extracted as vectors by using convolution layer and maximum pooling layer. Existing studies have shown that CNN can extract morphological information (such as prefix or suffix) from the characters of words for encoding to form the character feature vector. [8] extracted character-level features by CNN that has achieved good results in the field of named entity recognition. Therefore, this paper uses CNN to extract the characteristic features of Tibetan words with rich morphology.

By using CNN network and highway network, we can effectively reduce model parameters and training time. At the same time, the performance of the Tibetan word segmentation can be improved effectively by the proposed method. The structure of the CNN model is shown at the bottom of Figure 2. It consists of a character vector table, convolution layers and pooling layers.

The character vector table contain totaling 64 characters that contains 41 consonant letters, 15 vowel letters, 8 punctuation marks and an uncertain character which is not in the character set. The corresponding character vector table is generated by 64 characters respectively. The function of the character vector table is to convert each character of words into a corresponding character vector to form the corresponding character vector matrix of Tibetan words. Because of the various length of Tibetan words, the size of the generated character vector is different. This paper takes length of the longest Tibetan word as the length of character vector and uses placeholder to equal the length of each character vector. Then all character vectors are superimposed to form matrix C. Convolution operation is conducted between matrix C and multiple filter matrices. In this layer, we use 12 filters, three filters with 2 widths, four filters with 3 widths and five filters with 4 widths. The character-level features of each word are obtained by maximizing the pooling and then are inputted into the highway network. In the training process of convolution neural network, character vector table updates character vector matrix automatically through back propagation mechanism.

C. Highway Network Layer

The propagation equation of traditional neural networks (ignoring bias and layer index) is as follows:

$$y = H(x, W_H) \quad (1)$$

Where H is a no-linear function, W is a weight, x is input, and y is the output. The propagation equation of highway networks [9] is as follows:

$$y = H(x, W_H) \cdot T(x, W_T) + x \cdot C(x, W_C) \quad (2)$$

Where T is transform gate, C is carry gate. When $C = 1 - T$, equation (2) is that:

$$y = H(x, W_H) \cdot T(x, W_T) + x \cdot (1 - T)(x, W_C) \quad (3)$$

Among them, the dimension of $X, Y, H(x, W_H), T(x, W_T)$ must be the same. If the dimension is not enough, it needs to fill with zero, so we can get:

$$y = \begin{cases} x, & \text{if } T(x, W_T) = 0 \\ H(x, W_H), & \text{if } T(x, W_T) \neq 0 \end{cases} \quad (4)$$

That is to say, when the gate is 1, all the original x is output without activation. When the mean value of the gate is 0.5, half of all the original information input into the next layer is activated, which retains a lot of information. Moreover, in the process of back-propagation, more information gradient can directly return to the input layer without a non-linear transformation.

D. BiLSTM model

Long short-term memory [10] is a time recursive neural network, which can model long-distance dependent information effectively. The information is input from the input gate and then flows into the circulating cell unit which controls whether the information flows to the input gate or to the forgetting gate. The calculation equations of each unit at each time are presented in the equations as follows:

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i) \quad (5)$$

$$f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f) \quad (6)$$

$$o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o) \quad (7)$$

$$c_t = f_t \times c_{t-1} + i_t \times (\tanh(w_c \cdot [h_{t-1}, x_t] + b_c)) \quad (8)$$

$$h_t = o_t \times \tanh(c_t) \quad (9)$$

Where, i_t, f_t, o_t, c_t respectively represent the outputs of input gate, forgetting gate, output gate and cell state at t time, h_t and x_t stand for hidden layer vectors and input vectors at t time respectively. σ represent the sigmoid activation function and it can output values between 0 and 1 to describe how much each part can pass. 0 stands for "no quantity is allowed to pass" and 1 stands for "any quantity is allowed to pass". w and b stand for the weight matrix and bias vector respectively.

In spite of the good performance of LSTM network in Tibetan word segmentation, the model advances from left to right, which makes the weight of the front words smaller than that of the latter words. However, the weight of each word in the sentence should be the same for Tibetan word segmentation. Therefore, in order to obtain more accurate context information of Tibetan words, we use BiLSTM model. This model combines forward LSTM and backward LSTM models. In addition to the usage of previous input features and sentence-level markup information, it can also use future input features. The h_t can be expressed by equation (10).

$$h_t = \vec{i}_t \cup \vec{o}_t \quad (10)$$

E. CRF model

CRF layer, which is the decoding layer, mainly corrects the results of Tibetan word segmentation predicted by softmax. Although the context information is obtained, the output results are independent of each other. Softmax classifier merely selects a label output with the maximum probability, but it will produce sentence with incorrect

grammatical structure. Hence, CRF layer is used to correct these errors.

The input of CRF layer is the output of softmax. CRF layer is a matrix P of $n*m$, where n is the number of Tibetan words and m is the type of label. p_{ij} is defined as a transition score matrix from the i tag to the j tag. For a predicted tag sequence $y = y_1, y_2 \dots y_n$, probability distribution is shown in equation (11).

$$\text{score}(x, y) = \sum_{i=1}^n p_{i, y_i} + \sum_{i=1}^{n+1} A_{y_{i-1}, y_i} \quad (11)$$

The whole sequence probability consists of two parts. One is the P matrix output from BiLSTM layer, the other part is the transfer matrix A of CRF layer. y_0 and y_n in equation (11) are respectively the end and beginning markers of predictive sentences.

The likelihood function of maximizing markers for a training sample x, y^x in the training of CRF layer is shown in equation (12).

$$\log(p(y^x | x)) = \text{score}(x, y^x) - \log(\sum_{y'} e^{\text{score}(x, y')}) \quad (12)$$

Where y' represents the real markup value. In the prediction process, Viterbi algorithm of dynamic programming is used to solve the optimal path. As shown in equation (13).

$$y^* = \arg \max_{y'} \text{score}(x, y') \quad (13)$$

III. TRAINING OF THE NEURAL NETWORK

A. Word vector

In the experiment, 64 Tibetan characters and punctuation symbols were randomly distributed to initialize the character vector query table. The dimension of the character vector was 10, and its value range was [-0.5 0.5].

B. Optimization

At present, popular optimization algorithms in neural networks are SGD, Momentum, Adagrad, Adadelta, RMSprop, Adam, Adamax, etc. [13-14]. In this paper, SGD algorithm is utilized to optimize the model. The experimental results show that the performance of the model can be improved by SGD optimization algorithm. We set the learning rate as η_0 , set the initial value as 0.001 and set the momentum as 0.9. The learning rate η_0 is updated automatically by equation $\eta_t = \eta_0 / 1 + \rho_t$ in each training cycle. Among them, we set the delay rate as $\rho_t = 0.5$, and t as the number of training cycles that have been completed already.

C. Dropout parameter

Dropout [15] is a very useful technique in regularization methods. Generally speaking, it will delete some neurons randomly and train different neural network structures according to different batches. In the experiment, the value of Dropout and its position in the model are very important, which directly affects the performance of the model. In most neural networks, Dropout value is 0.5, which can effectively prevent over-fitting. However, in this experiment, due to the limited training data, the model

is cross-validated with different Dropout values. The experimental results show that the best recognition effect is achieved when the Dropout value is 0.3. In this paper, the parameters of the neural network are set as showed in Table 1.

TABLE I. PARAMETER SETTING OF CNN-BiLSTM-CRF MODEL

parameter	value	parameter	value
Character dimension	vector	10	Minimum frequency of Tibetan words
Character Window	Feature	5	Number of convolution kernels
learning rate		0.01	Depth of LSTM
Dropout		0.3	Layer Number of LSTM

IV. EXPERIMENTS AND RESULTS ANALYSIS

A. Experimental Data Set

Due to the lack of publicly annotated data set for Tibetan word segmentation, a Tibetan word segmentation data set is established to verify the effectiveness of CNN-BiLSTM-CRF model in Tibetan word segmentation. The corpus of this data set is manually extracted from the Tibetan Language Network of China, Qinghai Tibetan Language Network, Kangba Satellite TV and other websites. It comes to a grand total of 3000 Tibetan articles which mainly include various news, celebrity anecdotes, novels and other topics. Afterwards, the text is filtered and proofread manually. We divide the text into sentences and then cut them into words by maximum matching method. Finally, the corpus containing 125 386 Tibetan sentences are obtained. It includes a total of 1196 907 Tibetan words. In the experiment, 80% of the sentences is randomly selected as the training set, 10% is used as the verification set, and the remaining 10% is used as the test set. In Table 2, we use a short statistic data set. The out-of-vocabulary (oov) refers to the words which has nothing to do with the train data set.

TABLE II. SETTING OF EXPERIMENTAL DATA SET

train	sent	20793
	word	281960
dev	sent	2607
	word	35200
	oov	985
test	sent	2828
	word	36772
	oov	1139

In the experiment, we use the recall rate, precision rate and F value of Tibetan words as the evaluation index of the model. In order to evaluate the effectiveness of the proposed model, we conduct a series of feature experiments on the test hierarchy.

B. Analysis of experimental results

Five groups of experiments are carried out under the above corpus to verify the effectiveness of CNN-BiLSTM-CRF model in Tibetan word segmentation. Among them, the recall rate, precision rate and F value are used.

Experiment 1. This experiment has two purposes. One is to test the performance of CRF model on the data set by using the CRF model as a benchmark model; the other is to summarize a series of problems found in Tibetan word segmentation using CRF model. In the experiment,

CRF++ toolkit [16] is used which is popular at present. Since the tagging dataset is based on sentences, we merely consider the lexical features for CRF++ toolkit. After using CRF toolkit for word segmentation tasks, we find some problems.

Firstly, Tibetan words which are not included in the training corpus can't correctly be recognized by the CRF model.

Secondly, there are plenty of stick-in forms in Tibetan, but CRF model can't recognize the compact form. It seems like a new character, but it represents two words. For example, “**ཇ**” (my) in “**ཇ****ཇ**” (my mother) is stick-in form. It is a stick-in form formed by word and genitive markers (ཇ), which CRF model cannot accurately identify.

Thirdly, the lack of clear segmentation principles and uncertain segmentation units in Tibetan word segmentation has resulted in incomplete meaning and unreasonable grammatical structure. For example, the separation of “ནས་
တྲୟାଵୁଳ୍ୱା” and “ନୀତ୍ସାପ୍ତିଦ୍ୱାରା” in “ନୀତ୍ସାପ୍ତିଦ୍ୱାରା” (The sky has dawned) is difficult to understand, which leads to unreasonable grammatical structure.

Fourthly, there are a lot of long words in Tibetan, especially some extended place names and long organization names, which make CRF model unable to recognize similar long words accurately. For example, “ **Xinjiang Uygur Autonomous Region**” (Xinjiang Uygur Autonomous Region).

Finally, part of Tibetan corpus collection comes from some forums, microblogs, news and other networks. There are a lot of spelling errors in this text. CRF model can't recognize the misspelled words accurately. Its accuracy rate is 89.97%, recall rate is 91.01% and F value is 90.49%.

Experiment 2. This experiment has two purposes. The one is to study whether the advantages of deep neural network are more than those of the statistical model in Tibetan word segmentation. The other is to study whether the problems found in the CRF statistical model can be solved by the deep neural network.

In the experiment, we use simple RNN model, LSTM model and BiLSTM model to perform Tibetan word segmentation tasks. From Table 3, we can see that the performance of simple RNN model is almost the same as that of CRF model. The performance of LSTM model and BiLSTM model are better than that of CRF model. The F value of BiLSTM model is 2.31% which is higher than that of CRF model.

TABLE III. EXPERIMENTAL RESULTS OF BASELINE NEURAL NETWORK MODEL

model	P (%)	R (%)	F (%)
CRF	89.97	91.01	90.49
RNN	90.46	90.59	90.52
LSTM	90.12	91.74	90.92
BiLSTM	93.12	92.49	92.80

Experiment 3. The purpose of this experiment is to determine whether we add CRF model to LSTM model, and whether BiLSTM model can further improve the performance of Tibetan word segmentation. On the basis of experiment 2, the output of BiLSTM model is input into CRF layer, and the output sequence with the greatest probability is obtained. The experimental results in Table

4 show that the accuracy of LSTM-CRF model and BiLSTM-CRF model are improved after adding CRF layer, and the F value of BiLSTM-CRF model is 1.31% which is higher than that of BiLSTM model.

TABLE IV. EXPERIMENTAL RESULTS AFTER ADDING CRF MODELS

experimental design	P (%)	R (%)	F (%)
LSTM_CRF	93.43	92.56	92.99
BiLSTM_CRF	94.33	93.89	94.11

Experiment 4. The purpose of the experiment is to verify the validity of the CNN model.

CNN model is added to LSTM-CRF model and BiLSTM-CRF model. We use CNN model to acquire character features which are input into LSTM or BiLSTM model for training. The experimental results in Table 5 demonstrate that the performance of Tibetan word segmentation is improved when CNN network is concatenated to LSTM-CRF model and BiLSTM-CRF model. At the same time, we can see that the F value of CNN-LSTM-CRF model is 0.41% which is higher than that of LSTM-CRF model, and the F value of CNN-BiLSTM-CRF model is 1.06% which is higher than that of BiLSTM-CRF model.

TABLE V. COMPARISON OF EXPERIMENTAL RESULTS AFTER CONCATENATING CNN MODEL

experimental design	P (%)	R (%)	F (%)
CNN-LSTM-CRF	93.62	93.18	93.40
CNN-BiLSTM-CRF	94.71	95.64	95.17

Experiment 5. The purpose of the experiment is to verify the validity of the CNN model. Our operating system is CentOS Linux release 7.6.1810 (Core), Intel Xeon E5-2620 CPU, and 12GB graphics card is Tesla K40C GPU. With the concatenation of CNN model and highway model, the training speed of the model has been significantly improved, which can be observed in the experimental results in Table 6. The training speed of CNN-BiLSTM-CRF model with CNN model and highway model has been improved significantly, compared with BiLSTM-CRF model. The training speed is increased by 2.5 hours and the test speed is increased by 1.06 seconds. Regardless of the accuracy of word segmentation, the test is to calculate the time spent on segmenting the Tibetan text by loading the models that have been trained respectively.

TABLE VI. THE INFLUENCE OF CNN MODEL ON EXPERIMENTAL SPEED

experimental design	training time (h)	testing time (s)
LSTM-CRF	10.7	3.05
BiLSTM-CRF	11.5	3.18
CNN-LSTM-CRF	8.5	2.36
CNN-BiLSTM-CRF	9.0	2.12

V. CONCLUDING REMARKS

This paper takes the traditional CRF statistical model as the benchmark to perform experiments. We summarize the problems in Tibetan word segmentation to construct a

neural network model based on CNN-BiLSTM-CRF framework. The model captures character level feature vectors in CNN layer, obtains the past and future context information of current words in BiLSTM layer, decodes the output of BiLSTM layer in CRF layer, and finally outputs the optimal tag sequence. Experiments based on the constructed corpus further verify the effectiveness of the CNN-BiLSTM-CRF framework for Tibetan word segmentation. The Tibetan text classification and the Tibetan speech synthesis system which combined with the word segmentation results achieved the best effect. Subsequently, we will further improve the corpus, add part-of-speech tagging feature information, and test the performance of Tibetan word segmentation and part-of-speech tagging based on the original corpus.

ACKNOWLEDGMENT

This research has received funding from the academic requirements for the National Science Foundation of China (NSFC) under grant No.11664036, No.31860285 and No.31660281 and High School Science and Technology Innovation Team Project of Gansu (2017C-03). We also want to thank the reviewers for their thoughtful comments and efforts towards improving our paper.

REFERENCES

- [1] D. Cai and H. Zhao, "Neural word segmentation learning for chinese," vol. 1, 06 2016, pp. 409–420.
- [2] L. Huidan, N. Minghua, Z. Weinan, W. Jian, and H. Yipin, "Segta practical tibetan word segmentation system," journal Of chinese information processing, vol. 26, no. 1, pp. 97–104, 2012.
- [3] K. Cajun, L. Congjun, and J. Di, "Segmentation of tibetan abbreviated forms based on word position," Computer Engineering and Applications, vol. 50, no. 11, pp. 218–222, 2014.
- [4] L. Karten, Y. Yuanyuan, and Z. Xiaobin, "Tibetan automatic word segmentation based on conditional random fields and knowledge fusion," journal Of chinese information processing, vol. 29, no. 6, pp. 213–219, 2015.
- [5] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware neural language models," Computer Science, 2015.
- [6] Long congjun and liu huidan, Research on the Theory and Method of Tibetan Automatic Word Segmentation. Intellectual Property Publishing House, 2016.
- [7] F. Dernoncourt, J. Young Lee, and P. Szolovits, "Neuroner: an easy-to-use program for named-entity recognition based on neural networks," 05 2017.
- [8] X. Ma, "End-to-end sequence labeling via bi-directional lstm-cnns-crf," 03 2016.
- [9] R. Kumar Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," 05 2015.
- [10] Y. Yao and Z. Huang, "Bi-directional lstm recurrent neural network for chinese word segmentation," 02 2016.
- [11] C. Kang, D. Jiang, and C. Long, "Tibetan word segmentation based on word-position tagging," 08 2013, pp. 239–242.
- [12] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," International Conference on Learning Representations, 12 2014.
- [13] Z. Yang, R. Salakhutdinov, and W. Cohen, "Multi-task cross-lingual sequence tagging from scratch," 03 2016.
- [14] T. Mikolov, M. Karafiat, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," vol. 2, 01 2010, pp.1045–1048.
- [15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," Journal of Machine Learning Research, vol. 15, pp. 1929–1958, 06 2014.

- [16] Y. Leng, W. Liu, S. Wang, and X. Wang, “A feature-rich crf segmenter for chinese micro-blog,” vol. 10102, 12 2016, pp. 854- 861.

Separate Answer Decoding for Multi-class Question Generation

Kaili Wu, Yu Hong, Mengmeng Zhu, Hongxuan Tang, Min Zhang

School of Computer Science and Technology, Soochow University

Suzhou, China

{klwu,mmzhu,hxtang}@stu.suda.edu.cn,tianxianer@gmail.com,minzhang@suda.edu.cn

Abstract—Question Generation (QG) aims to automatically generate questions by understanding the semantics of source sentences and target answers. Learning to generate diverse questions for one source sentence with different target answers is important for the QG task. Despite of the success of existing state-of-the-art approaches, they are designed to merely generate a unique question for a source sentence. The diversity of answers fail to be considered in the research activities. In this paper, we present a novel QG model. It is designed to generate different questions toward a source sentence on the condition that different answers are regarded as the targets. Pointer-Generator Network(PGN) is used as the basic architecture. On the basis, a separate answer encoder is integrated into PGN to regulate the question generating process, which enables the generator to be sensitive to attentive target answers. To ease the reading, we name our model as APGN for short in the following sections of the paper. Experimental results show that APGN outperforms the state-of-the-art on SQuAD split-1 dataset. Besides, it is also proven that our model effectively improves the accuracy of question word prediction, which leads to the generation of appropriate questions.

Keywords-Question Generation; Target Answer; Question Word;

I. INTRODUCTION

Question Generation (QG) is an important task in Question Answering (QA). Rus et al. [1] gave the definition of the QG in 2010, that is, given a free text, automatically generate a natural question according to the text. Since then, relevant researches have been conducted and applied to medical and educational fields [2], [3]. In addition, the QG task can generate a large number of question-answer pairs, which can be used to expand the corpus of the QA task and assist in the construction of the QA system.

Previous work for the QG task mainly utilized manually written rules to transform a sentence into a question [4], [2], [5]. However, the method is usually applied to a specific field and difficult to process large-scale data. Compared to the rule-based method, neural network-based method [6], [7], [8] rely on no rules, and it is in an end-to-end fashion driven by large-scale data. Even though, the shortcoming of this method is there are still gaps with natural questions in the expression.

In this paper, we focus on sentence-level QG task which takes a declarative sentence and a target answer as inputs and generates a related question. By analyzing the datasets, we find a phenomenon a source sentence may contain several target answers which are corresponding to different questions. As shown in Example 1, there are two target answers underlined in the source sentence. Our baseline

model PGN [9] can only generate the same question (“PGN Q1”, “PGN Q2”) for the source sentence (“S”). As we can see that “*Jerome Green*” is the answer to this question, but “*Around 1899*” cannot be used to answer the question. To overcome the limitations, we fuse target answers based on PGN. Concretely, we encode separately the target answer and initialize the decoder state. “APGN Q1” and “APGN Q2” are generated questions of our model. Those questions are close to the standard questions “Q1” and “Q2”.

Example 1:

S: Around 1899, professor Jerome Green became the first American to send a wireless message.

Q1: In what year did Jerome Green send his first wireless message?

PGN Q1: Who became the first American to send a wireless message?

APGN Q1: When did Jerome Green become the first American to send a wireless message?

Q2: Which professor sent the first wireless message in the USA?

PGN Q1: Who became the first American to send a wireless message?

APGN Q2: Who became the first American to send a wireless message?

We propose a simple and effective method for improving QG performance. Moreover, we alleviate the problem in which a source sentence contains several target answers. Furthermore, we find our proposed method help to generate correct question words. These question words will generate subsequence words in the right way.

II. RELATED WORK

QG task has been mainly tackled with two types of methods: rule-based and neural network-based. Rule-based QG depends on deep linguistic knowledge and well-designed rules for transforming declarative sentences to questions. Lindberg et al. [4] proposed a complex template-based method combined with semantic tagging information. Heilman et al. [2] overgenerated questions by using manually written rules. Then these questions were ranked by a logistic regression model. In addition, Liu et al. [5] used lexical and syntactic information to generate questions in Chinese. Rule-based methods have low universality and rely heavily on rules. Generally, rule-based methods pay more attention to the syntactic roles of words, but not their semantic roles.

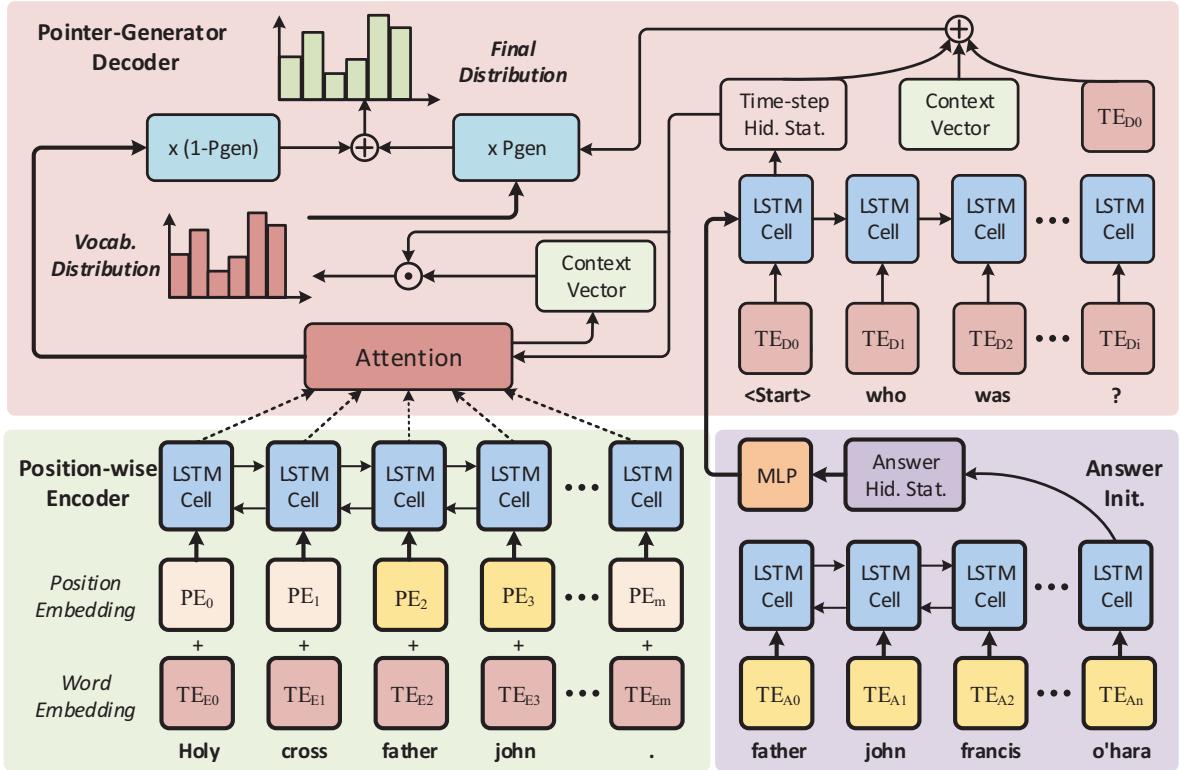


Figure 1. The APGN architecture. We use the separate answer encoder to get the representation of the answer. Then, we use it to initialize the decoder.

To solve the problem of rule-based methods, Du et al. [6] introduced a deep Seq2seq model to generate questions for the sentences extracted from reading comprehension datasets. The method achieved better performance than rule-based methods in both automatic and human evaluations. Zhou et al. [7] added lexical features (POS, NER) and answer positions to enrich the encoder. Moreover, they used copy mechanism to generate words in the source sentence. Dong et al. [8] predicted the types of the target answers by using classification model and then added to generate questions.

The end-to-end neural network can make the model self-adaptive learning and avoid complex rules. In addition, the release of large-scale reading comprehension datasets provides more corpus resources for QG task and promotes the development of the neural network-based QG methods. Our method is based on PGN which is an attention-based Seq2seq model with copy mechanism.

III. TASK DEFINITION

Given a sentence $X = \{x_1, x_2, \dots, x_m\}$ and a target answer $A = \{a_1, a_2, \dots, a_n\}$, A are continuous tokens of X , i.e. $A \subset X$. The QG task aims to generate a natural question Y related to the information of X and A . The QG task is defined as finding \bar{y} :

$$\bar{y} = \arg \max_y P(y|X, A) \quad (1)$$

where $P(y|X, A)$ denotes the log-likelihood conditioned on X and A .

IV. MODEL

We use PGN as our baseline model. In order to generate different questions to diverse target answers in one source sentence, our model incorporates target answers and combines with source sentences in the decoder. Concretely, we encode a target answer separately to get the representation. Then, we use the representation to initialize the decoder. So, target answers and source sentences work together in the decoder. Fig. 1 exhibits the model overview.

A. Baseline:Pointer-Generator Network (PGN)

1) Encoder: We use a singer-layer Bidirectional Long Short-Term Memory (Bi-LSTM) [10] to encode the source sentence X . It produces a sequence of hidden states $o = \{o_1, o_2, \dots, o_m\}$ through the layer. Each hidden state concatenates the forward and the backward LSTM representation:

$$\overrightarrow{o_i} = \overrightarrow{\text{LSTM}}(x_i^e, \overrightarrow{o_{i-1}}) \quad (2)$$

$$\overleftarrow{o_i} = \overleftarrow{\text{LSTM}}(x_i^e, \overleftarrow{o_{i+1}}) \quad (3)$$

$$o_i = [\overrightarrow{o_i}, \overleftarrow{o_i}] \quad (4)$$

where x_i^e is word-level embedding at step i .

2) *Decoder*: The decoder serves as an attention-based LSTM layer with copy mechanism. At each decoding step t, the last hidden state h_{t-1} and generated word embedding x_t^e are fed to obtain current step hidden state h_t . Then we use attention mechanism [11] to compute the context vector c_t , which denotes a fixed-size representation of the importance score e_i^t between current hidden state h_t and each encoder hidden state o_i . The e_i^t are normalized by softmax function, and then get the c_t via weight sum:

$$e_i^t = v^T \tanh(W_o o_i + W_h h_t + b_{attn}) \quad (5)$$

$$a^t = \text{softmax}(e^t) \quad (6)$$

$$c_t = \sum_i^m a_i^t o_i \quad (7)$$

where v^T , W_o , W_h , and b_{attn} are learnable parameters.

Then we combine the current state h_t and context vector c_t to predict next word with a softmax layer over the vocabulary:

$$P_{vocab}(W) = \text{softmax}(V'(V[h_t, c_t] + b) + b') \quad (8)$$

where V' , V , b , and b' are learnable parameters.

We employ copy mechanism [9] into our model. For each decoder step, p_{gen} denotes the probability of the word from the vocabulary, versus copying word from source sentence. At current step t, we use a non-linear layer with the current state h_t , context vector c_t and word embedding x_t^e to calculate p_{gen} :

$$p_{gen} = \sigma(W_c c_t + W'_h h_t + W_x x_t^e + b_p) \quad (9)$$

where W_c , W'_h , W_x , and b_p are learnable parameters, σ is sigmoid activation function.

Finally, we compute the final probability distribution with p_{gen} :

$$P(W) = p_{gen} P_{vocab}(W) + (1 - p_{gen}) \sum_{i:w_i=w} a_i^t \quad (10)$$

When the generated word is “STOP” or attain the maximum decoding length, the question is generated completely.

B. Separate Target Answer Encoder

In the model, we encode target answer A separately and use the final step hidden state as the representation of the target answer A . Then we initialize the decoder hidden state with the representation to improve the accuracy of the generated question. We use pre-trained word embedding to convert A into vectors $A^e = \{a_1^e, a_2^e, \dots, a_n^e\}$ with $a_i^e \in \mathbb{R}^{d_e}$, where d_e is the dimensionality of word-level target answer embedding. To capture more context information, we use Bi-LSTM to encode A^e . Each hidden state is a concatenation of a forward and a backward LSTM representation:

$$\vec{s}_i = \overrightarrow{\text{LSTM}}(a_i^e, \vec{s}_{i-1}) \quad (11)$$

$$\overleftarrow{s}_i = \overleftarrow{\text{LSTM}}(a_i^e, \overleftarrow{s}_{i+1}) \quad (12)$$

Table I
THE NUMBER OF DATASETS

Dataset	Train	Dev	Test
split-1	70484	10570	11877
split-2	86607	8964	8963
MS MARCO	93267	10363	8770

$$s_i = [\vec{s}_i, \overleftarrow{s}_i] \quad (13)$$

where the Bi-LSTM does not share parameters with (2) and (3).

The final step hidden state s_n contains previous context information, so it is used to represent the target answer. In order to make full use of the target answer information, we use a fully connected layer to get deeper semantic representation:

$$new_s_n = \text{Relu}(W_s s_n + b_s) \quad (14)$$

where W_s and b_s are learnable parameters, Relu is an activation function. new_s_n denotes the new target answer representation and is used to initialize decoder hidden state.

C. Combination

Decoder Initialization We use the target answer representation new_s_n to initialize the decoder hidden state:

$$h_0 = new_s_n \quad (15)$$

Then, the process of the decoder is modified. Because of the initialization, the baseline model PGN (5)-(10) incorporate information of target answers. Besides, the information of target answers and source sentences will work together better.

V. EXPERIMENTAL SETUP

A. Datasets

We conduct the experiments on SQuAD [12] and MS MARCO [13]. To be fair, we use the same datasets that were used by previous work. SQuAD has two division that are denoted as *split-1* and *split-2*.

SQuAD The original SQuAD consists of 536 articles from Wikipedia and more than 100k question-answer pairs posed about the articles by crowd-workers. Answers are sub-spans in the articles.

split-1 Du et al. [6] extracted sentences and paired them with the questions, and then re-divided them into train/dev/test splits. Because we use target answers, we extract them from sentences.

split-2 Zhou et al. [7] extracted sentence-answer-question triples to build train, dev and test sets.

MS MARCO In this dataset, there are 1,010,916 questions which sampled from real anonymized user queries. Each question has corresponding answers and passages from real web documents. We extract a subset of MARCO data where answer is a sub-span within the article. We use dev set as test set, and split train set into train and dev set with ratio 9:1.

We tokenize datasets with Stanford CoreNLP [14] and then lower-case them. Table I shows the number of datasets.

B. Implementation Details

We set the cutoff length of the input sentence as 100 words and the question as 30 words. We use 50,000 most frequent words that appeared in train set as vocabulary in both source and target. We use 300-dimensional pre-trained Glove [15] embedding and 16-dimensional randomly initialized position embedding. The hidden size of both the encoder and decoder is 256. During training, we use Adam [16] optimizer with learning rate 0.0005. When testing, we conduct beam search with beam size of 4.

C. Evaluation

We use the evaluation package released by Chen et al. [17]. The package includes BLEU-4 [18], METEOR [19] and ROUGE_L [20].

VI. RESULTS AND ANALYSIS

A. Main Results

To prove the effectiveness of our model, we compare it with several competitive systems. Next, we briefly introduce their approaches and experimental settings.

Du [6] is an attention-based Seq2seq model and does not use the target answer information.

NQG++ [7] is a Seq2seq model with copy mechanism. It uses Bi-GRU to encode the concatenation of word embedding, answer position and lexical features (POS, NER). Answer positions use BIO scheme.

M2S+cp [21] consists of a multi-perspective encoder and a decoder with the copy mechanism. It uses multi-perspective context matching algorithm to detect whether each source word belongs to the relevant context of the answer.

Sun [22] is a hybrid model based on Pointer-Generator network. Answer-focused model generates a question word in a restricted vocabulary of question words. Position-aware model incorporates word position embeddings to gain position-aware attention for further generation.

s2s-a-ct-mp-gsa [23] is a Seq2seq model with answer tagging, gated self-attention and maxout pointer mechanism.

Ass2s [24] is an answer-separated Seq2seq which masks answer tokens with “a” and a keyword-net which extracts key information from target answer.

Table II shows results on split-1 and split-2, and Table III is the MS MARCO result. Note that the results is the final which uses position feature. Our model performance achieves the start of the art on split-1. In addition, BLEU-4 is 0.98% higher than the PGN on MS MARCO as shown in Table III. Expect Du, other systems use position feature. In our model, we propose a simpler way to use target answer position information than other competitive systems and achieve higher performance. Note that we do not use lexical features in our model when we experiment on split-2.

Table II
PERFORMANCE COMPARISON(%).

Model	split-1			BLEU-4
	BLEU-4	METEOR	ROUGE _L	
Du	12.28	16.62	39.75	-
NQG++	-	-	-	13.29
M2S+cp	13.98	18.77	42.72	13.91
Sun	-	-	-	15.64
s2s-a-ct-mp-gsa	15.32	19.29	43.91	15.82
Ass2s	16.20	19.92	43.96	16.17
APGN	17.05	20.53	44.06	15.16

Table III
THE PERFORMANCE ON MS MARCO(%)

	BLEU-4	METEOR	ROUGE _L
PGN	8.18	22.15	35.68
APGN	9.16	23.61	38.21

B. Discussion

1) *Impact of A Source Sentence with Different Target Answers:* In this section, we statistics the number of different target answers in the one source sentence as shown in Table IV. “Difference” represents the number of target answers in one source sentence. “Count” represents the number of source sentences. In the test dataset, there are almost 70% source sentences that have different target answers.

Table IV
THE NUMBER OF DIFFERENT TARGET ANSWERS IN ONE SENTENCE

Difference	Count	Difference	Count
1	3557	7	21
2	1882	8	12
3	683	9	4
4	304	10	2
5	143	12	1
6	42	13	1

To verify the effectiveness of our method, we calculate the BLEU-4 scores of PGN and our model towards the source sentences mentioned above. Our model gains 34.54% BLEU-4 score which is 7.06% higher than PGN

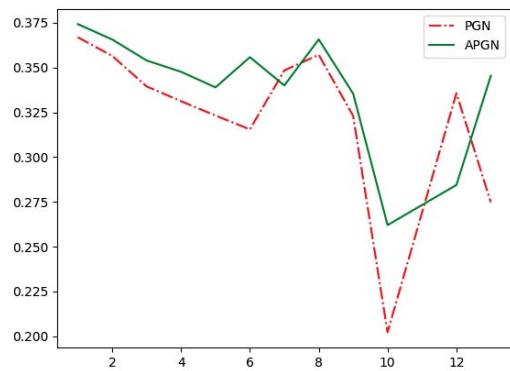


Figure 2. The BLEU-4 scores

(27.48%). To make it clearer, we draw a line chart as shown in Fig. 2. Generally, our model is higher than the PGN. It illustrates our proposed method can generate different questions to a source sentence with diverse target answers.

2) Reasonability of Separate Target Answer Encoder to Initialize the Decoder: To prove the reasonability of decoder initialization with a separate target answer encoder, we design the following experiments. Firstly, we encode target answers sharing parameters with source sentences encoder. Secondly, we add the representation of the target answer in the decoding step except for the first. The results show in Table V.

Table V
THE BLEU-4 SCORES (%)

Model	BLEU-4
PGN	15.23
PGN+share	15.60
PGN+add_init	1.07
APGN	16.13

As shown in Table V, the separate encoder is better than the sharing parameters one. The reason is that encoding separately makes the model more focused on the target answer, thus generating its corresponding question. For initialization, only initialize the first state of the decoder is valid. We think adding the representation of the target answer at each step can greatly mislead the generation process, resulting in lower performance.

3) Question Words: In this section, we analyse the distribution of generated question words. A question word represents the type of a target answer in one source sentence. We select common question words from split-1 test set and statistic proportion in target questions (TGT), PGN and APGN. As shown in Table VI, it can be observed that our APGN is closer to the percentage of question words on TGT. To make it clearer, we draw a chart (see Fig. 3) that shows the difference of question words number between APGN, PGN and TGT respectively.

Table VI
THE PERCENTAGE OF QUESTION WORDS ON TGT, PGN AND APGN

Question Word	TGT(%)	PGN(%)	APGN(%)
What	56.41	65.03	62.81
How	12.15	11.18	12.55
Who	11.31	10.31	14.06
Which	8.55	0.15	1.27
When	5.57	7.54	4.61
Where	3.76	5.54	3.92
Why	0.78	0.18	0.24
Others	1.47	0.07	0.54

As shown in Fig. 3, expect for “who” type, the difference of question words number between APGN and TGT is smaller than that between PGN and TGT. It further illustrates incorporating the target answer helps to capture the question word corresponding to the target answer, thus ensuring the generated question is used to ask the target answer.

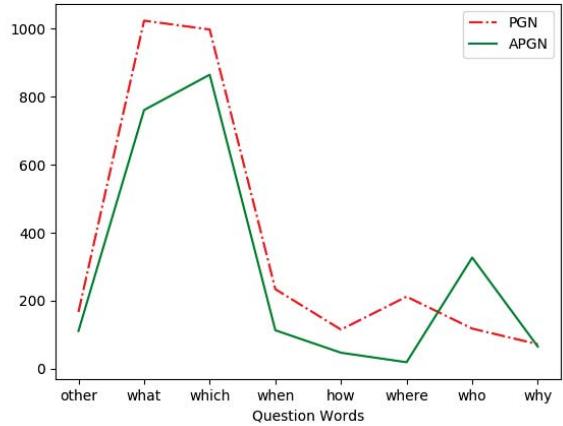


Figure 3. The difference of question words number between PGN, APGN and TGT respectively.

4) Impact of Generation Process: In general, the first token of a question is a question word. In the last section, we have proved our proposed method improves the accuracy of question words. In this section, we analyse the impact of question words to subsequence words of the generation process.

We force the first generated word to be correct in the decoder for baseline model PGN, then we evaluate those generated questions. We find the BLEU-4 is 2.27% higher than PGN. Thus, once question words are generated correctly, the following process of generation will be better.

To some extent, our proposed method has a good effect on the subsequence generation process by improving the accuracy of question words. We think this is because we have determined the type of question by the initialization of the target answer. Because of the characteristic of the LSTM, correctness will be propagated in the following generation.

5) Position Feature: In the model, we concatenate word-level embedding with position embedding. We use “01” shame to label words in the source sentences. Label 1 denotes the word is in the target answer. In contrast, label 0 denotes the word is not in the target answer. Then, the answer position labels are embedded in real-valued vectors with the dimensionality of 16.

VII. CONCLUSION

In this paper, we propose encoding separately target answers to get the representation. Then, we use this representation of the target answer to initialize the decoder. Through analysis of generated questions, we find our proposed method can generate different questions to a source sentence that has different target answers. Besides, we describe the reasonability of a separate target answer encoder to initialize the decoder. We also find our method can generate question words more correctly than PGN. Once the question word is generated correctly, the subsequence of the question will be better.

ACKNOWLEDGMENT

This research work is supported by National Natural Science Foundation of China (Grants No.61672368, No.61703293 and 2017YFB1002104). The authors would like to thank the anonymous reviewers for their insightful comments and suggestions. Yu Hong, Professor Associate in Soochow University, is the corresponding author of the paper, whose email address is tianxianer@gmail.com.

REFERENCES

- [1] V. Rus, B. Wyse, P. Piwek, M. Lintean, S. Stoyanchev, and C. Moldovan, “The first question generation shared task evaluation challenge,” in *Proceedings of the 6th International Natural Language Generation Conference*. Association for Computational Linguistics, 2010, pp. 251–257.
- [2] M. Heilman and N. A. Smith, “Good question! statistical ranking for question generation,” in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 609–617.
- [3] J. Weizenbaum, “Eliza—a computer program for the study of natural language communication between man and machine,” *Communications of the ACM*, vol. 9, no. 1, pp. 36–45, 1966.
- [4] D. Lindberg, F. Popowich, J. Nesbit, and P. Winne, “Generating natural language questions to support learning online,” in *Proceedings of the 14th European Workshop on Natural Language Generation*, 2013, pp. 105–114.
- [5] M. Liu, V. Rus, and L. Liu, “Automatic chinese factual question generation,” *IEEE Transactions on Learning Technologies*, vol. 10, no. 2, pp. 194–204, 2017.
- [6] X. Du, J. Shao, and C. Cardie, “Learning to ask: Neural question generation for reading comprehension,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2017, pp. 1342–1352.
- [7] Q. Zhou, N. Yang, F. Wei, C. Tan, H. Bao, and M. Zhou, “Neural question generation from text: A preliminary study,” in *National CCF Conference on Natural Language Processing and Chinese Computing*. Springer, 2017, pp. 662–671.
- [8] X. Dong, Y. Hong, X. Chen, W. Li, M. Zhang, and Q. Zhu, “Neural question generation with semantics of question type,” in *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer, 2018, pp. 213–223.
- [9] A. See, P. J. Liu, and C. D. Manning, “Get to the point: Summarization with pointer-generator networks,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2017, pp. 1073–1083.
- [10] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [11] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.
- [12] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “SQuAD: 100,000+ questions for machine comprehension of text,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2016, pp. 2383–2392.
- [13] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng, “Ms marco: A human generated machine reading comprehension dataset,” in *Proceedings of 30th Conference on Neural Information Processing Systems*, 2016.
- [14] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, “The Stanford CoreNLP natural language processing toolkit,” in *Association for Computational Linguistics (ACL) System Demonstrations*, 2014, pp. 55–60.
- [15] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.
- [16] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [17] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick, “Microsoft coco captions: Data collection and evaluation server,” *arXiv preprint arXiv:1504.00325*, 2015.
- [18] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.
- [19] M. Denkowski and A. Lavie, “Meteor universal: Language specific translation evaluation for any target language,” in *Proceedings of the ninth workshop on statistical machine translation*, 2014, pp. 376–380.
- [20] C.-Y. Lin, “Rouge: A package for automatic evaluation of summaries,” *Text Summarization Branches Out*, 2004.
- [21] L. Song, Z. Wang, W. Hamza, Y. Zhang, and D. Gildea, “Leveraging context information for natural question generation,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018, pp. 569–574.
- [22] X. Sun, J. Liu, Y. Lyu, W. He, Y. Ma, and S. Wang, “Answer-focused and position-aware neural question generation,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3930–3939.
- [23] Y. Zhao, X. Ni, Y. Ding, and Q. Ke, “Paragraph-level neural question generation with maxout pointer and gated self-attention networks,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3901–3910.
- [24] Y. Kim, H. Lee, J. Shin, and K. Jung, “Improving neural question generation using answer separation,” *arXiv preprint arXiv:1809.02393*, 2018.

Converting an Indonesian Constituency Treebank to the Penn Treebank Format

Jessica Naraiswari Arwidarasti, Ika Alfina, and Adila Alfa Krisnadhi

Faculty of Computer Science, Universitas Indonesia

Depok, Indonesia

Email: jessica.naraiswari91@ui.ac.id, ika.alfina@cs.ui.ac.id, adila@cs.ui.ac.id

Abstract—A constituency treebank is a key component for deep syntactic parsing of natural language sentences. For Indonesian, this task is unfortunately hindered by the fact that the only one constituency treebank publicly available is rather small with just over 1000 sentences, and not only that, it employs a format incompatible with readily available constituency treebank processing tools. In this work, we present a conversion of the existing Indonesian constituency treebank to the widely accepted Penn Treebank format. Specifically, the conversion adjusts the bracketing format for compound words as well as the POS tagset according to the Penn Treebank format. In addition, we revised the word segmentation and POS tagging of a number of tokens. Finally, we performed an evaluation on the treebank quality by employing the Shift-Reduce parser from Stanford CoreNLP to create a parser model. A 10-fold cross-validated experiment on the parser model yields an F1-score of 70.90%.

Keywords-constituency parsing; Indonesian; Penn Treebank; Stanford parser; treebank format

I. INTRODUCTION

In Natural Language Processing (NLP) research, syntactic parsing plays a major role in determining the semantics of the sentences. Syntactic parsing is a key prerequisite subtask for many general NLP tasks such as machine translation, grammar checking, natural language generation, information extraction, and question answering system.

In general, syntactic parsing can be divided into two classes, phrase-structure or constituency parsing and dependency parsing [1]. Constituency parsing aims to parse a sentence into its constituent phrases. Meanwhile, dependency parsing is used to uncover the dependency relation between words in the sentence.

Figure 1 shows an example of a constituency tree for the sentence "Bus menabrak sebuah motor." ("A bus hit a motorcycle."). The root *S* represents that this sentence is a simple declarative clause. This sentence is parsed into two parts: an NP-SBJ (Noun Phrase as subject) for the segment "Bus" (*bus*) and a VP (Verb Phrase) for the segment "menabrak sebuah motor" (*hit a motorcycle*). The VP is then parsed into two further parts: a terminal with Part-of-Speech (POS) tag of VB for "menabrak" (*hit*) and an NP (Noun Phrase) for the segment "sebuah motor" (*a motorcycle*). The latter phrase is the parsed into two terminals: "sebuah" (*a*) with POS tag DT (determiner) and "motor" (*motorcycle*) as NN (noun).

The availability of syntactic corpus (treebank) is very important to build a parser model using data-driven parsing method. For English, various treebanks are available,

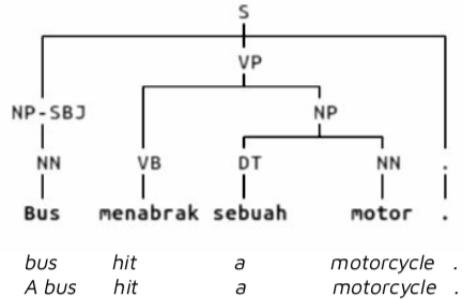


Figure 1. A constituency tree

among which is the Penn Treebank [2]. The annotation format of this treebank is considered the standard format in building constituency treebanks.

Various tools to process the treebanks with the Penn Treebank format are readily available, such as the Stanford Parser [3], the Trance Parser [4], and Discodop [5]. With those tools, we can train treebanks of any language to build a constituency parser model.

On the other hand, resources for syntactic parsing for Indonesian are very limited, despite being a language spoken by more than 260 million people. The only constituency treebank available was developed by Universitas Indonesia¹ in 2015 as the continuation of the development of their POS-tagger corpus [6]. This treebank, hereinafter called Universitas Indonesia Constituency Treebank (UI-CTB), consists of 1,030 sentences containing 27,115 words and was annotated manually. It adopted the Penn Treebank format with some differences in the bracketing format and the POS tagset.

The main difference between UI-CTB and the Penn Treebank format is in the handling of compound words. A compound is "*a combination of two simple words which come together to form a complex word*" [7]. In Indonesian there are three ways to write the compound words: 1) as a single word, such as *kacamata* (eyeglasses); 2) hyphenated, such as *pemuda-pemudi* (youngsters); and 3) as multiple tokens, such as *rumah sakit* (hospital). The bracketing format of UI-CTB was designed to maintain the last type of compound word as a single unit in the treebank. Table I shows some examples of how UI-CTB represents the compound words.

While the bracketing format used by UI-CTB retains the meaning of the compound words, this format unfortunately

¹<https://github.com/famrashel/idn-treebank>

Table I
THE REPRESENTATION OF COMPOUND WORDS IN UI-CTB.

Compound	Representation in UI-CTB
<i>tanggung jawab</i> (responsibility)	(NN (tanggung jawab))
<i>bertanggung jawab</i> (to be responsible)	(VB (bertanggung jawab))
<i>luar biasa</i> (excellent)	(JJ (luar biasa))
<i>sama sekali</i> (at all)	(RB (sama sekali))
<i>salah satu</i> (one of)	(CD (salah satu))
<i>salah seorang</i> (one of)	(CD (salah seorang))
<i>lagi pula</i> (moreover)	(CC (lagi pula))
<i>di mana</i> (where)	(IN (di mana))

does not comply with the expected input format of various constituency treebank processing tools. These tools, in general, expected input in the Penn Treebank format for which compound words are treated rather differently. This format will be explained in Section 2.

To overcome this problem, earlier works that utilized UI-CTB to build Indonesian constituency parser converted the treebank so that compound words become one token and adjusted the bracketing format to the Penn Treebank format. For example, Filino & Purwarianti [8] and [9] opted to change the representation of "*luar biasa*" from (*JJ (luar biasa)*) to (*JJ luar_biasa*) by combining the individual tokens into a single token by using an underline character. We consider this solution oversimplified since there are no such words in real sentences.

The objective of our work is thus to convert the UI-CTB so that it conforms to the Penn Treebank format, both the bracketing format and the POS tagset. We have made the new treebank public² so that it can be used by other studies on Indonesian syntactic parsing. To evaluate the quality of the treebank, we used it to build an Indonesian constituency parser model using Stanford Parser and got F1-score of 70.90%.

The rest of this paper is organized as follows: Section 2 discusses the related work; Section 3 presents the proposed conversion method; Section 4 is for experiments using Stanford Parser, and finally, Section 5 presents the conclusion and future work.

II. RELATED WORK

In this section, we discuss the Penn Treebank format, UI-CTB, Stanford parser, and Shift-Reduce parser.

A. The Penn Treebank format

The Penn Treebank [2] is a constituency treebank for English built by University of Pennsylvania in 1993. It contains around 7 million words with Part-of-Speech (POS) tags and around 3 million words with syntactic tags [10]. The Penn Treebank annotation (hereinafter referred to as PTB) used the POS tagset of 48 labels and the syntactic tagset of 14 labels. The annotation scheme of PTB is considered as the reference standard in creating the constituency treebanks for English and other languages.

Unfortunately, we could not find any published work discussing the procedure employed by PTB to annotate compound words. We thus resort to direct observation to examples of compound nouns and verbs in PTB to gain some understanding on how PTB generally represents compound words. Table II lists some of these examples. From the observation, we found that:

- For the compound nouns, each token is labeled as NN and all tokens are wrapped in the NP (Noun Phrase) syntactic tag. For example, for *paper work* that is a compound noun, *paper* and *work* are labeled as NN and both tokens are wrapped in an NP syntactic tag.
- For the compound verbs, the second token is labeled with POS tag of RP (particle) and this token also wrapped out with a special syntactic tag of *PRT*. Then, all tokens are wrapped in a VP (Verb Phrase) syntactic tag.

Table II
THE REPRESENTATION OF COMPOUND WORDS IN PTB.

Compound	Representation in PTB
<i>paper work</i>	(NP (NN paper) (NN work))
<i>point out</i>	(VP (VBP point) (PRT (RP out)))
<i>picking up</i>	(VP (VBG picking) (PRT (RP up)))
<i>took away</i>	(VP (VBD took) (PRT (RP away)))

B. UI Constituency Treebank

UI Constituency Treebank (UI-CTB) was built in 2015, as the continuation of a project to build a POS tagger corpus by Dinakaramani et al. [6]. This POS tagger corpus consists of 10,000 sentences containing 262,330 lexical tokens that had been annotated manually. The sentences of this corpus are the first 10,000 sentences taken from IDENTIC corpus that built by Larasati in 2012 [11]. IDENTIC consists of 45,000 Indonesian-English parallel sentences. In [11], it was stated that some parts of the IDENTIC corpus are taken from the corpus from PAN Localization Project³ [12].

Although UI-CTB annotation format is similar to the PTB format, there are some notable differences. First, there are differences in bracketing format as explained in the earlier section, and second, it uses a different Part-of-Speech (POS) tagset. UI-CTB POS tagset consists of 23 tags [6], while PTB has a total of 48 POS tags (36 POS tags plus 12 other tags for punctuation and currency symbols).

As far as we know, UI-CTB has been utilized in two earlier works. In 2016, Filino & Purwarianti [8] used UI-CTB as the basis for building an Indonesian constituency parser model using Trance parser, which is a shift-reduce neural constituent parser employing recurrent neural networks [4]. In that work, 978 trees were used out of the total of 1,030 trees in UI-CTB as trees that do not represent complete sentences were excluded. Some preprocessing steps were done to the UI-CTB, such as deletion of *U* symbol, modification of compound words, deletion of null

²<https://github.com/ialfina/kethu>

³http://www.panl10n.net/indonesia/#Linguistic_Resources

element, deletion of null subordinate conjunction, and trees normalization. Also, the compound words are joined into a single token by using the underline character. They achieved an F1-score of 74.91% when using parameter of hidden state vector size of 128 and word representation vector size of 2048.

The second work that utilized UI-CTB is from Kurianwan [9]. He built an Indonesian constituency parser from UI-CTB using Recurrent Neural Network Grammars (RNNG) [13]. All trees in the treebank were used in this work and compound words are treated similarly as in Filino & Purwarianti's work [8]: they are joined into a single token using the underline character. From several scenarios, it was reported that the highest F1-score of 78.63% was achieved when using Generative RNNG with a pre-trained word embedding.

C. Stanford Parser

Stanford Parser is a part of Stanford CoreNLP [3]. For constituency parsing, Stanford Parser provides three classifiers to create the parser model: 1) Probabilistic Context-Free Grammar (PCFG) parser; 2) Recursive Neural Network (RNN) parser; and 3) Shift-Reduce Parser.

PCFG parser was built by Klein and Manning [14] in 2003. The RNN parser was made by Socher et al. [15] in 2011, and the Shift-Reduce parser was written by John Bauer based on the works of Zhu et al. [16] and others. Based on experiments run in 2014, when tested on PTB, it was reported that the F1-score of those three parses are: 85.54% for the PCFG parser, 88.55% for the Shift-Reduce parser, and 89.96% for the RNN parser.

In this work, we only utilized the Shift-Reduce parser of the Stanford CoreNLP to evaluate the new treebank.

D. Shift-Reduce Parser

The Shift-Reduce Parser performs the parsing by maintaining a state of the current parsed tree, with the words of the sentence on a queue and partially completed trees on a stack. The parser applies the transitions to the state until the queue is empty and the current stack only contains a finished tree. According to Zhu et al. [16], the Shift-Reduce (SR) parser was developed based on the shift-reduce process proposed by Sagae & Lavie [17] and employs global perceptron training and beam search.

The shift-reduce parsing is based on a scan of the input sentence. A transition action is applied to consume an input word or construct a new phrase-structure at each step. A stack is used to maintain partially constructed phrase structures, while the input words are stored in a buffer. The set of transitions are [16]:

- SHIFT: pop the front word from the buffer, and push it onto the stack.
- REDUCE-L/R-X: pop the top two constituents off the stack, combine them into a new constituent with label X, and push the new constituent onto the stack.
- UNARY-X: pop the top constituent off the stack, raise it to a new constituent with label X, and push the new constituent onto the stack.

- FINISH: pop the root node off the stack and ends parsing

The SR parser is considered much faster than other existing parsers based on the experiments for parsing sentences in the Penn Treebank. Not only faster, but it is also considered more accurate than other parsers outside the RNN parsers. The Trance parser [4], a neural Shift-Reduce parser, was reported to achieve an F1-score of 90.68% on the Penn Treebank, and the Shift-Reduce parser of the Stanford Parser was reported to achieve the best F1-score of 88.55% for the Penn Treebank on their website.

III. PROPOSED METHOD

In this section, we present our proposed method in converting UI-CTB to the PTB format. The conversion consists of a data cleansing as a preprocessing task and the main conversion task composed of two subtasks: adjustment of the bracketing format and modification of the POS tags. The main conversion task is implemented in a program we call **PTB Converter**, while we named the resulting treebank after conversion as **Kethu**⁴.

A. Data Cleansing

Prior to converting the UI-CTB to the PTB format, sentences in UI-CTB require somewhat extensive corrections. This is achieved in the data cleansing task consisting of (1) splitting/merging sentences, (2) correcting typographical errors, (3) correcting the word segmentation, (4) correcting lost punctuation, (5) correcting POS tagging, and (6) correcting the annotation for reported speeches.

First, splitting and merging sentences are necessary since the current UI-CTB contains data where more than one sentences are associated with a tree, while some trees are associated with incomplete sentences. Splitting is performed for the former and merging for the latter.

Second, we correct typographical errors by (i) deleting excessive words; (ii) correcting the letter case so that uppercase and lowercase letters are used appropriately, e.g., when writing proper names; and (iii) adding missing characters.

Third, we correct the word segmentation. This concerns several things. One, reduplicated word such as *anak-anak* (kids), should not be split into multiple tokens. Two, abbreviations like *No.*, should not be split into multiple tokens. Three, punctuation that is part of a name, such as the dash character in *APBN-P*, should not cause the word to be split. Four, bounded morpheme part of a word, like *non-*, *anti-*, *pasca-*, or *antar-*, should not cause the word to be split. Finally, we perform special handling for words ended by the clitic of *-nya* as explained in [18]. In *mencintainya* (*to love him/her/it*), *-nya* should be separated from its main word, while in *terbentuknya* (the formation (of)), *-nya* should not be separated.

The fourth step is inserting lost punctuation. For example, in a sentence like "... *Crude Palm Oil CPO...*", CPO is the abbreviation of the previous words and should be wrapped with a pair of parentheses. This sentence

⁴Kethu is the name of a forest in Wonogiri, Central Java, Indonesia

should be corrected to "... *Crude Palm Oil (CPO)...*". Unfortunately, the occurrence of this kind of error is abundant in UI-CTB. We revised more than 200 words that lose their parenthesis. We investigated further and found out these mistakes originated from the source of the treebank [12].

The fifth step is correcting POS tagging errors. This includes (1) revising incorrect POS tagging, (2) changing POS tag due to merging/splitting tokens, and (3) revising POS tags due to ambiguous words with more than one possible POS tags. Some words whose POS tag were changed are *ingin* (want), *perlu* (need), and *mulai* (begin), which we converted to VB (verb).

The last step is correcting the reported speech annotation. In UI-CTB, the pair of double quotes were placed outside the syntactic tag of S-TPC-1 that contains the reported speech, while in PTB format, those double quotes should be put in the S-TPC-1 fragment.

B. Main conversion task: PTB Converter

The PTB Converter performs the conversion via two steps: (1) adjusting the bracketing format and (2) changing the POS tags.

1) *Adjusting the bracketing format:* In this step, we change the bracketing format so that it conforms to the PTB bracketing format. Here, we also split the compound words into tokens. There are 442 words represented as compound words in UI-CTB. Since we only found examples for compound noun and verb in PTB, for other types of compound in Indonesian, we propose an annotation format exemplified in Table III, which also shows examples of our proposed mapping from UI-CTB to PTB bracketing format.

Table III
MODIFICATION IN THE BRACKETING FORMAT.

No	UI-CTB	Kethu
1	(NN (laporan))	(NN laporan)
2	(NN (tanggung jawab))	(NP (NN tanggung) (NN jawab))
3	(VB (bertanggung jawab))	(VP (VB bertanggung) (PRT (RP jawab)))
4	(JJ (luar biasa))	(ADJP (PRT (RP luar)) (JJ biasa))
5	(RB (sama sekali))	(ADVP (PRT (RP sama)) (RB sekali))
6	(CD (salah satu))	(NP (PRT (RP salah)) (CD satu))
7	(CD (salah seorang))	(NP (PRT (RP salah)) (DT seorang))
8	(CC (Lagi pula))	(ADVP (RB Lagi) (PRT (RP pula)))
9	(IN (di mana))	(ADVP (WRB di) (PRT (RP mana)))
10	(NP-SBJ (*))	(NP-SBJ (-NONE- *))
11	(-LRB- (Z (&brl;)))	(-LRB- -LRB-)
12	(-RRB- (Z (&brrr;)))	(-RRB- -RRB-)

Example 1 shows the modification to a single token, examples 2–9 are for compound words modification, and examples 10–12 are for other bracketing issues. In example

2 and 3, we adopted PTB's approach in handling compounds of noun and verb. Examples 4–9 are our proposed annotation for compound adjective and others.

Example 10 is for the case where the segment (*) is considered a syntax error in PTB since it has no POS tag. In PTB, these tokens are given the POS tag -NONE-, so we change every occurrence of (*) to (-NONE- *).

Example 11 and 12 are for the tokens -LRB- (left parenthesis) and -RRB- (right parenthesis). In UI-CTB, both tokens are treated as a syntactic tag, while in PTB they are a POS tag. Moreover, these labels only occur 3 times in UI-CTB as syntactic tags, while in Kethu we have 255 pair of them, as the POS tags.

At the end of this step, we have converted the bracketing format of UI-CTB into PTB. Now, UI-CTB can be processed by any tool that expects treebank in PTB format. However, we want to make sure that our new treebank, Kethu, also complies with the POS tagset of PTB. So, we need to do the next step, mapping the POS tagset of UI-CTB to PTB.

2) *Mapping the POS Tagset from UI-CTB to PTB:* We analyze the differences between UI-CTB POS tagset [6] and PTB tagset [10]. Table IV shows our proposed mapping of specific tags in UI-CTB to the associated POS tags in PTB.

Table IV
THE MAPPING OF UI-CTB'S SPECIFIC TAGS TO PTB TAGS.

UI-CTB	Description	PTB
NEG	negation	RB
NND	classifier noun	NN
OD	ordinal number	JJ, CD
PR	demonstrative pronoun	DT
PRP	pronoun	PRP, PRP\$, DT
SC	subordinating conjunction	IN
VB	verb	VB, VBZ
WH	question word	WP, WP\$, WRB, WDT, IN
X	unknown	FW
Z	punctuation	-LRB-, -RRB-, comma (,), period (.), ;, “,”

For OD in UI-CTB, only those that describes a noun is converted to JJ (adjective) such as in "*anak kedua*" (*the second child*), in "*kedua anak*" (*both children*) *kedua* will be labeled as CD.

For PRP in UI-CTB, we converted the tokens of possessive personal pronoun to PRP\$. Some tokens with PRP but we considered the semantic are as the determiner were converted from PRP to DT (determiner), such as in (*NP (NN hal) (DT nya)*) (*the case*).

In UI-CTB there is only one type of verb. We decided to differentiate verb (VB) into VB and VBZ. VBZ will be used to label copula verbs that have special characteristic compare to other verbs in Indonesian. According to [7], copula verbs in Indonesian are *adalah* and *ialah*.

In UI-CTB, there is only one label for question word, WH. Since in PTB, there are four types of question words, we mapped WH in UI-CTB to 5 labels: WP, WP\$, WRB, WDT and IN. WP for *apa* (*what*) and *siapa* (*who*); WP\$

for *siapa* following noun; WRB for *di mana* (*where*), *kapan* (*when*), *ketika* (*when*) and *bagaimana* (*how*); WDT for *yang* (*which, that*) or *yang mana* (*which*); and finally IN, if those question words are used as the subordinating conjunction.

Furthermore, we also found that in UI-CTB, words like *beberapa* (*some*), *seorang* (*a*), *semua* (*all*) are labeled as CD (cardinal number). Since in PTB these words are tagged as a determiner (DT), we also revised them.

At the end of this step, the treebank conversion is done, and we obtain a new version of UI-CTB that conforms to the Penn Treebank format.

C. Statistics of the Treebanks

We present some statistics of both UI-CTB and the Kethu treebank to give pictures of how far we have changed the UI-CTB.

1) The comparison of the number of sentences and words: Table V shows the number of sentences, number of words and the average length of sentences in both treebanks. Although there is no change in the number of sentences, actually we did some splitting and merging on UI-CTB sentences. For the number of words, it is important to note that what we mean here by "word" conformed to how each treebank represents the word. In UI-CTB, a compound word is considered as a word, while in Kethu, a word is a single token. So, it's not surprising that the number of words in Kethu is higher than UI-CTB. Not to mention we also added more than 200 pairs of parentheses. In the end, the average sentence length of Kethu treebank is a bit higher than the UI-CTB.

Table V
COMPARISON OF THE NUMBER OF SENTENCES AND WORDS.

Description	UI-CTB	Kethu
Number of sentences	1,030	1,030
Number of words	27,115	28,117
Avg sentence length	26.33	27.3

2) The distribution of POS tags: Table VI shows the distribution of POS tag in UI-CTB and Kethu treebank. The column with "-" means that the POS tag was not the member of the corresponding treebank POS tagset.

The total number of the first seven POS tags that represent the punctuation in Kethu is 3426, with a margin of 774 with Z in UI-CTB. This amount difference is contributed by the additional 255 pairs of brackets and more than 90 pairs of double quotes.

The number of CD in Kethu was decreased around 500 while number of DT was increased as the result of converting words like *beberapa* (*some*), *seorang* (*a*), *semua* (*all*) from CD to DT. The number of IN in Kethu was increased significantly since almost all words with SC tag in UI-CTB was converted to IN.

IV. EXPERIMENTS AND RESULTS

In this section, we present the experiments and results.

Table VI
THE POS TAG DISTRIBUTION.

POS	UI-CTB	Kethu	POS	UI-CTB	Kethu
-LRB-	-	255	NNP	3868	4070
-RRB-	-	255	OD	100	-
,	-	1430	PR	508	-
:	-	89	PRP	704	478
"	-	1012	PRP\$	-	145
"	-	242	RB	564	732
"	-	243	RP	18	70
CC	776	802	SC	1156	-
CD	2372	1854	SYM	331	286
DT	20	594	UH	4	5
FW	423	390	VB	2885	2874
IN	2264	3349	VBZ	-	37
JJ	1073	1068	WH	8	-
MD	535	492	WP	-	13
NEG	127	-	WRB	-	11
NN	6455	7321	X	24	-
NND	148	-	Z	2752	-

A. Experiments

To evaluate the quality of the resulting treebank (Kethu), we use the Shift Reduce parser of the Stanford CoreNLP to build the Indonesian constituency parser model. The evaluation method used is 10-fold cross-validation. We divided Kethu treebank into 10 parts, each consists of 103 sentences. The training dataset consists of 8 parts of 824 sentences, the development and test dataset each consists of 103 sentences.

B. Results

After conducting the 10-fold cross validation method, we got the average of Labeled Precision (LP) of 70.37%, Labeled Recall (LR) of 71.46% and F1-score of 70.90%.

Table VII shows a summary of the F1-score of 4 studies on Shift-Reduce parsing. In general, we can see that Trance parser outperforms the Stanford parser, both for English and Indonesian. If we compare the accuracy of English and Indonesian parser, we can see the Indonesian parser trained using a variant of UI-CTB has a lower accuracy than the English parser trained using PTB.

Table VII
THE COMPARISON OF SOME WORKS ON SHIFT-REDUCE PARSING

Parser	Treebank	F1 (%)
Trance parser	PTB	90.68
Trance parser	UI-CTB revised by [8]	74.91
Stanford parser	PTB	88.55
Stanford parser	Kethu	70.90

Our result is lower than [8] that also uses a revised UI-CTB. However, since we only conducted a simple experiment to Kethu treebank without doing tree normalization or binarization like Filino & Purwarianti [8] did, we cannot compare the quality of Kethu treebank to the revised UI-CTB they use.

We suggest three reasons why Indonesian parser has lower accuracy: 1) the size of the treebank that is very small compared to PTB; 2) the quality of the treebank, and 3) special characteristics of the Indonesian grammar.

To have a better Indonesian parser, building a new treebank with a bigger size and also in better quality in word segmentation and POS tagging is needed. We also need to involve Indonesian linguists to investigate whether some special treatments are needed to compute Indonesian grammar.

V. CONCLUSION AND FUTURE WORK

We have proposed a method to convert an Indonesian constituency treebank to the Penn Treebank format. This needs to be done so that we can utilize the various tools provided for constituency treebanks such as the Stanford Parser. The conversion consists of two main tasks: data cleansing and developing the component named the PTB Converter.

We evaluated the quality of the new treebank by building the Indonesian constituency parser model using Shift-Reduced parser provided by the Stanford CoreNLP. The experiments show that our parser model has an F1-score of 70.90%.

The main contribution of our work is to provide an Indonesian constituency treebank that conforms to PTB format so that it can be used as the input of any tools that expected a treebank in PTB format.

In this work, we only revised the word segmentation, the POS tagging of UI-CTB, and correcting the annotation of reported speech. We have not reviewed other syntactic tagging of the treebank. In future work, we will conduct analysis to its syntactic annotation in order to improve the quality of the treebank.

We also recommend building a larger treebank with a bigger size and better quality so that we can produce a better Indonesian parser. The format of this treebank should also conform to the international standard format so that we can utilize tools that are already available.

ACKNOWLEDGMENT

This research was supported by the research grant of "Hibah PITTA B 2019" from Universitas Indonesia, No. NKB-0508/UN2.R3.1/HKP.05.00/2019.

REFERENCES

- [1] D. Jurafsky and J. H. Martin, *Speech and Language Processing, 2nd Edition: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Prentice Hall, 2008.
- [2] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz, "Building a large annotated corpus of English: The Penn Treebank," *Computational Linguistics*, vol. 19, no. 2, pp. 313–330, 1993.
- [3] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP Natural Language Processing Toolkit," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2015, pp. 55–60.
- [4] T. Watanabe and E. Sumita, "Transition-based Neural Constituent Parsing," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015, pp. 1169–1179. [Online]. Available: <http://www.aclweb.org/anthology/P15-1113>
- [5] A. Van Cranenburgh, R. Scha, and R. Bod, "Data-Oriented Parsing with Discontinuous Constituents and Function Tags," *Journal of Language Modelling*, 2016.
- [6] A. Dinakaramani, F. Rasheed, A. Luthfi, and R. Manurung, "Designing an Indonesian part of speech tagset and manually tagged Indonesian corpus," in *Proceedings of the International Conference on Asian Language Processing 2014, IALP 2014*, 2014, pp. 66–69.
- [7] J. N. Sneddon, A. Adelaar, D. N. Djener, and M. C. Ewing, *Indonesian Reference Grammar.* A&U Academic, 2010.
- [8] M. Filino and A. Purwarianti, "Indonesian shift-reduce constituent parser," in *Proceedings of 2016 International Conference on Data and Software Engineering, ICODSE 2016*, 2016.
- [9] K. M. Kurniawan, "Exploring Recurrent Neural Network Grammars for Parsing Low-Resource Languages," Master Thesis, University of Edinburgh, 2017.
- [10] A. Taylor, M. P. Marcus, and B. Santorini, "The Penn Treebank: An Overview," in *Text, Speech and Language Technology*, 2003, pp. 5–22.
- [11] S. Larasati, "IDENTIC Corpus: Morphologically Enriched Indonesian-English Parallel Corpus," in *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, 2012, pp. 902–906.
- [12] BPPT, "Research Report on Corpus Design and Collection and Cleaning Tools English to Bahasa Indonesia," Tech. Rep., 2010.
- [13] C. Dyer, A. Kuncoro, M. Ballesteros, and N. A. Smith, "Recurrent Neural Network Grammars," in *NAACL 2016*, 2016. [Online]. Available: <http://arxiv.org/abs/1602.07776>
- [14] D. Klein and C. D. Manning, "Accurate unlexicalized parsing," in *ACL '03 Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1* pp 423-430, Sapporo, Japan, 2003.
- [15] R. Socher, C.-Y. Lin, A. Ng, and C. Manning, "Parsing natural scenes and natural language with recursive neural networks," in *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, 2011.
- [16] M. Zhu, Y. Zhang, W. Chen, M. Zhang, and J. Zhu, "Fast and Accurate Shift-Reduce Constituent Parsing," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2013.
- [17] K. Sagae and A. Lavie, "A classifier-based parser with linear run-time complexity," in *Proceedings of the Ninth International Workshop on Parsing Technology*, Vancouver, British Columbia, 2005.
- [18] I. Alfina, A. Dinakaramani, M. I. Fanany, and H. Suhartanto, "A Gold Standard Dependency Treebank for Indonesian," in *Proceeding of the 33rd Pacific Asia Conference on Language, Information and Computation (PACLIC 33)*, Hakodate, Japan, 2019.

How to Answer Comparison Questions

Hongxuan Tang, Yu Hong, Xin Chen, Kaili Wu, Min Zhang
*School of Computer Science and Technology, Soochow University
 Suzhou, China*

{*hxtang,xchen,klwu*}@*stu.suda.edu.cn*,*tianxianer@gmail.com*,*minzhang@suda.edu.cn*

Abstract—“Which city has the larger population, Tokyo or New York?”. To answer the question, in general, we necessarily obtain the prior knowledge about the populations of both cities, and accordingly determine the answer by numeric comparison. Using Machine Reading Comprehension (MRC) to answer such a question has become a popular research topic, which is referred to as a task of Comparison Question Answering (CQA). In this paper, we propose a novel neural CQA model which is trained to answer comparison question. The model is designed as a sophisticated neural network which performs inference in a step-by-step pipeline, including the steps of attentive entity detection (e.g., “city”), alignment of comparable attributes (e.g., “population” of the target “cities”), contrast calculation (larger or smaller), as well as binary classification of positive and negative answers. The experimentation on HotpotQA illustrates that the proposed method achieves an average F1 score of 63.09%, outperforming the baseline with about 10% F1 scores. In addition, it performs better than a series of competitive models, including DecompRC, BERT.

Keywords-Machine Reading Comprehension; Comparison Question Answering; Calculation

I. INTRODUCTION

Machine Reading Comprehension (MRC) aims to extract answers from textual data. It is an important task in the field of natural language processing (NLP). After the release of the SQuAD dataset [1], MRC has spurred tremendous interest in the NLP community. CQA is an instance of MRC, questions of which need comparison, counting and arithmetic. It is quite challenging for current MRC models. In this paper, we evaluate our CQA model on comparison type questions in HotpotQA [2] (HotpotCQA). Table 1 shows an example from HotpotCQA.

Table I
 AN EXAMPLE FROM HOTPOTCQA.

Question: Who has more well known films, Quincy Perkins or Bill Forsyth?
Document α : Quincy Perkins (born July 16, 1980 in Key West, Florida) is an American director most famous for directing, producing and writing the narrative fiction short film “Swingers Anonymous” which debuted at the (Cannes Film Festival) in 2015.
Document β : William David “Bill” Forsyth (born 29 July 1946) is a Scottish film director and writer known for his films “Gregory’s Girl” (1981), “Local Hero” (1983), and “Comfort and Joy” (1984).
Document γ :
Answer: William David “Bill” Forsyth

As the question shown in Table I, “Who has more well known films, Quincy Perkins or Bill Forsyth?”. Two

entities, Quincy Perkins and Bill Forsyth, can be focused easily. The golden answer is probably one of them. But it is still hard to determine which is the golden answer. Commonly, we determine the answer as follows: Firstly, figuring out the larger value or smaller value is needed by analyzing the question. We call the questions which are looking for larger values positive questions, otherwise negative questions. Then, we should know the numbers of well known films directed by Quincy Perkins or Bill Forsyth respectively. Actually, we don’t need the exact numbers of well-known films directed by each of them. Due to all questions in HotpotCQA are comparing between two entities and a document describe only one entity, thus we only need to know if the corresponding value of entity α (document α contains) is greater than the corresponding value of entity β (document β contains). This is a document polarity problem. If document α contains the larger value, the document polarity is positive, otherwise negative document polarity. For the example in Table 1, the question polarity is positive because of the word “more” in the question, and the document polarity is negative because document β contains the larger value (3 well-known films in document β compared to 1 well-known film in document α). Question polarity and document polarity constitute a semantic-level XOR problem, and it is challenging for current MRC models.

In this paper, we propose a CQA model with question polarity module and document polarity module. The two modules can determine question polarity and document polarity respectively. At last, We get the answer by question polarity XOR document polarity. The way we determine the golden answer is shown in Fig.1. Besides, we jointly train the two modules. Our main contributions can be summarized as follows:

- We discovered a semantic-level XOR problem in the comparison question.
- We propose an objective function to retrieve related documents and a neural CQA model to answer comparison questions. We conduct experiments on HotpotCQA dataset and outperforms state-of-the-art methods.

II. RELATED WORK

We review the related work about MRC in Section II-A. Then we introduce the characteristics of CQA in Section II-B.

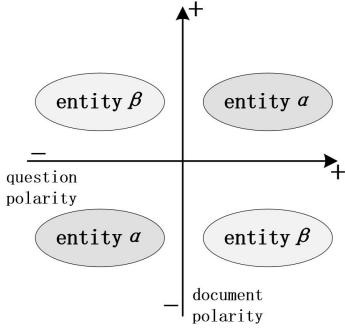


Figure 1. Getting the answer by question polarity and document polarity

A. End-to-end MRC model

Recently, MRC makes significant progress in span-based question task. Common approaches [3], [4], [5] use Recurrent Neural Networks (RNN) [6] to encode textual data, and use sophisticated interactions to capture the high-dimensional information, and use Pointer Network [7] to predict the start and end position of the answer. Pointer Network has been the standard output layer. However, different kinds of attention mechanism are widely used in MRC models to capture the high-dimensional information. Bidirectional attention mechanism [8] is used to capture the interaction among the document words conditioned on question word, self attention mechanism [9] is used to align the sentence representation against itself and multi-head attention [10] is used to capture semantic information hierarchically.

B. Why CQA

In recent years, several datasets and evaluation metrics on MRC have been built [11], [12], [1], [13]. On SQuAD 1.0, SQuAD 2.0 [1] and CoQA [14], neural network models can achieve either super-human or the ceiling performance, but it does not mean machine can do reading comprehension better than human. Machine still makes some simple mistakes which humans hardly make [15], it indicates that sometimes machine does not understand natural language which is one of the aims of MRC. So we should use multi-hop datasets to evaluate the comprehension ability of current neural network models. The comparison question is a kind of multi-hot question because we need to find knowledge about comparison objects respectively. While the knowledge is hidden in different documents, it means more than one document is needed to support the answer. Therefore study on CQA can make contributions to MRC and natural language understanding (NLU).

III. MODEL

In CQA, the model is given a set of documents and a related question. The model aims to answer the question supported by more than one documents. We call the supporting documents as “golden documents”. There are two golden documents in HotpotCQA which describe two comparison objects respectively. In this paper, we mark two golden documents as document α and document β .

Our model solves CQA in two steps: 1) Comparison object detection. 2) Answer prediction, comparison and selection. The first step is to retrieve golden documents from the given documents set by an objective function (Section III-A). Then, in the second step, we use a CQA neural network model to extract the candidate answers from two golden documents respectively. Then we calculate the question polarity and document polarity. Lastly, we select one of the candidate answers as the final answer.

The CQA model contains four modules: 1) Answer prediction (Section III-B) extract answers from two golden documents respectively. 2) Question polarity (Section III-C) determine the polarity of the related question, in order to decompose the semantic-level XOR problem. 3) Documents polarity (Section III-D) calculate the degree of document polarity and determines the polarity of documents. 4) Answer selection (Section III-E) considers the results of the previous two modules, and selects the final answer based on conditional probability. The overview of our approach is shown in Fig. 2.

A. Document Selection

In order to retrieve golden documents, we evaluate the documents by related questions, and we use the recall rate to rank the documents. We treat the documents’ titles and questions as two bags-of-words, then compute the number of overlaps between titles and questions:

$$\begin{aligned} score_i &= f(question, title_{doc_i}) \\ &= RECALL/TWORDS \end{aligned} \quad (1)$$

where $question$ denotes the bags-of-word of given question, $title_{doc_i}$ denotes the bags-of-word of i-th document’s title, $RECALL$ denotes the number of words both in $question$ and $title_{doc_i}$, $TWORDS$ denotes the number of $title_{doc_i}$.

According to the score of each document, we consider two documents with the highest score as golden documents and the two documents describe two comparison objects mentioned in the question. We mark the two golden documents as document α and document β .

B. Answer Prediction

1) Encoding Layer: We get word-level distributed word representations by mapping each word in question Q and golden documents $\{D_\alpha, D_\beta\}$ to a high-dimensional vector space. Then, we use Convolutional Neural Networks (CNN) [16] to get character-level distributed word representations. Lastly, we concatenate word-level embedding, character-level embedding and three-input EM (TEM) feature as the final embedding.

Let $D = \{x_1, x_2, \dots, x_d\}$ and $Q = \{q_1, q_2, \dots, q_q\}$ represent the words in the document and question. In order to get encoding representations of golden documents and questions, we use a bidirectional Gated Recurrent Unit (BiGRU) [6], [17] with h hidden size.

$$H_Q = BiGRU(Q) \quad (2)$$

$$H_{D_{\alpha,\beta}} = BiGRU(D_{\alpha,\beta}) \quad (3)$$

where $H_{D_\alpha}, H_{D_\beta} \in \mathbb{R}^{d*2h}$ and $H_Q \in \mathbb{R}^{q*2h}$.

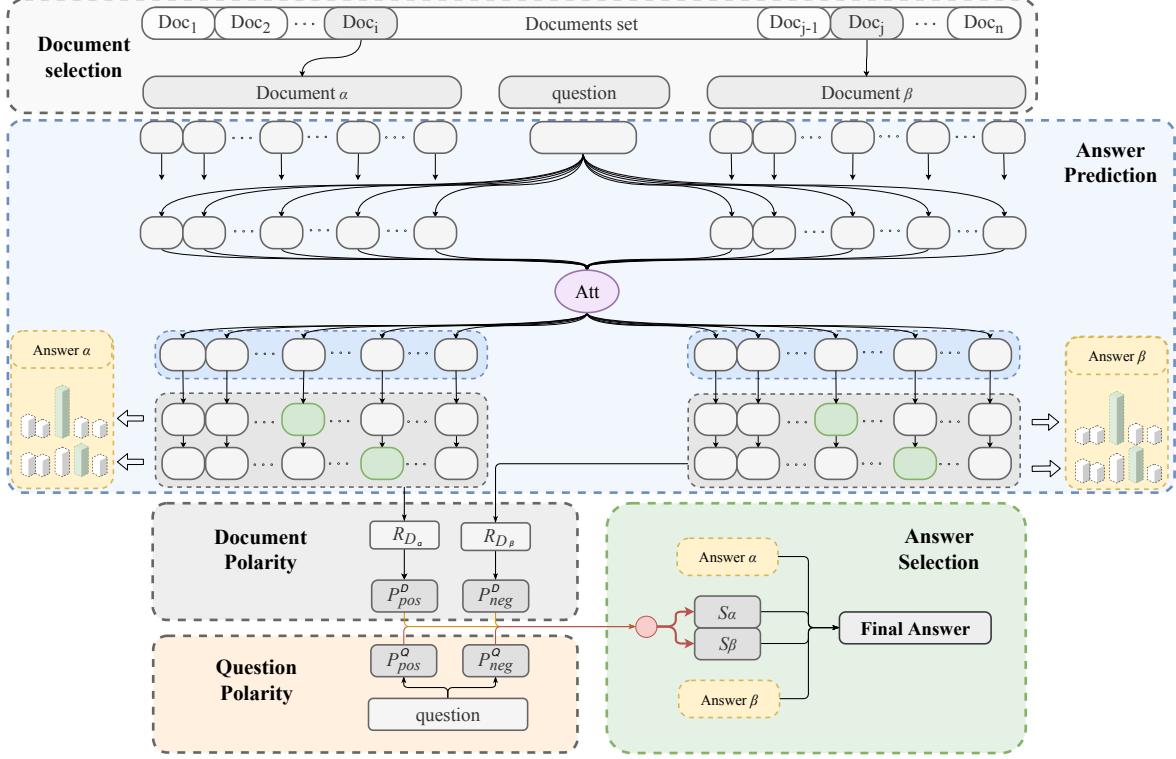


Figure 2. Overview of our approach

2) *TEM Feature*: EM feature [18] can significantly improve the performance of MRC models. Now we propose a new version EM feature for the three-input model. Different from the traditional EM feature, we use five numbers to mark different words instead of a simple binary number. Each number maps to a high-dimensional vector space. Fig. 3 shows the rule we mark each word and the difference between two versions.

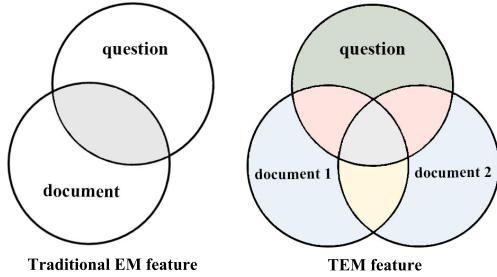


Figure 3. Traditional EM and TEM features. The three circles in TEM figure represent the three inputs of our CQA model. The intersection of the circles represents the same words in inputs. Different colors denote different numbers which be mapped to by words.

3) *Attention Layer*: We use three kinds of attention mechanisms: question-document, document-document and self attention. After we get the encoding representations H_{D_α} , H_{D_β} and H_Q , we use a bi-directional question-document attention mechanism [8], [2] to update the H_D , in order to capture semantic information between the question and the documents. Then we capture semantic information between two documents. Lastly, self attention [9] is used to align document presentation. We get

the encoding representations of two documents G_α and G_β as follows:

$$S_{D_{\alpha,\beta}}^1 = W_1 * Att(H_{D_{\alpha,\beta}}, H_Q) + b_1 \quad (4)$$

$$S_{D_{\alpha,\beta}}^2 = W_2 * Att(S_{D_{\alpha,\beta}}^1, S_{D_{\beta,\alpha}}^1) + b_2 \quad (5)$$

$$G_{\alpha,\beta} = W_3 * Att(S_{D_{\alpha,\beta}}^2, S_{D_{\alpha,\beta}}^2) + b_3 \quad (6)$$

where $W_1, W_2, W_3 \in \mathbb{R}^{8h*8h}$ are trainable weight vectors. $S_{D_{\alpha,\beta}}^1 \in \mathbb{R}^{d*8h}$ is the representation of golden documents which obtains the question information. $S_{D_{\alpha,\beta}}^2 \in \mathbb{R}^{d*8h}$ combines the information in D_α and D_β . $G_{\alpha,\beta} \in \mathbb{R}^{d*8h}$ aligns the document representation.

4) *Decoding Layer*: In order to extract the answer span from the documents, we use BiGRU to get the decoding representation of start position $O^{start} \in \mathbb{R}^{d*2h}$. Different from start position, we concatenate the decoding representation of start position O^{start} and document representation G to get the decoding representation of end position $O^{end} \in \mathbb{R}^{d*3h}$. At last, we use Pointer Network [7] to get the probability of each word to be the start $p^1 \in \mathbb{R}^d$ and the end $p^2 \in \mathbb{R}^d$:

$$O_{\alpha,\beta}^{start} = BiGRU(G_{\alpha,\beta}) \quad (7)$$

$$O_{\alpha,\beta}^{end} = BiGRU([O_{\alpha,\beta}^{start} : G_{\alpha,\beta}]) \quad (8)$$

$$p_{\alpha,\beta}^{1,2} = softmax(\max(O_{\alpha,\beta}^{start,end})) \quad (9)$$

where subscripts α and β are used to distinguish between two golden documents.

We get Ans_α from document α based on p_α^1 and p_α^2 . Similarly, we get Ans_β from document β based on p_β^1 and p_β^2 .

C. Question Polarity

In order to determine the question polarity, we import thirteen comparative words to label the questions in training set. The comparative words include 7 more-like words (more, higher, taller ect.) and 6 less-like words (less, lower ect.). If the question contains any one of more-like words, the polarity of the question is positive, otherwise negative polarity. We label the positive question “pos” and negative question “neg”. We calculate the probabilities of positive polarity and negative polarity as follow:

$$P_{pos}^Q = \text{sigmoid}(W_Q * H_Q + b_Q) \quad (10)$$

$$P_{neg}^Q = 1 - P_{pos}^Q \quad (11)$$

where P_{pos}^Q represents the probability of positive question polarity, P_{neg}^Q represents the probability of negative question polarity.

D. Documents Polarity

In question polarity module, we get the question polarity. Thus, in documents polarity module we should only make the module able to give higher scores for documents with larger values. We concatenate the two representations, O^{start} and O^{end} , of two documents in decoding layer respectively. Then, we get two new documents representations R_D^S and R_D^S of each document. Lastly, we use a linear to calculate the score of two documents, and use a soft-max function to get the probability of positive document polarity and negative document polarity.

$$R_D^S = [O^{start} : O^{end}] \quad (12)$$

$$\text{score}_\alpha = W_D * R_{D_\alpha}^S + b_D \quad (13)$$

$$\text{score}_\beta = W_D * R_{D_\beta}^S + b_D \quad (14)$$

$$P_{pos}^D, P_{neg}^D = \text{softmax}(\text{score}_\alpha, \text{score}_\beta) \quad (15)$$

where $P_{pos}^D, P_{neg}^D \in \mathbb{R}^1$, score_α and score_β represent the degree of document polarity respectively. P_{pos}^D represent the probability of positive document polarity, P_{neg}^D represent the probability of negative document polarity.

E. Answer Selection

In this module, the result of question polarity and documents polarity will be used to calculate the scores of two candidate answers based on the probability theory. We mark the probability that answer is from the document α as P_α and the probability that answer is from the document β as P_β .

$$P_\alpha = P_{pos}^Q * P_{pos}^D + P_{neg}^Q * P_{neg}^D \quad (16)$$

$$P_\beta = P_{pos}^Q * P_{neg}^D + P_{neg}^Q * P_{pos}^D \quad (17)$$

where $P_{pos}^Q + P_{neg}^Q = 1$, $P_{pos}^D + P_{neg}^D = 1$, $P_\alpha + P_\beta = 1$.

Generally, we select Ans_α in two conditions: First, we get a positive question and a positive document, such as example 1 shown in Table II. Second, we get a negative question and a negative document, such as example 2 shown in Table II.

Table II
EXAMPLES FOR ANSWER SELECTION.

Example 1
Question: Which city has larger population, CITY α or CITY β ?
Document α : CITY α has a population of 2 million.
Document β : CITY β has a population of 1 million.
Example 2
Question: Which city has less population, CITY α or CITY β ?
Document α : CITY α has a population of 1 million.
Document β : CITY β has a population of 2 million.
Answer : CITY α

F. Joint Training

We jointly train answer prediction module, question polarity module, document polarity module and answer selection module, the loss of $question_i$ ($loss_i$) contains the loss of answer prediction L_{ans} , the loss of question polarity L_{ques} , the loss of document polarity L_{doc} and the loss of answer selection L_{sele} . Specially, we multiply an indicator function $I(i)$ before L_{ans} as a switch, L_{ans} will be set to zero if answer selection module doesn't make a correct judgment. Besides, to avoid getting a smaller loss when the wrong answer is selected, we multiply a hyperparameter multiplier λ before L_{sele} :

$$loss_i = I(i)L_{ans} + L_{ques} + L_{doc} + \lambda * L_{sele} \quad (18)$$

$$\text{Among } I(i) = \begin{cases} 1, & \text{predict} = \text{label} \\ 0, & \text{otherwise} \end{cases} \quad (19)$$

where L_{ans} , L_{ques} , L_{doc} and L_{sele} are cross entropy loss function.

IV. EXPERIMENTS

We evaluate the model on HotpotCQA which is a part of HotpotQA [2]. HotpotCQA contains 18,943 question-answer pairs on 189,430 documents. The dataset not only provides the answer span but also provides sentence-level supporting facts required for reasoning. The solution proposed in this paper can provide document-level supporting facts instead of sentence-level.

A. Implementation Details

We develop our CQA model using Pytorch deep learning framework. We use Glove 300-dimensional word embeddings [19] and 8-dimensional char embeddings. Our encoder hidden size is set to 50 for word-level embedding, and 100 for character-level embedding. We use Adam [20] as our optimizer, and set learning rate to 0.00007. Multiplier λ which is used to keep balance between document selection and answer prediction is set to 1.0.

B. Experimental Results

Following SQuAD [1] and HotpotQA [2], we use EM and F1 scores to evaluate our model. We resplit the dataset evenly and evaluate our model by four-fold cross-validation because the test set of this part is not released. Each fold contains about 16,594 training data, 1,181 dev data, and 1,181 test data. We use the golden documents in training set to test our document selection module, the result shows that over 90% scores of golden

Table III
FOUR-FOLD CROSS-VALIDATION RESULTS ON HOTPOTCQA.

Model	Document Selection									
	1st		2nd		3rd		4th		Avg	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
ours	88.63	94.14	88.80	94.23	88.70	94.01	88.91	94.28	88.76	94.17
Final Results										
Model	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
baseline-cp	47.20	53.68	44.65	51.74	47.35	53.79	42.25	48.95	45.36	52.04
baseline-total	48.66	55.38	43.63	51.01	46.49	52.66	43.94	50.02	45.68	52.27
ours	56.24	63.16	54.84	62.56	57.36	64.76	55.80	61.87	56.06	63.09

Table IV
EXPERIMENTAL RESULTS ON THE DEV SET OF HOTPOTCQA.

setting	EM	F1
ours	58.43	66.22
DecompRC	-	62.78
BERT	-	57.81
baseline	48.55	55.05

documents are ranged from 0.8 to 1.0. The results show that the objective function we proposed can retrieve golden documents effectively. The score distribution is shown in Fig.4.

We train the baseline model in two ways, using total HotpotQA datasets and using HotpotCQA only. Four-fold cross-validation results show that our approach reaches an average performance of 63.09% F1 and 56.06% EM, including over 10% F1 score gains. The four results in document selection are similar, it shows the proposed objective function is reliable. The result of four-fold cross-validation is shown in Table III.

Table IV compares the results of our CQA model with other proposed models on the HotpotCQA development set. The results show that our CQA model outperforms BERT [3] 8.41% F1 score and outperforms DecompRC [21] 3.44% F1 score.

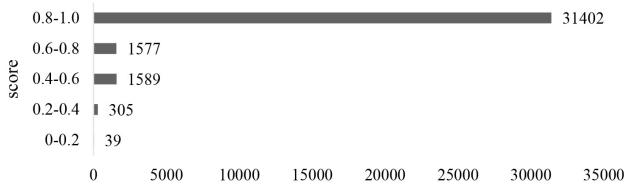


Figure 4. Score distribution of the golden documents

C. Interpretability Experiments

In order to prove the interpretability of our model, we test the model by manual data. We modify only one of the words in the question to structure an opposite question. For example, original question “Who has more well known films, Quincy Perkins or Bill Forsyth?” becomes “Who has less well known films, Quincy Perkins or Bill Forsyth?”. We use our model to extract answers from the same documents set relating these two opposite questions. And the result shows in Table V.

For further research, the intermediate parameters of each module are shown in Table VI. Two opposite samples

Table V
RESULT OF INTERPRETABILITY EXPERIMENT.

Document α : <i>Quincy Perkins (born July 16, 1980 in Key West, Florida) is an American director most famous for directing, producing and writing the narrative fiction short film “Swingers Anonymous” which debuted at the [Cannes Film Festival] in 2015.</i>
Document β : <i>William David “Bill” Forsyth (born 29 July 1946) is a Scottish film director and writer known for his films “Gregory’s Girl” (1981), “Local Hero” (1983), and “Comfort and Joy” (1984).</i>
Original Question: Who has more well known films, Quincy Perkins or Bill Forsyth?
Answer: William David “Bill” Forsyth
Opposite Question: Who has less well known films, Quincy Perkins or Bill Forsyth?
Answer: Quincy Perkins

Table VI
THE INTERMEDIATE PARAMETER OF EACH MODULE.

parameters	original	opposite
P_{pos}^Q	99.41%	1.29%
P_{neg}^Q	0.59%	98.71%
P_{pos}^D	15.46%	18.32%
P_{neg}^D	84.54%	81.68%

got similar document polarity score, but got different question polarity score. According to the Eq. 16 and Eq. 17 given in section III-E, we get $P_\alpha = 0.16\%$ and $P_\beta = 0.84\%$ in original sample, but $P_\alpha = 0.81\%$ and $P_\beta = 0.19\%$ in opposite sample. Our CQA model answers both two opposite questions correctly, and the result shows that question polarity module distinguishes the difference between two questions and judges the question polarity correctly.

D. Ablation Study

To show the effects of our model, we conduct an ablation study on the development set of HotpotCQA. The results of ablation study are shown in Table VII. We find that self-attention contributes little to the performance. One possible reason is that the length of the textual data is short enough after document selection. Self-attention makes more contributions when the textual data is long enough. In contrast, the TEM feature contributes remarkably to the performance. To make a comparison, we add EM feature to baseline model, and the result shows that TEM feature has obvious advantages over EM feature. Because of the characteristic of CQA, TEM can highlight the relation between questions and documents, and the

Table VII
RESULT OF ABLATION STUDY.

setting	EM	F1
ours	58.43	66.22
<i>baseline + EM</i>	49.42	56.04
<i>baseline</i>	48.55	55.05
<i>-TEM feature</i>	53.19	60.19
<i>-self Attention</i>	58.17	65.19
<i>-DocAttention</i>	58.10	64.82
<i>-L_{ques}</i>	54.80	61.52

relation between each document. Besides, we set L_{ques} to zero to simulate the situation without distinguishing question polarity. And the result shows that question polarity module provides about 4.7% F1 scores.

V. CONCLUSION

In this paper, we discovered a semantic-level XOR in the comparison question and propose a method to decompose this XOR problem. We also propose an objective function to retrieve documents and a CQA model to answer comparison questions with considering question polarity and document polarity. Our approach achieve the state-of-the-art performance on HotpotCQA. Moreover, our method of decomposition semantic-level XOR problem to improve the answer location ability is effective.

VI. ACKNOWLEDGMENTS

This research work is supported by National Natural Science Foundation of China (Grants No.61672368, No.61703293, No.2017YFB1002104). The authors would like to thank the anonymous reviewers for their insightful comments and suggestions. Yu Hong, Professor in Soochow University, is the corresponding author of the paper, whose email address is tianxianer@gmail.com.

REFERENCES

- [1] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “Squad: 100,000+ questions for machine comprehension of text,” *arXiv preprint arXiv:1606.05250*, 2016.
- [2] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning, “Hotpotqa: A dataset for diverse, explainable multi-hop question answering,” *arXiv preprint arXiv:1809.09600*, 2018.
- [3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [4] M. Hu, Y. Peng, Z. Huang, X. Qiu, F. Wei, and M. Zhou, “Reinforced mnemonic reader for machine reading comprehension,” *arXiv preprint arXiv:1705.02798*, 2017.
- [5] H. Y. Huang, E. Choi, and W. T. Yih, “Flowqa: Grasping flow in history for conversational machine comprehension,” 2018.
- [6] T. Mikolov, M. Karafiat, L. Burget, J. Černocky, and S. Khudanpur, “Recurrent neural network based language model,” in *Eleventh annual conference of the international speech communication association*, 2010.
- [7] O. Vinyals, M. Fortunato, and N. Jaitly, “Pointer networks,” in *Advances in Neural Information Processing Systems*, 2015, pp. 2692–2700.
- [8] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, “Bidirectional attention flow for machine comprehension,” *arXiv preprint arXiv:1611.01603*, 2016.
- [9] S. Wang and J. Jiang, “Machine comprehension using match-lstm and answer pointer,” *arXiv preprint arXiv:1608.07905*, 2016.
- [10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.
- [11] K. M. Hermann, T. Kočiský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, “Teaching machines to read and comprehend,” in *Advances in neural information processing systems*, 2015, pp. 1693–1701.
- [12] F. Hill, A. Bordes, S. Chopra, and J. Weston, “The goldilocks principle: Reading children’s books with explicit memory representations,” *Computer Science*, 2015.
- [13] M. Richardson, C. J. Burges, and E. Renshaw, “Mctest: A challenge dataset for the open-domain machine comprehension of text,” in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 193–203.
- [14] S. Reddy, D. Chen, and C. D. Manning, “Coqa: A conversational question answering challenge,” *arXiv preprint arXiv:1808.07042*, 2018.
- [15] D. Chen, “Neural reading comprehension and beyond,” <https://cs.stanford.edu/~danqi/papers/thesis.pdf>.
- [16] Y. Kim, “Convolutional neural networks for sentence classification,” *arXiv preprint arXiv:1408.5882*, 2014.
- [17] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.
- [18] D. Chen, A. Fisch, J. Weston, and A. Bordes, “Reading wikipedia to answer open-domain questions,” *arXiv preprint arXiv:1704.00051*, 2017.
- [19] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.
- [20] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [21] S. Min, V. Zhong, L. Zettlemoyer, and H. Hajishirzi, “Multi-hop reading comprehension through question decomposition and rescoring,” *arXiv preprint arXiv:1906.02916*, 2019.

Improved DNN-HMM English Acoustic Model Specially For Phonotactic Language Recognition

Wei-Wei Liu, Guo-Chun Li, Cun-Xue Zhang, Hai-Feng Yan, Jing He, Ying-Xin Gan, Yan-Miao Song, Jian-Hua Zhou, Jian-Zhong Liu, Ying Yin, Ya-Nan Li*, Yu-Bin Huang, Ting Ruan, Wei Liu, Rui-Li Du, Hua-ying Bai, Wei Li, Sheng-Ge Zhang

Department of Electronic Engineering, Tsinghua University, Beijing 100842, China, liu-ww10@hotmail.com

**Academy of Military Science, Beijing 100091, China, bennyhappy@gmail.com*

Abstract—The now-acknowledged sensitive of Phonotactic Language Recognition (PLR) to the performance of the phone recognizer front-end have spawned interests to develop many methods to improve it. In this paper, improved Deep Neural Networks Hidden Markov Model (DNN-HMM) English acoustic model front-end specially for phonotactic language recognition is proposed, and series of methods like dictionary merging, phoneme splitting, phoneme clustering, state clustering and DNN-HMM acoustic modeling (DPPSD) are introduced to balance the generalization and the accusation of the speech tokenizing processing in PLR. Experiments are carried out on the database of National Institute of Standards and Technology language recognition evaluation 2009 (NIST LRE 2009). It is showed that the DPPSD English acoustic model based phonotactic language recognition system yields 2.09%, 6.60%, 19.72% for 30s, 10s, 3s in equal error rate (EER) by applying the state-of-the-art techniques, which outperforms the language recognition results on both TIMIT and CMU dictionary and other phoneme clustering methods.

Keywords-DPPSD English acoustic model; Phonotactic language recognition

I. INTRODUCTION

Language recognition is the process of identifying a language from an utterance, which is an enabling technology in many applications, such as spoken document retrieval, speech translation, information security and forensics and multilingual speech recognition [1]. Currently, acoustic systems [2] and phonotactic systems [1] are two broad kinds of language recognition systems that are widely used. Phonotactic language recognition (PLR) is based on the assumption that phonotactic constraints contain information to identify the languages.

A typical parallel phone recognizer followed by vector space model (PPR-VSM) language recognition system is illustrated in Figure 1, where a collection of parallel phone recognizers is employed to convert the utterances into phone lattices by Viterbi algorithm according to the given acoustic model of phonemes without language models, then the lattices are used to perform phonotactic analysis to classify languages in Support Vector Machine (SVM) [3]. It is obvious that the performance of the phone recognizer front-end affects the succeeding proceeding work of the language recognition system. So the work of building a dramatic acoustic model of phone recognizer plays an important role in language recognition system.

Usually, each phone recognizer has a phone inventory of a single specific language. To deduce the effect of the incorrectness in phone tokenizing, the phone recognizers in the PPR-VSM

system are usually trained either on multiple features like Mel Frequency Cepstral Coefficients (MFCC) and Perceptual Linear Predictive (PLP) or on different acoustic models [4], such as GMM-HMM, ANN-HMM [5] and DNN-HMM and then fuse the recognition results. All these methods achieve good performance but need to repeat the training process as much as the number of front-ends and the type of feature adopted, which usually compute at high computational cost.

Researchers also find that in training material, native speakers do not actually speak in the way the language is written or listening materials is represented. The perceptual saliency of spoken English is often reduced creating variation in the way English is spoken in very formal contexts from more naturally occurring English, which is called “reduced forms” [6]. Reduced forms refers to basic elements of this naturally occurring spoken English, integral and pervasive elements of spoken English, that are seriously neglected in phoneme modeling.

In the language recognition task, the phone recognizer is commonly based on the same structure with that is used in speech recognition (SR) system. Actually, the requirements for the phone recognizers in PLR and SR are different due to the difference of the recognizing ranges and goals. Speech recognition is a language-dependent task, the relationship of phonemes is relatively less than that in language recognition. So the acoustic model is acquired to contain as much states as possible to describe the specific language more precisely, which limits the generalizing ability of the acoustic model. While the language recognition is a language-independent task, the phone recognizer needs to carry on in many languages of the phone recognizing task, which requests high consistency and the robustness of phone recognizing in the different language.

In this paper, an acoustic model is proposed specially for the language recognition tasks to solve the problems mentioned above, whose building process contains dictionary merging, phoneme splitting, phoneme clustering, state clustering and DNN-HMM acoustic modeling (DPPSD). DPPSD acoustic modeling method takes account on the reduced forms of the English spoken training materials and calculates the distance measure at the state-level, which is more accurate than other methods based on the phone-level alignment such as the acoustic likelihood method, can get a phone inventory that achieves more superior language recognizing performance.

The remainder of the paper is organized as follows: Section 2 shows the difference between the DPPSD English acoustic model and the traditional English acoustic model in formulation and

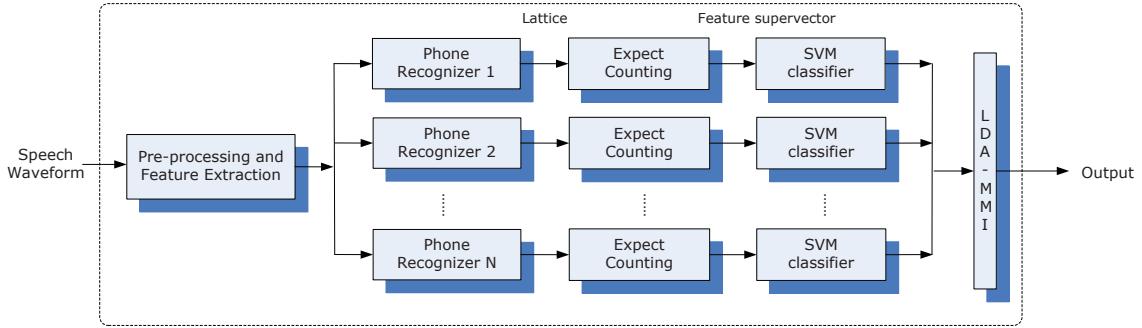


Figure 1. Architecture of PPR-VSM system.

the DPPSD phonotactic language recognition system. Section 3 shows the experimental setup. To evaluate the performance of the proposed DPPSD acoustic model, experiments are carried out and the results are analyzed in Section 4. Finally, Section 5 concludes the paper.

II. DPPSD ACOUSTIC MODEL BASED PHONE RECOGNIZER SPECIALLY FOR PHONOTACTIC LANGUAGE RECOGNITION

DPPSD phone recognizer front-end based phonotactic language recognition system is shown in Figure 2. The building process contains English dictionary merging, phoneme splitting, phoneme clustering, state clustering and DNN-HMM acoustic modeling.

A. English Dictionary Merging

In traditional speech recognition, the CMU [7] dictionary or TIMIT dictionary [8] is often used to define the standard English pronunciation. The CMU dictionary contains a total of 133,354 words' pronunciations labelled by ARPAbet symbol set, which is developed by the Advanced Research Projects Agency (ARPA). A total of 39 phonemes are contained in ARPAbet symbol set, in which are 24 consonants and 15 vowels. The TIMIT dictionary contains a total of 6,229 words' pronunciations labelled by American English pronunciation dictionary symbols developed by Kenyon and Knott of the DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus, which contains 27 consonants and 18 vowels in 45 phonemes.

There are two sides of the problem need to take into account when developing the phoneme set for the English phone recognizer for language recognition. On one hand, the more accurate description of the English phonemes are involved in the training speech, the more accurate acoustic model can be built; On the other hand, the expensive cost of organizing training resources such as training data annotation and dictionaries must be considered, so existing standard pronunciation dictionaries and English voice data annotation resources are needed to make maximum use. The TIMIT phonetic description of the various pronunciations are more subtle than the CMU phonetic transcription but contains only 6,229 words' pronunciations. The rich resource of CMU dictionary indicates that it can be an effective compensation for the TIMIT dictionary. In this paper a refined dictionary is developed using the detailed features of CMU dictionary words and TIMIT phoneme sets to describe phonemes with more standard pronunciation definitions.

The improved pronunciation dictionary of English phone recognizer in this paper is based on the CMU dictionary. The difference between CMU dictionary and TIMIT dictionary is a small number of phonemes in the CMU dictionary that have different corresponding relationships with TIMIT phonemes according to the syllable structure and the change in stress, and the remaining phonemes correspond to TIMIT dictionary phonemes one by one. Therefore, the pronunciation dictionary of English phone recognizer in this paper is constructed by putting the same phonemes of TIMIT and CMU together and subdividing the phonemes according to reduced form of spoken English.

B. Reduced Form of Spoken English and Phone Splitting

All languages have this type of variation from written to spoken texts: "It results from a simple law of economy, whereby the organs of speech, instead of taking a new position for each sound, tend to draw sounds together with the purpose of saving time and energy". With English, this process of assimilation is combined with contractions, elision, and reduction to produce the connected speech commonly referred to as "reduced forms". Naturally occurring English conversation, whether formal or informal, fast or slow, is full of these reduced forms. This creates a serious challenge for building acoustic model. In practical applications, it is necessary to adjust the pronunciation dictionary according to the reduced forms of spoken English, divide the phonemes that have the reduced pronunciation into another independent phonemes and build acoustic model for them to describe the syllable pronunciation more accurately.

C. Phoneme clustering

In this paper a State-Time-Alignment (STA) [9] phoneme clustering method is proposed to balance the performance and the complexity of the phone recognizer. There are two kinds of popular methods widely used, one is to search an universal and compact phone inventory by using phonetic knowledge [10] such as the International Phonetic Alphabet (IPA), the other is to merge the phone models using data-driven clustering method based on model-distance measures such as acoustic likelihood [11] or Bhattacharyya [12] distance. These methods have been used in Automatic Speech Recognition (ASR) systems and improve the performance of ASR, however, they are not suitable for language recognition. For language recognition work involves many type of languages, it is very important to determine an accurate phone inventory for the phone recognizer front-end of the language recognition system. STA method calculates the

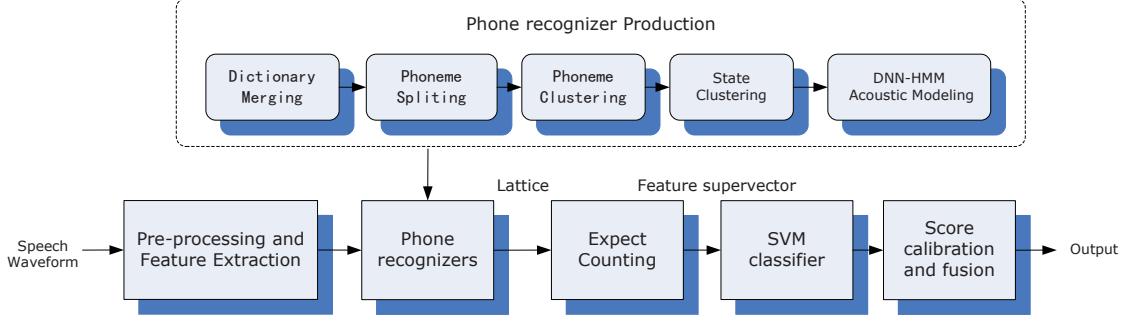


Figure 2. Flowchart of English phone recognizer production.

distance measure at the state-level, which is more accurate than other methods based on the phone-level alignment such as the acoustic likelihood method, that can get a phone inventory that achieves more superior language recognizing performance.

In STA clustering method, the distance between phone model p and q are defined as:

$$D(p, q) = \frac{1}{2} \left(\sum_{j,k} c(p_j, q_k) D(p_j, q_k) + \sum_{j,k} c(q_j, p_k) D(q_j, p_k) \right), \quad (1)$$

here $c(p_j, q_k)$ is the count of times t where model p has its state j active and model q has its state k active, which is calculated as:

$$c(p_j, q_k) = \sum_i \sum_{t=1}^{T_i} \gamma_{p_j}(t) \gamma_{q_k}(t), \quad (2)$$

in which i is the sub-segment index, T_i is the ending time of each sub-segment, $\gamma_{p_j}(t)$ and $\gamma_{q_k}(t)$ are occupation probabilities (0 or 1) defined in the forward-backward algorithm.

$D(p_j, q_k)$ is the distance of the j th state of model p and the k th state of model q in sub-segment by the Bhattacharyya distance measure, which is calculated as:

$$\begin{aligned} D(p_j, q_k) &= \frac{1}{8} (\mu_{p_j} - \mu_{q_k})^T \left[\frac{\Sigma_{p_j} + \Sigma_{q_k}}{2} \right]^{-1} (\mu_{p_j} - \mu_{q_k}) \\ &\quad + \frac{1}{2} \ln \frac{\left| \frac{\Sigma_{p_j} + \Sigma_{q_k}}{2} \right|}{\sqrt{\left| \Sigma_{p_j} \right| \left| \Sigma_{q_k} \right|}}, \end{aligned} \quad (3)$$

where μ_{p_j} and Σ_{p_j} denote mean vector and variance of the j th Gaussian state of phone model p .

In STA processing, we merge two phone models according to the minimum distance, in which the parameters of model l are updated as:

$$\mu_{l_y} = \frac{m_{p_y} \mu_{p_y} + m_{q_y} \mu_{q_y}}{m_{p_y} + m_{q_y}}, \quad (4)$$

$$\Sigma_{l_y} = \frac{m_{p_y} (\mu_{p_y}^2 + \Sigma_{p_y}) + m_{q_y} (\mu_{q_y}^2 + \Sigma_{q_y})}{m_{p_y} + m_{q_y}} - \mu_{l_y}^2 \quad (5)$$

here m_{p_y} and m_{q_y} denotes the data statistics of y th state of model p and model q . The merging process is repeated until desired number phone classes is reached. All the phone models are turning into new DPPSD phoneme set after phoneme clustering. The dictionary, questions for decision tree and transcription files for training data are all processed based on the clustering information.

D. DPPSD Phone Recognizer Front-end Based Phonotactic Language Recognition

After the DPPSD acoustic modeling, the input data x is mapped from input space \mathcal{X} into a new high dimensional DPPSD feature space $\mathcal{F}_{\text{DPPSD}}$: $\Phi : \mathcal{X} \rightarrow \mathcal{F}_{\text{DPPSD}}$, and then linear machines is builded to classify in the feature space.

Then training or test utterance x is mapped into the DPPSD feature space as:

$$\Phi_{\text{DPPSD}} : x \rightarrow \varphi_{\text{DPPSD}}(x). \quad (6)$$

and the N-gram feature supervectors $\varphi(x)$ is calculated as:

$$\varphi_{\text{DPPSD}}(x) = [p(d_1|\ell_x), p(d_2|\ell_x), \dots, p(d_F|\ell_x)], \quad (7)$$

here d_q is the DPPSD phone inventory based N-gram phoneme string $d_q = s_q \dots s_{q+N-1}$ ($n = N$) and $F = f^N$ (f denotes the size of the DPPSD based phone inventory). ℓ_x denotes the DPPSD phone inventory based lattice converted from data x . $p(d_q|\ell_x)$ is the probability of the N-gram $s_q \dots s_{q+N-1}$ in the lattice ℓ_x .

Also the probability of the phone sequence $s_q \dots s_{q+N-1}$ in the DPPSD phone inventory based lattice is calculated as follows:

$$p(s_q \dots s_{q+N-1} | \ell_x) = \frac{c(s_i \dots s_{i+N-1} | \ell_x)}{\sum_{\forall m} c(s_m \dots s_{m+N-1} | \ell_x)}, \quad (8)$$

Given the DNN-HMM based DPPSD acoustic model Λ_{DNN} , the expected counts over all possible hypotheses in the lattice of speech utterance x are computed as follows [13]:

$$\begin{aligned} &c(s_i, \dots, s_{i+N-1} | \ell_x) \\ &= E[c(s_i, \dots, s_{i+N-1}) | X, \Lambda_{\text{DNN}}, M'] \\ &= \sum_{s_i \dots s_{i+N-1} \in \ell_x} [\alpha(s_i) \beta(s_{i+N-1}) \prod_{j=i}^{i+N-1} \xi(s_j)], \end{aligned}$$

where M is the estimates of the N-gram probabilities that maximize $\sum_H f(X|H, \Lambda_{\text{DNN}}) P(H|\mathcal{L})$ ($H = s_i \dots s_{i+N-1}$, \mathcal{L} is the language under consideration, $f(X|H, \Lambda_{\text{DNN}})$ is the likelihood of the speech utterance X given \mathcal{L} and H). $\alpha(s_i)$ is the forward probability of the starting node of $s_i \dots s_{i+N-1}$ and $\beta(s_{i+N-1})$ is the backward probability of the ending node of $s_i \dots s_{i+N-1}$. $\xi(s_j)$ denotes the posterior probability of the edge s_j .

Then the SVM output score is calculated as follows:

$$\begin{aligned} &f(\varphi_{\text{DPPSD}}(x)) \\ &= \sum_v \alpha_v K_{\text{TFLLR}}(\varphi_{\text{DPPSD}}(x), \varphi_{\text{DPPSD}}(x_v)) + d, \end{aligned} \quad (9)$$

here $\varphi_{\text{DPPSD}}(x_v)$ are support vectors. The TFLLR kernel is calculated as [14]:

$$K_{\text{TFLLR}}(\varphi_{\text{DPPSD}}(x_i), \varphi_{\text{DPPSD}}(x_j)) = \sum_{q=1}^F \frac{p(d_q|x_i)}{\sqrt{p(d_q|\ell_{all})}} * \frac{p(d_q|x_j)}{\sqrt{p(d_q|\ell_{all})}}, \quad (10)$$

Then the posterior probabilities of all the belief score vector are maximized using the LDA-MMI algorithm [15] with objective function as follows [16]:

$$F_{\text{MMI}}(\lambda) = \sum_{\forall i} \log \frac{p(\mathbf{x}_i|\lambda_{g(i)})P(g(i))}{\sum_{\forall j} p(\mathbf{x}_i|\lambda_j)P(j)}, \quad (11)$$

here $\mathbf{x} = [f(\varphi_{\text{DPPSD}}(x))]$ and $g(i)$ indicates its class label. $P(j)$ is the prior probability of class j . $p(\mathbf{x}|\lambda)$ is weighted Gaussian mixtures that describe a general distribution:

$$p(\mathbf{x}|\lambda) = \sum_{\forall m} \omega_m \mathcal{N}(x; \mu_m, \Sigma_m), \quad (12)$$

here $\mathcal{N}(\cdot)$ denotes the normal distribution with a parameter set that is often referred as $\lambda = \{\omega_m, \mu_m, \Sigma_m\}$. Here μ_m , Σ_m and ω_m are the mean vector, covariance matrix and the weight of the m -th Gaussian mixture.

Such acoustic model has three advantages. First, STA method calculates the distance measure at the state-level, which is more accurate than other methods based on the phone-level alignment such as the acoustic likelihood method, that can get a phone inventory that achieves more superior language recognizing performance. Second, DPPSD acoustic model adds dictionary merging, phoneme splitting, phoneme clustering, state clustering in building processing than the traditional acoustic model, while it costs no more computation in decoding but gets a remarkable improvement. Third, DPPSD based phone recognizer can explore more discriminative information than traditional phone recognizer, then more effective information can be extracted for the language recognition system to classify.

III. EXPERIMENTAL SETUP

A. System setup

A PR-SVM language recognition system is used as baseline system in this paper. The first step is to tokenize speech by the means of running phone-recognizers and the decoder named HVite produced by HTK [17] is used to produce phone lattices, and an open software named lattice-tool (SRILM) [18] is used to provide the posterior probabilities of the phone occurrences to produce phone counts. Then, a popular classifier LIBLINEAR [19] is employed to classify the feature supervector. Finally, LDA-MMI algorithm [20] is used for score calibration among the different acoustic models.

The DPPSD English DNN-HMM acoustic model is trained using about 100 hours' Switchboard English corpus [21]. The same training algorithm are applied to train DNNs as in [22] in this work. Input features to DNNs in the training stage are 13-dimensional PLP features plus their first and second order derivatives, which are all normalized to have zero mean and unit variance based on conversation-side information [23]. A CUDAMat library [24] is used as the implementations of the DNN.

B. Training, test and developing database

About 180,000 training data used in this paper belongs to: (1) the Call-Home Corpus; (2) the OHSU Corpus provided by NIST LRE 2005; (3) the VOA Corpus; (4) the OGI Corpus; and (5) the Call-Friend Corpus.

The experiments are conducted on the test trials of the National Institute of Standards and Technology Language Recognition Evaluation 2009 (NIST LRE 2009) tasks. The test database comprises 41793 test segments of 23 languages for 30-s, 10-s, and 3-s nominal duration test.

22701 conversations are selected from the database provided by NIST for the 2003, 2005 and 2007 LRE and VOA for developing purposes.

C. Evaluation measures

The performance of language recognition system in this paper is reported by Equal Error Rate (EER) and average cost performance (C_{avg}), which are defined by NIST LRE 2009 [25].

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we present our experimental results of the language recognition system using English phone recognizer on the NIST LRE 2009. More than 1,800,000 utterances are used for training.

A. Comparison of different kinds of dictionaries

The performance of different kinds of dictionary based language recognition system are investigated in this subsection and the EER and C_{avg} of the different front-end language recognition system are listed in Table I, in which the dictionary of 39 phonemes is CMU and the dictionary of 45 phonemes is TIMIT.

It can be seen from the results of Table I that the number of the phonemes of the dictionary is varied from 39 to 58. For each fixed frontend, the performance of the language recognition system first decreases and then increases with the decreasing of number of the phonemes inventory of the acoustic model, and the best performance occurs at the phone inventories of size 47. In fact the performance of the 47 phonemes' DPPSD English dictionary based phonotactic language recognition outperforms that of both CMU and TIMIT based system, which means the balance point of generalization and accuration of the acoustic model. Notice that the performance of the 47 phonemes' DPPSD English dictionary based phonotactic language recognition also outperforms the systems whose acoustic model using knowledge-based and data-driven phoneme clustering method. So when 47 phonemes DPPSD set English dictionary is adopted and the language recognition system achieves the best performance.

B. Effects of the number of GMMs' states

Table II shows the language recognition performance of different number of GMMs' states based English DNN-HMM acoustic model. The English GMM-HMM triphone acoustic model contains from 144 to 9308 [23] states with 32 Gaussians each. Note that 9308 GMMs' states acoustic model corresponds to the acoustic model used in speech recognition system. Better performance is achieved using 154-states DNN-HMM acoustic model than that of 258-states, 904-states and 9032-states, that means the 154-states acoustic model is more generalizing and robust in recognizing phonemes of different languages than

Table I
PERFORMANCE OF DIFFERENT KINDS OF DICTIONARY BASED LANGUAGE RECOGNITION SYSTEM, NIST LRE 2009, (EER/CAVG IN %).

Number of phonemes		30s	10s	3s
		EER	EER	EER
Baseline System	39(CMU)	2.37	7.04	21.42
	45(TIMIT)	2.23	6.84	20.20
Phoneme Clustering	knowledge-based	2.19	6.99	20.26
	data-driven	2.13	6.80	20.17
STA	58	2.29	6.93	20.62
	53	2.26	6.81	20.12
	51	2.18	6.69	20.15
	47	2.09	6.66	19.72
	46	2.13	6.68	20.04

other acoustic models, even than the acoustic model used in speech recognition system. Therefore, English DNN-HMM acoustic model with 47 phoneme DPPSD dictionary and sigmoidal networks of 154 GMM-HMM states are used for all the following experiments.

Table II
PERFORMANCE OF DNN-HMM ACOUSTIC MODEL WITH DIFFERENT GMMs' STATES, NIST LRE 2009 (EER/CAVG IN %).

Number of GMMs' States	30s	10s	3s
	EER	EER	EER
9032	2.70	8.45	23.83
904	2.42	7.33	21.24
258	2.27	6.97	20.11
154	2.09	6.66	19.72

C. Comparison of different kinds of acoustic model

Table III compares the results of language recognition system using an ANN-HMM model, a GMM-HMM model with 154 states and a DNN-HMM model, which are all trained using the 100-hour subset of English Switchboard corpus. ANN-HMM acoustic model is trained using TRAP feature and a context window of 21 frames. Experiments show that DNN-HMM acoustic model provide dramatic improvements in language recognition accuracy and offers a relative EER reduction of 28.42%, 14.06%, 18.70% over the ANN-HMM acoustic model, and a relative EER reduction of 12.55%, 7.20%, 2.47% over the GMM-HMM acoustic model for 30s, 10s and 3s, respectively. The performance of longer speech utterances (30s) improves more dramatically than that of 10s and 3s because DNNs are more powerful in modeling long context acoustic events than GMM-HMMs. Figure 3 shows the DET curves for NIST LRE 2009.

D. Comparison of real time factor for decoding

Table IV shows decoding real time factor of TIMIT, CMU and DPPSD acoustic model. Although the training cost of DPPSD acoustic model is a little expensive compared with training TIMIT and CMU acoustic model, decoding in DPPSD acoustic model is very efficient because the structure of DNN-HMM acoustic model is same with TIMIT and CMU acoustic model,

Table III
PERFORMANCE OF DIFFERENT KINDS OF ACOUSTIC MODEL BASED SYSTEMS, NIST LRE 2009, (EER/CAVG IN %).

Acoustic model of phone recognizer	30s	10s	3s
	EER	EER	EER
EN-GMM-HMM (39 phonemes)	2.54	7.88	21.20
EN-GMM-HMM (47 phonemes)	2.39	7.20	20.22
EN-ANN-HMM (39 phonemes)	3.13	8.30	25.52
EN-ANN-HMM (47 phonemes)	2.92	7.75	23.41
EN-DNN-HMM (39 phonemes)	2.37	7.04	21.42
EN-DNN-HMM (47 phonemes)	2.09	6.66	19.72
EN fusion	1.39	4.28	15.83

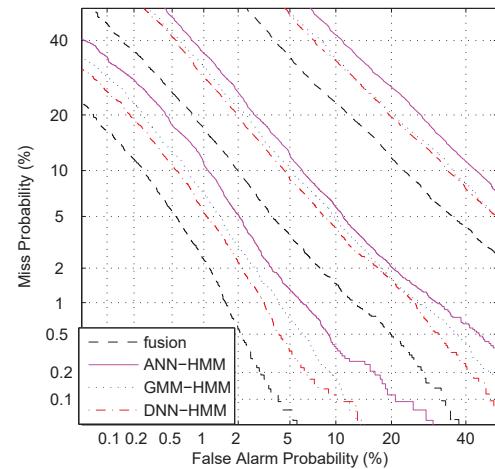


Figure 3. DET curves for NIST LRE 2009, English phone recognizer frontend.

the decoding real time factor only increases with the increasing of the number of phoneme inventory.

Table IV
COMPARISON OF REAL TIME FACTOR FOR DECODING, NIST LRE 2009, 30-S TEST. CPU: XEON E5520@2.27 GHZ, RAM: 8 GB, SINGLE THREAD. GPU: GEFORCE GTX 275, RAM: 1 GB, 240 CUDA CORE.

acoustic model	TIMIT	CMU	DPPSD
RT factor	0.069	0.064	0.071

V. CONCLUSION

This paper has presented how to build a dramatic English acoustic model specially for phone recognizer front-end of phonotactic language recognition. The DPPSD English acoustic model is generalizing and robust to different languages, which is more suitable for language recognition than that is used in speech recognition and increases the performance of the system on accuracy without sacrificing its structure simplicity and computational effort. The experimental results evaluated on NIST LRE 2009 tasks have confirmed that the proposed DPPSD English acoustic model based system yields an EER of 2.07%, 6.66% and 19.22%, which achieves a 11.81%, 5.39% and 7.94%

relative deduction for 30s, 10s and 3s, respectively compared with the traditional CMU dictionary based English acoustic model, and a 6.28%, 2.63% and 7.94% relative deduction for 30s, 10s and 3s, respectively compared with the traditional TIMIT dictionary based English acoustic model.

As for future work, we will develop effective adaptation techniques of DNN and use DPPSD method to build multilingual phone recognizer. More work needs to be done, especially, in the direction of understanding which TDNN architectures would work best for phone recognition, and how to find such architecture, including determining good loss and learning rate, activation functions, cost optimizers, etc.

ACKNOWLEDGMENTS

Project supported by the National Natural Science Foundation of China (No.61370034, No.61403224 and No.61273268). Yan-Nan Li is the corresponding author.

REFERENCES

- [1] Zissman, M.A.: Comparison of four approaches to automatic language identification of telephone speech. *IEEE Transactions on Speech and Audio Processing* **4**(1), 33–44 (1996)
- [2] Torres-Carrasquillo, P.A., Singer, E., Kohler, M.A., Greene, R.J., Reynolds, D.A., Deller, J.R.: Approaches to language identification using gaussian mixture models and shifted delta cepstral features. *Proc. ICSLP*, 33–36 (2002)
- [3] Li, H., Ma, B., Lee, K.-A.: Spoken language recognition: from fundamentals to practice. *Proceedings of the IEEE* **101**(5), 1136–1159 (2013)
- [4] Sim, K.C., Li, H.: On acoustic diversification front-end for spoken language identification. *IEEE Trans. on Audio, Speech and Language Processing* **16**(5), 1029–1037 (2008)
- [5] Schwarz, P.: Phoneme recognition based on long temporal context (2009)
- [6] Brown, J.D., Helferty, A., Enright, S.: Listening for reduced forms. *Tesol Quarterly* **20**(4), 759–763 (2012)
- [7] University, C.M.: Cmu pronouncing dictionary
- [8] Zue, V.W., Seneff, S.: Transcription and alignment of the timit database (1996)
- [9] Qian, Y., Jia, L.: Phone modeling and combining discriminative training for mandarinenglish bilingual speech recognition. In: *IEEE International Conference on Acoustics Speech & Signal Processing* (2010)
- [10] Yu, S., Zhang, S., Bo, X.: Chinese-english bilingual phone modeling for cross-language speech recognition. In: *IEEE International Conference on Acoustics* (2004)
- [11] Hller, J.: Multilingual Phone Models for Vocabulary-independent Speech Recognition Tasks, (2001)
- [12] Mak, B., Barnard, E.: Phone Clustering Using the Bhattacharyya Distance, (1996)
- [13] Gauvain, J.L., Messaoudi, A., Schwenk, H.: Language Recognition Using Phone Lattices. In: *Proc. ICSLP*, Jeju Island, pp. 1283–1286 (2004)
- [14] Campbell, W.M., Campbell, J.P., Reynolds, D.A., Jones, D.A., Leek, T.R.: Phonetic speaker recognition with support vector machines. *Advances in neural information processing systems* **16** (2003)
- [15] Matejka, P., Burget, L., Glembek, O., Schwarz, P., Hubeika, V., Fapso, M., Mikolov, T., Plchot, O.: BUT system description for NIST LRE 2007. In: *Proc. 2007 NIST Language Recognition Evaluation Workshop*, pp. 1–5 (2007)
- [16] Povey, D.: Discriminative training for large vocabulary speech recognition. Cambridge, UK: Cambridge University **79** (2004)
- [17] Young, S., et al: The HTK book. Cambridge University Engineering Department **3** (2002)
- [18] Stolcke, A.: SRILM - An Extensible Language Modeling Toolkit. [Online]. Available: <http://www.speech.sri.com/projects/srilm/> (2002)
- [19] Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., Lin, C.-J.: Liblinear: A library for large linear classification. *The Journal of Machine Learning Research* **9**, 1871–1874 (2008)
- [20] Zhang, W.-Q., Hou, T., Liu, J.: Discriminative score fusion for language identification. *Chinese Journal of Electronics* **19**, 124–128 (2010)
- [21] Godfrey, J.J., Holliman, E.C., McDaniel, J.: Switchboard: Telephone speech corpus for research and development. In: *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference On*, vol. 1, pp. 517–520 (1992). IEEE
- [22] Dahl, G.E., Yu, D., Deng, L., Acero, A.: Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on* **20**(1), 30–42 (2012)
- [23] Cai, M., Shi, Y., Liu, J.: Deep maxout neural networks for speech recognition. In: *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop On*, pp. 291–296 (2013). IEEE
- [24] Mnih, V.: Cudamat: a cuda-based matrix class for python. Department of Computer Science, University of Toronto, Tech. Rep. UTMTR **4** (2009)
- [25] NIST: The 2009 NIST language recognition evaluation plan. In: <Http://www.itl.nist.gov/iad/mig/tests/lang/2009/> (2009)

Confidence Modeling for Neural Machine Translation

Taichi Aida

*Nagaoka University of Technology
Nagaoka, Niigata, Japan
aida@jnlp.org*

Kazuhide Yamamoto

*Nagaoka University of Technology
Nagaoka, Niigata, Japan
yamamoto@jnlp.org*

Abstract—Current methods of neural machine translation output incorrect sentences together with sentences translated correctly. Consequently, users of neural machine translation algorithms do not have a way to check which outputted sentences have been translated correctly without employing an evaluation method. Therefore, we aim to define the confidence values in neural machine translation models. We suppose that setting a threshold to limit the confidence value would allow correctly translated sentences to exceed the threshold; thus, only clearly translated sentences would be outputted. Hence, users of such a translation tool can obtain a particular level of confidence in the translation correctness. We propose some indices; sentence log-likelihood, minimum variance, and average variance. After that, we calculated the correlation between each index and bilingual evaluation score (BLEU) to investigate the appropriateness of the defined confidence indices. As a result, sentence log-likelihood and average variance calculated by probability have a weak correlation with the BLEU score. Furthermore, when we set each index as the threshold value, we could obtain high quality translated sentences instead of outputting all translated sentences which include a wide range of quality sentences like previous work.

Keywords-machine translation; confidence estimation;

I. INTRODUCTION

Neural machine translation (NMT) [1], which uses neural networks for the purposes of machine translation (MT), is one of the most discussed subjects in the field of natural language processing. NMT has a great impact on society; it is applied to Google Translate and is used by various companies.

In Japan, the Tokyo Olympics will be held in 2020. At this event, many tourists from all over the world are expected to come to Japan, and their language needs to be addressed. Hence, the demand for MT is increasing even more.

MT systems are expected to output high-quality translations. However, recent systems may generate a wide range of quality sentences. Therefore, users cannot ensure which outputted sentences were translated correctly, and which ones were not, until an evaluation method is applied. For a user, it is a serious and time-consuming problem to check all output sentences without information on the quality of outputted translation obtained by a MT system. Although there are some methods to evaluate the translation result, BLEU [2], which is a typical evaluation method, requires a reference translation as an answer, and the evaluation model based on the task called Quality Estimation [3]–[8] uses a large corpus for training.

To solve this issue, we seek to define the confidence in NMT models. Information on the translation confidence would allow understanding whether an output sentence has been translated with a high level of confidence or not. Furthermore, when a user sets a threshold and to prevent a model from outputting sentences with the confidence value lower than the threshold, the user of such a translation tool can have guarantees that the outputted translation is correct and focus on the other translations.

In this paper, we propose the indices to define the confidence value. Next, in order to investigate whether the proposed index can be used for confidence, we correlated it with the BLEU score, the main evaluation method of machine translation. After that, we used a reasonable index of confidence as a threshold, and examined the number of sentences output and the average of BLEU by changing the threshold. Our method requires neither references nor an external large-scale corpus which is necessary to train an evaluation model. We suppose that we can obtain high-quality outputted sentences by setting a threshold to limit the output with the confidence value.

The experimental results show that some indices are correlated with the BLEU score, which means that these indices are appropriate as confidence values. In addition, when we use each index as a threshold value, we can filter high quality sentences from the output, which is unlike the previous works.

II. RELATED WORK

A. Confidence Estimation

In statistical machine translation, a confidence estimation task was defined [9]–[11]. In that task, the system labels each word in outputted sentences to correct or incorrect.

To validate the result, a rating scale called Classification Error Rate (CER) had been used:

$$CER = \frac{\text{Count}(\text{incorrect-label})}{\sum_{word \in sent} \text{Count}(word)} \quad (1)$$

CER, as it is suggested by its name, is defined as a mislabeled percentage of all words labeled by the system.

B. Automatic Evaluation

To evaluate various sentences outputted by machine translation, there are two classical methods. One is a major evaluation method, BLEU. The other is a quality

estimation method based on using a large parallel corpus.

1) *BLEU*: BLEU [2] is a representative method for evaluating a translation result using a reference translation. The BLEU score is calculated by examining the number of matches of n-grams between the candidate and the reference translation. Specific formulas are shown below:

$$BLEU = \text{penalty} \left(\prod_{n=1}^4 p_n \right)^{1/4} \quad (2)$$

$$\text{penalty} = \min(1, \exp(1 - \frac{\text{len}_{\text{ref}}}{\text{len}_{\text{cand}}})) \quad (3)$$

$$p_n = \frac{\sum_{C \in \{\text{Cand}\}} \sum_{\text{ngram} \in C} \text{Count}_{\text{clip}}(\text{ngram})}{\sum_{C' \in \{\text{Cand}\}} \sum_{\text{ngram}' \in C'} \text{Count}(\text{ngram}')} \quad (4)$$

Finally, the BLEU score is calculated in the range of 0 to 100, and the higher is this value, the closer the output sentence is to the reference translation.

2) *Quality Estimation*: However, when we use a translation system, we often do not have sources and reference pairs. For this reason, a Quality Estimation (QE) task is defined which implies using only a parallel corpus and not using references.

This is an actively researched approach and is used to train a large amount of external parallel corpus as an evaluation model, and to use the obtained model as a new translation pair for evaluation.

Many works suggest extracting indices and using them as features of support vector regression (SVR) to build the QE model [3]–[6]. There are also other studies that suggest training neural-based QE models separately from translation models, and then these models demonstrated high correlation with manually evaluated test data in QE tasks [7], [8].

However, many QE systems require a large amount of a parallel corpus to train their evaluation model. In other words, it is difficult for these systems to adapt to language pairs with few bilingual corpora such as Asian languages.

Thus, normally, the output sentence is evaluated only from the outside. Therefore, we consider that it may be necessary to perform evaluation from the inside of the model, too.

In the semantic parsing task, the confidence of the outputted sentence was defined [12]. Several indices are proposed in the model and in the corpus used. When the confidence value is limited by a threshold, only the results with high accuracy are outputted.

Moreover, the other research shows that weights that are identified based on the attention mechanism often used in machine translation are appropriate for estimating the confidence [13]. They asserted attention weight, which can be useful for more purposes than just visualizations. They defined two metrics for confidence, coverage deviation penalty and absentmindedness penalty using attention weight α_{ji} between input token j and output token i .

$$CDP = -\frac{1}{J} \sum_j \log(1 + (1 - \sum_i \alpha_{ji})^2) \quad (5)$$

$$AP = -\frac{1}{I} \sum_i \sum_j \alpha_{ji} \log \alpha_{ji} \quad (6)$$

In the above equations J and I are the length of input and output sentence. Coverage deviation penalty (CDP) can penalize not only lacking attention but also too much attention per input token. Also, the absentmindedness penalty (AP) can indicate how scattered the attention weights in each token are. Finally, they defined the confidence value:

$$\text{conf}_{\text{attn}} = CDP + AP_{\text{output}} + AP_{\text{input}} \quad (7)$$

Inspired by their work, we define the confidence and its indices in NMT models. More information about the proposed approach is described in Section III.

III. PROPOSAL

In Section II, it was noted that many systems evaluate candidates from only outside normally. We think that it is necessary to perform evaluation also from inside of the model.

We propose a method to calculate confidence values inside of the model. Figure 1 gives an overview of our method. As shown in Figure 1, recent NMT models output all translated sentences which include high and low quality sentences. In our process, the model computes the confidence value for each sentence. We believe that we can get high-quality translated sentences by setting a threshold to limit the output with the confidence value. The advantage of this method is that it does not require a reference sentence for the translation, or a large-scale bilingual corpus from outside, as well as training on any other model used for evaluation.

Confidence indices are described in detail below.

A. Sentence log-likelihood

In various sequence-to-sequence (seq2seq) machine translation models, one word generated at each position has the highest probability assigned among several word candidates. Therefore, by taking the sum of log-probabilities of all outputted words and then, taking into account the average number of words, we consider that it may become an index of confidence without affecting the word length:

$$\text{likelihood}_{\text{sent}} = \frac{\sum_{\text{word} \in \text{sent}} \log p_{\text{word}}}{\sum_{\text{word} \in \text{sent}} \text{Count}(\text{word})} \quad (8)$$

B. Variance

As we mentioned earlier, machine translation models output the results with the probability being the highest one among other candidates. In this index, we calculate the degree of dispersion with probability that a word of the highest probability actually output for the top five candidates in each word of the output sentence:

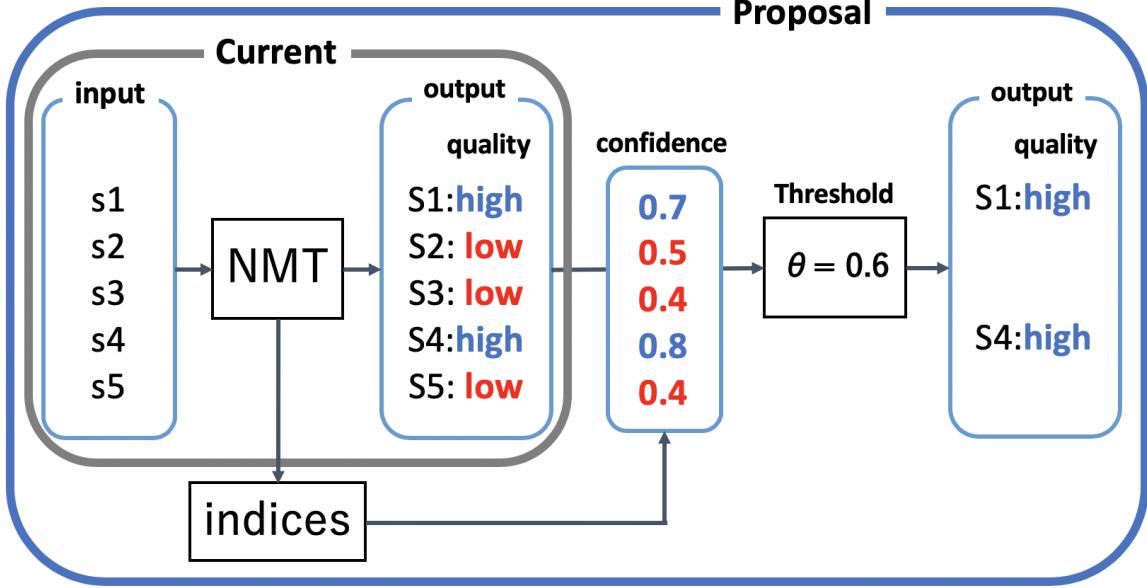


Figure 1: Overview of the proposed method; when sentences $s_{1:5}$ are inputted, current methods output all sentences $S_{1:5}$ which are included both high and low quality sentences. Our method computes confidence value from several indexes in the model and allows the model to output only sentences whose confidence value exceeds a threshold. We suppose that this process can obtain high-quality sentences by using no references and no large amount of parallel corpora to train other models. In this example, the threshold value is 0.6.

$$V_{top} = \frac{1}{4} \sum_{n=2}^5 (P_n - P_{top})^2 \quad (9)$$

Equation 9 is similar as in case of variance, the higher the V_{top} value, the farther the outputted word's probability outstands from the other candidates, and if the V_{top} value is low, it means the probability of outputted word is closer to them. We think that this index can represent “how much the model got lost when outputting each word.”

By using $\min(V_{top})$ with the smallest variance V_{top} for each word in the sentence thus obtained, it may be considered as an index of certainty factor.

In addition, we consider that the average of the variance of each word in each sentence can be used as an indices of certainty:

$$Ave(V_{top}) = \sum_{word \in sent} \frac{V_{top}(word)}{Count(word)} \quad (10)$$

We propose these two indices to define the confidence value. This method enables performing the evaluation in the process of generating the output translated sentences even in the environment available for a user (as they have only sentences to be translated) without the need in a large corpus.

IV. EXPERIMENT

First, we attempt to conduct this experiment to measure the appropriateness of each index we proposed. As this is a task similar to that of machine translation, the model is first trained on training and development data, thereafter, using the test data, the model translates source language

sentences into target language sentences. At the same time, the model calculates a confidence index for each sentence. Candidates are evaluated by the BLEU method. Consequently, the correlation is measured between confidence index and the BLEU score.

Second, we use index as a threshold value, and measure the average BLEU score and the number of outputted sentences when the threshold is changed.

A. Dataset

In order to train and test the translation model, we used ASPEC-JE corpus. ASPEC [14] is constructed by translating scientific papers. There are two language pairs, Japanese-English (ASPEC-JE) and Japanese-Chinese (ASPEC-JC). Details of the number of parallel pairs per a language pair are shown in Table I.

Lang-pair	Train	Dev	DevTest	Test
Ja-En	3,008,500	1,790	1,784	1,812
Ja-Cn	672,315	2,090	2,148	2,107

Table I: ASPEC, Asian Scientific Paper Excerpt Corpus, is constructed by translating scientific papers.

In ASPEC-JE, there are 3 train sets, one of each 1M pairs. They are arranged in descending order of similarity of language pairs, and this time we used only the top 1M pairs of a train-1 set with high similarity between parallel language pairs. Therefore, the actual number of parallel pairs are 1M/1,790/1,812 for Train/Dev/Test.

B. Models and Settings

In this experiment, we used the fairseq [15] transformer. Moreover, for the preprocessing purposes, the proposed

model used the functionality existing in fairseq and made a word of ten times or less appearance frequency into unknown token.

Details of parameters are as follows: lr=0.1, clip_norm=0.1, dropout=0.2, embedding dimension=300, beam_size=5, encoder and decoder layer=4, encoder and decoder attention_heads=5, max_epoch=100, and batch_size=64.

V. RESULTS/DISCUSSION

A. Appropriateness of indices

1) *Sentence log-likelihood*: Figure 2 shows the distribution of the sentence log-likelihood and the BLEU score in each sentence.

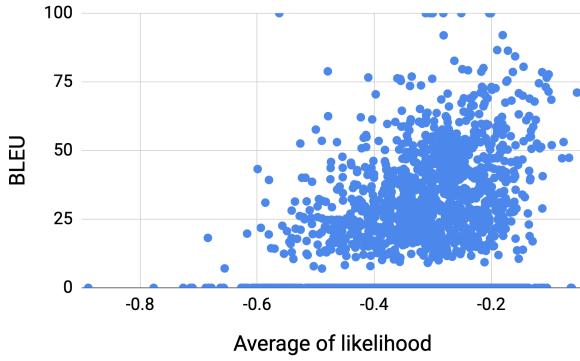


Figure 2: Distribution of log-likelihood and BLEU.

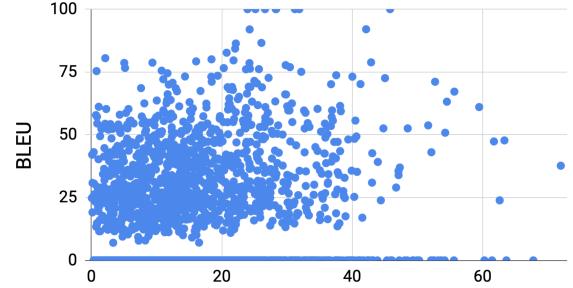
According to Figure 2, it has a triangular distribution, and the Pearson correlation coefficient is 0.308, which means that two indicators have a weak correlation. Therefore, sentence log-likelihood is suitable for confidence indices.

2) *Variance*: In this section, we analyze the usefulness of variance V_{top} .

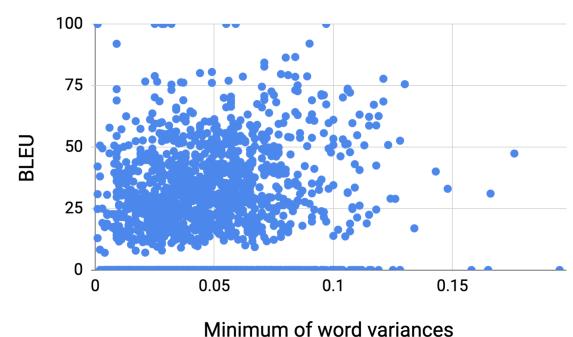
First, we examine the correlation between the minimum of variance each word in the sentence $\min(V_{top})$ and the BLEU score. When we calculate the variance, we use both log-probability and probability. The result is shown in Figure 3.

Figure 3 shows that the distribution is concentrated at the bottom left in both cases. The correlation coefficient with BLEU, in this case, is 0.077 for log-probability and 0.112 for probability. This result means that there is no correlation between the minimum variance calculated by either log-probability or probability and the BLEU score. Also, because the log-probability has a value range of 0 to minus infinity, and the variance is squared, we think that the correlation is lower than that of the probability. For that reason, we use only the probability to calculate variance in each word from this experiment. As a result, the minimum variance of each word in the sentence cannot be used for the index of confidence.

Second, we examine the correlation between the average of variance each word in the sentence $\text{Ave}(V_{top})$ and the BLEU score using the probability. The result is shown in Figure 4.



(a) Log-probability



(b) Probability

Figure 3: Distribution of the minimum variance of each word in the sentence and BLEU.

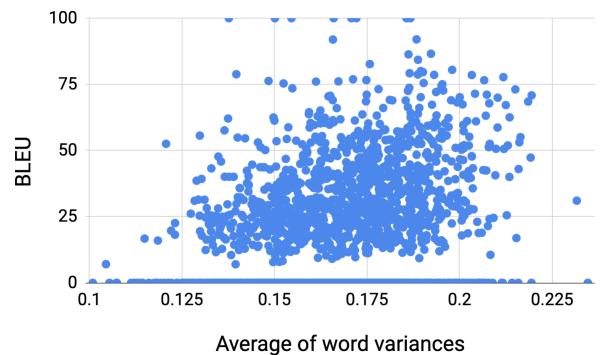


Figure 4: Distribution of average of variance and BLEU.

Similar to the sentence log-likelihood distribution presented in Figure 2, the distribution of the average variance is close to a triangle shape in Figure 4. Also, the Pearson correlation coefficient between the average variance and the BLEU score is 0.2676, which means low correlation between them. This result shows that the average variance $\text{Ave}(V_{top})$ calculated by probability can be used as the confidence indices.

According to these experiments, the sentence log-likelihood and the average of word variances of in each sentence can be used for the index of confidence.

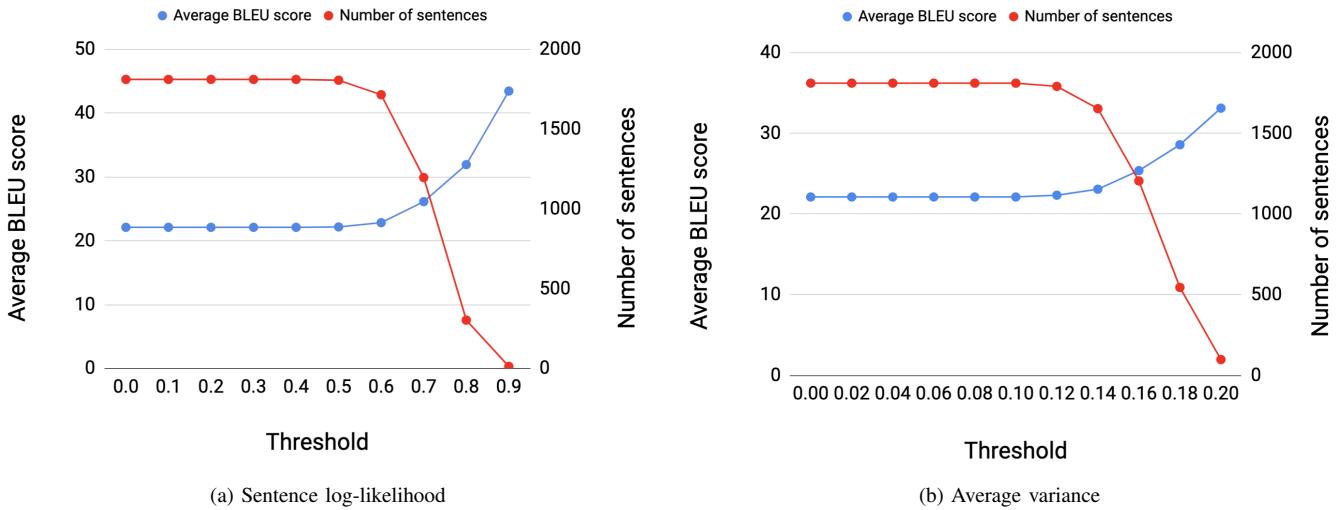


Figure 5: The number of sentences and the BLEU score when the threshold was changed. We used the sentence log-likelihood(left) and the average variance which is calculated by probability(right) as a threshold. If the threshold is high, the number of output sentences is low, but the average of the BLEU score is increased.

B. Using threshold

Next, we used sentence log-likelihood and average variance as threshold. We examined the number of outputted sentences and the average value of the BLEU score (Figure 5). When we set no threshold, 1,812 sentences for ASPEC-JE test set were output and the average of the BLEU score is 22.11.

According to Figure 5, it can be seen that increasing the threshold value in both cases reduces the number of output sentences and increases the average of the BLEU score.

1) *Sentence log-likelihood*: In Figure 5(a), the sentence log-likelihood is distributed from 0.6 to 0.9 when we used exponential to define its value between 0 to 1. When the threshold was set to 0.9, only 13 sentences were output, and the average value of BLEU is 43.45.

2) *Average variance*: On the other hand, the average variance is distributed from only 0.1 to 0.2 (Figure 5(b)). When we set the threshold to 0.2, 98 sentences were output, and the average value of BLEU is 33.12.

From these results, high-quality translated sentences can be obtained by using the sentence log-likelihood or the average variance as a threshold value to limit the output.

VI. CONCLUSION

In this paper, we analyzed the confidence and corresponding indices used in NMT models. Firstly, we proposed the confidence indices. Secondly, we analyzed the correlation between several indices and the BLEU score to verify if these indices can be used for confidence estimation. Thirdly, we set each index as the threshold value and examined the number of output sentences and the average of the BLEU score. In comparison to the previous work, this method requires neither a reference sentence as the answer to the translation, an external large-scale bilingual corpus, nor training any other model,

which is conventionally used for evaluation. As a result, the sentence log-likelihood and the average variance have a weak correlation with the BLEU score, which means these indices can be used as the component of confidence. Furthermore, when we set each index as the threshold value, we could filter high quality sentences from the output, which is unlike the previous works.

VII. FUTURE WORK

In the future, we plan to investigate the correlation between other indices and the BLEU score. For example, we consider using the attention weight [13] and unknown words. We think that if the probability of unknown words is higher than for other candidates and the model outputs $<unk>$, the model should have the low confidence about that part. By using this index, we consider that the largest proportion of the $<unk>$ in a statement that corresponds to an output sentence, and the confidence in the output statement in the model is low. We suppose this index can represent “how much the model does not know.”

Finally, we plan to consider combining several indices as a confidence factor and collecting only high-quality sentences by using the threshold. We hope that our work will help translators and other users be able to obtain only reliable translations from the machine translation systems.

ACKNOWLEDGMENT

This work was supported in part by JSPS KAKENHI Grants-in-Aid for Challenging Research (Exploratory) Grant ID 17K18481.

REFERENCES

- [1] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3104–3112.

- [2] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: A method for automatic evaluation of machine translation,” in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL ’02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 311–318.
- [3] L. Specia, G. Paetzold, and C. Scarton, “Multi-level translation quality prediction with QuEst++,” in *Proceedings of ACL-IJCNLP 2015 System Demonstrations*. Beijing, China: Association for Computational Linguistics and The Asian Federation of Natural Language Processing, Jul. 2015, pp. 115–120.
- [4] C. Scarton, D. Beck, K. Shah, K. Sim Smith, and L. Specia, “Word embeddings and discourse information for Quality Estimation,” in *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 831–837.
- [5] K. Shah, F. Bougares, L. Barrault, and L. Specia, “SHEFLIUM-NN: Sentence level quality estimation with neural network features,” in *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 838–842.
- [6] Z. Chen, Y. Tan, C. Zhang, Q. Xiang, L. Zhang, M. Li, and M. Wang, “Improving Machine Translation Quality Estimation with Neural Network Features,” in *Proceedings of the Second Conference on Machine Translation*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 551–555.
- [7] J. Ive, F. Blain, and L. Specia, “deepQuest: A framework for neural-based quality estimation,” in *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 3146–3157.
- [8] J. Wang, K. Fan, B. Li, F. Zhou, B. Chen, Y. Shi, and L. Si, “Alibaba submission for WMT18 quality estimation task,” in *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*. Belgium, Brussels: Association for Computational Linguistics, Oct. 2018, pp. 809–815.
- [9] J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing, “Confidence estimation for machine translation,” in *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*. Geneva, Switzerland: COLING, aug 23-aug 27 2004, pp. 315–321.
- [10] R. Soricut and A. Echihabi, “TrustRank: Inducing trust in automatic translations via ranking,” in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden: Association for Computational Linguistics, Jul. 2010, pp. 612–621.
- [11] N. Ueffing and H. Ney, “Word-level confidence estimation for machine translation using phrase-based translation models,” in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Vancouver, British Columbia, Canada: Association for Computational Linguistics, Oct. 2005, pp. 763–770.
- [12] L. Dong, C. Quirk, and M. Lapata, “Confidence Modeling for Neural Semantic Parsing,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 743–753.
- [13] M. Rikters and M. Fishel, “Confidence through Attention,” in *Machine Translation Summit XVI*, Nagoya, Japan, 2017.
- [14] T. Nakazawa, M. Yaguchi, K. Uchimoto, M. Utiyama, E. Sumita, S. Kurohashi, and H. Isahara, “ASPEC: Asian Scientific Paper Excerpt Corpus,” in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2016)*, N. C. C. Chair), K. Choukri, T. Declerck, M. Grobelnik, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, Eds. Portorož, Slovenia: European Language Resources Association (ELRA), may 2016, pp. 2204–2208.
- [15] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, “fairseq: A Fast, Extensible Toolkit for Sequence Modeling,” in *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

Using WHY-type Question-Answer Pairs to Improve Implicit Causal Relation Recognition

Huibin Ruan, Yu Hong, Yu Sun, Yang Xu, Min Zhang

School of Computer Science and Technology, Soochow University

Suzhou, China

{*hbruan,ysun79,minzhang*}@stu.suda.edu.cn,{*tianxianer,andreaxu41*}@gmail.com

Abstract—Implicit causal relation recognition aims to identify the causal relation between a pair of arguments. It is a challenging task due to the lack of conjunctions and the shortage of labeled data. In order to improve the identification performance, we come up with an approach to expand the training dataset. On the basis of the hypothesis that there inherently exists causal relations in WHY-type Question-Answer (QA) pairs, we utilize WHY-type QA pairs for the training set expansion. In practice, we first collect WHY-type QA pairs from the Knowledge Bases (KBs) of the reading comprehension tasks, and then convert them into narrative argument pairs by Question-Statement Conversion (QSC). In order to alleviate redundancy, we use active learning (AL) to select informative samples from the synthetic argument pairs. The sampled synthetic argument pairs are added to the Penn Discourse Treebank (PDTB), and the expanded PDTB is used to retrain the neural network-based classifiers. Experiments show that our method yields a performance gain of 2.42% F1-score when AL is used, and 1.61% without using.

Keywords-Implicit causal relation recognition; PDTB; Discourse parsing;

I. INTRODUCTION

Discourse relation recognition is a task to determine the relation between a pair of arguments (abbr., Arg). This task is important because it can help for many practical Natural Language Processing (NLP) systems, such as automatic text summarization [1], question answering [2] and conversation [3]. In this task, implicit relation recognition is still a challenge due to the lack of explicit connectives (e.g., “because”). Such overt marker can strongly indicate the relation between two arguments [4].

The distinction between explicit and implicit relations is clearly defined in the Penn Discourse Treebank (PDTB) [5]. As shown in example (1) and (2), the explicit relation in (1) can be easily identified in that the explicit clue “because” straightforwardly indicates the causal relation. While the implicit relation in (2) is difficult to be recognized in a large part due to the lack of a connective. In this case, a deep semantic inference between two arguments is indispensable to determine the implicit relation.

- (1) *They shredded it simply because the Georgia-Pacific bid broke the market's recent gloom.*

(Contingency.Cause.Reason - wsj_0335)

- (2) *This is not the case. (because) Some diaries simply aren't worth snooping in.*

(Contingency.Cause.Result - wsj_0972)

Most recently, neural models are popular in discourse parsing, and the *F1*-score of four-way classification for

the four main relation types (*Expansion*, *Contingency*, *Comparison* and *Temporal*) has been increased to 51.06% [6]. As claimed in previous work [7], such classification models can be strengthened further if there is a larger dataset put into use for training. Therefore, data expansion becomes progressively important in this case.

Discourse-oriented data expansion can be boiled down to two aspects: mining inherently-related arguments and labeling the exact relations. Both will be labor-intensive and time-consuming if they are left to be done by human. To solve this problem, we propose an approach to expand PDTB with less human intervention.

As naturally related sentence pairs, we extract QA pairs from QA KBs for PDTB expansion. In practice, we convert a question into a declarative sentence, and combine it with its answer to form a pair of arguments. As shown in (3), a causally-related argument pair can be generated by a WHY-type QA pair.

- (3) **Question:** *Why is efficiency sometimes lost in phosphor-based LEDs?*

Answer: *heat loss from the Stokes shift*

Question type: WHY

Arg1: *“efficiency is sometimes lost in phosphor-based LEDs”.* **Result**

Arg2: *“heat loss from the Stokes shift”.* **Cause**

Relation type: Contingency.Cause.Reason

In addition, we follow Xu et al. [8] to apply active learning for redundancy elimination. Entropy based informativeness measurement is used to verify whether a pair of newly generated arguments is informative or redundant for learning. The goal is to facilitate the joint use of multiple-source data.

As a preliminary study to evaluate the feasibility of the expansion method mentioned above, we focus on implicit causal relation recognition. So only WHY-type QA pairs are utilized for data expansion [9]. But it is noteworthy that one may take the proposed approach as a baseline to carry out a brand new study of data expansion for multiple-class distant supervision, such as that on temporal and conditional relations other than causal ones, using WHEN-type and HOW-type QA pairs. Experimental results on PDTB show that our approach effectively improves the causal classification performance with in-domain QA pairs, improving the *F1*-score by 2.42%. Besides, active learning is proven effective for redundancy elimination.

II. RELATED WORK

In much previous work of causal relation recognition, metrics and traditional machine learning models are utilized for causality classification, such as distributional similarity [10] and rule-based approaches [11].

Neural network-based methods for implicit relation recognition have been proven effective [7], [12], [13], [6], [14], which require a large number of training data. Due to the data shortage problem in PDTB, discourse-oriented data expansion captures the interest of the research community. Therefore, many corpora are utilized for implicit data expansion, such as the explicit data in PDTB [15], [16], [8], FBIS and HongKong Law [7]. Using the expanded PDTB for classifier training helps strengthen relation recognition. Nevertheless, in such corpora, the semantic relations or homogeneous properties actually have been manually annotated more or less. Thus, the expansion methods fail to be migrated to other kinds of corpora freely and compatibly. The difficulty of obtaining a large training dataset cannot be overcome completely.

III. OUR APPROACH

Our approach mainly consists of three parts: the neural network-based classifier, Question-Statement Conversion (QSC) and Active Learning (AL) mechanism (See Fig. 1). Utilizing the benchmark dataset in PDTB for training, we first obtain a well-trained neural classifier. Based on QSC, the counterfeit causally-related argument pairs are converted from WHY-type QA pairs. In order to eliminate redundancy, the synthetic argument pairs are iteratively sampled by AL mechanism and added to the benchmark dataset. On the basis of the expanded training dataset, we retrain and redevelop the neural classifier [8].



Figure 1: The framework of the proposed approach.

A. Neural Classifier

Basic neural networks akin to Convolutional Neural Network (CNN) and Bi-directional Long Short Term Memory (Bi-LSTM) Network are popular in discourse parsing, which are respectively adopted as classifier in our approach. The reason why we retrain and test them using the expanded datasets is for the purpose of verifying whether our data expansion approach may possibly help to improve other more complicated neural classifier.

B. Question-Statement Conversion

We collect the WHY-type QA pairs and convert them into causally-related argument pairs. QSC is a critical step, which converts a question into a narrative sentence. Such a sentence is then specified as an argument in a role of *result*. Meanwhile the answer accompanied with the question is directly employed as the related argument in a role of *reason*. Therefore, what we necessarily deal with is just to perform interrogation-narration conversion for questions.

We first come up with the rule based method for QSC. In order to conduct QSC more automatically, we also utilize Pointer-generator Network [17] for QSC.

Rule based QSC (RQSC): We come up with a series of trivial rules for QSC, most of which can be boiled down to auxiliary verb (AUX) translocation. See the syntax tree shown in Fig. 2, in which the auxiliary is relocated behind the supreme Noun Phrase (NP). The auxiliaries we consider include modal verbs (e.g., “could”), copular verbs (e.g., “was”) and regular auxiliaries (e.g., “does”).

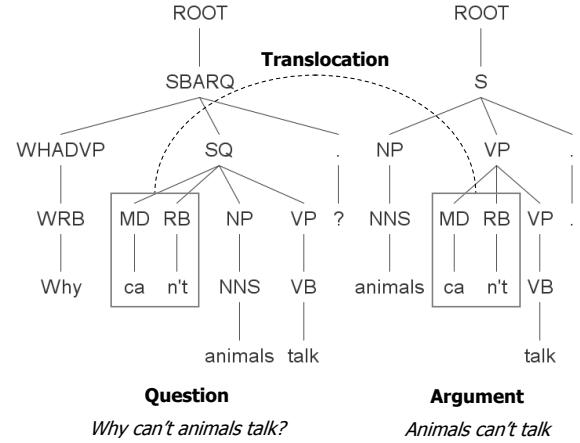


Figure 2: Rule based QSC.

When we conduct QSC for a question, we first parse the sentence using Stanford Parser¹. Second, we traverse the syntax tree in a depth-first order, seeking for auxiliaries by syntactic roles, such as MD (modality), VBZ (auxiliary in third person singular), etc. We lock onto the first auxiliary we met and the affix if have (e.g., a negative adverb “n’t”), and relocate them behind the supreme NP. Finally, we prune the question marker “Why” off the syntax tree. On the basis, we traverse the revised syntax tree in a depth-first order, arrange the words we met in a queue, and perform the dequeue in first-in, first-out order.

In this way, we retroactively produce a declarative sentence, which is used as the argument in the role of “*Result*”. The answer of the question is directly used as the argument in the role of “*Reason*”.

Pointer-generator network based QSC (PQSC): Pointer-generator network [17] is utilized for QSC to generate a narrative sentence from the given question. It is able to automatically decide whether to generate a word from the given word list, or copy a word from the input question. Given a question $Q = \{x_1, x_2, \dots, x_T\}$, each word x_i in it is encoded into an encoder state h_i via a Bi-LSTM. At each decoding step t , the inputs of the decoder are the embedding of the previous word, the previous decoder state s_{t-1} and the context vector h_t^* which is computed by the attention distribution a^t . In practice, a^t

¹<https://nlp.stanford.edu/software/lex-parser.shtml>

is computed as follows [18]:

$$e_i^t = v^T \tanh(W_h h_i + W_s s_t + b_{attn}) \quad (1)$$

$$a^t = \text{softmax}(e^t) \quad (2)$$

where, v , W_h , W_s and b_{attn} are learnable parameters. The distribution a^t is used to compute the context vector h_t^* . And thus, the generation probability $P_\lambda \in [0, 1]$ can be estimated using h_t^* , s_t and x_t .

$$h_t^* = \sum_i a_i^t h_i \quad (3)$$

$$P_\lambda = \sigma(w_{h*}^T h_t^* + w_s^T s_t + w_x^T x_t + b_{ptr}) \quad (4)$$

where, w_{h*} , w_s , w_x and b_{ptr} are learnable parameters, and σ is the sigmoid function. Acting as a soft switch, P_λ can be used in the determination of whether a word needs to be copied from the input question or generated from a given vocabulary [17]. In practice, it is grounded with a copy distribution $P_c(w)$ over a^t as well as the vocabulary distribution $P_v(w)$. This is implemented by a weighted aggregation function as follows:

$$P(w) = P_\lambda P_v(w) + (1 - P_\lambda) P_c(w) \quad (5)$$

where, $P(w)$ coordinates the probability distributions on the vocabulary and all tokens in the input question.

In order to reduce repetitive during narrative sentence generation, we follow See et al. [17] to maintain a coverage vector c^t and use it to optimize the loss function. Such a vector c^t is used to represent the coverage degree of cumulative attentions over the latent information of a token, thus it is computed as follows:

$$c^t = \sum_{t'=0}^{t-1} a^{t'} \quad (6)$$

where, $a^{t'}$ is a revised version of the attention vector a_t in Eqs. (1) and (2) [17].

We follow See et al. [17] to train the pointer-generator network. The loss consists of two parts: the negative log likelihood [19] of generation probability of the target word w_t^* at time step t and a weighted coverage loss:

$$\text{loss} = \frac{1}{T} \sum_{t=0}^T \left(-\log P(w_t^*) + \gamma \sum_i \min(a_i^t, c_i^t) \right) \quad (7)$$

where, γ is used to reweight the coverage loss.

C. Active Learning

To eliminate the noises and redundancy that brought from QSC based data expansion, AL mechanism [8] is employed for informative instances sampling.

Informative samples are those that classified by models with high uncertainty. We use $I_{r_j}(x_i; M)$ to represent the uncertainty level when a sample x_i is recognized as class r_j by a classifier M [8]. Thus x_i is informative only if it significantly increases uncertainty:

$$x^* = \arg \max_{r_j \in R} I_{r_j}(x_i; M) \quad (8)$$

We follow Xu et al. [8] to employ entropy-based uncertainty sampling function to measure the informativeness:

$$\begin{aligned} \text{Inf}(x_i) &= \sum_{r_j \in R} I_{r_j}(x_i; M) \\ &= - \sum_{r_j \in R} P(r_j | x_i) \log P(r_j | x_i) \end{aligned} \quad (9)$$

where, the entropy of the probabilities over causal relation is specified as the informativeness score for instance x_i .

AL mechanism [8] used in our approach mainly includes four steps:

- **Step 1:** Train a learning model over the labeled data in PDTB.
- **Step 2:** Use the well-trained model to classify the unlabeled data.
- **Step 3:** Evaluate the informativeness of the unlabeled data. To reduce the computational complexity, the samples with informativeness scores that higher than a threshold 0.95 [8] are selected. They should have been manually annotated, but the relation of each argument pair is already known.
- **Step 4:** Add the samples to the labeled data and retrain the classifier.

In general, the AL iteration process will not stop until the termination criteria are satisfied, such as the upper bound number of iteration is met [8].

Source code to reproduce the experiments will be made publicly available.

IV. EXPERIMENTS

A. Data Standardization and Expansion

We will first introduce the benchmark dataset of our approach, and then describe the data expansion procedure based on QSC.

Benchmark Dataset: Ponti and Korhonen [20] utilizes *Contingency* argument pairs as positive samples for causal classification, and other types as negative. However, *Contingency* contains 4 subtypes: *Cause*, *Pragmatic Cause*, *Condition* and *Pragmatic Condition*, which are slightly different in relation sense. Especially, there is no causal influence between the arguments of a *Pragmatic Cause* relation [21]. To conduct causal classification in a pure environment, only *Cause* argument pairs are used as positive samples in our work, which is a subset of the Ponti and Korhonen's benchmark dataset (**Sub-P&K's BD** for short).

Table I: Data distribution of Sub-P&K's BD.

Datasets	Train	Dev	Test	all
Positive	3,277	284	272	3,833
Negative	3,277	899	774	4,950

In our experiments, the benchmark dataset Sub-P&K's BD is separated into training, dev and test sets. Table I describes the data distribution after the numbers of positive and negative samples are balanced in the training set [22].

Table II: Causal classification performances of the retrained models. (Train: training set for the pointer-generator network. “-” denotes using RQSC (except for the benchmark dataset Sub-P&K’s BD); Estimate: training set for classifiers. In “Estimate” column, the QA KBs are those which used for Sub-P&K’s BD expansion.)

Datasets		CNN				Bi-LSTM			
Train	Estimate	P(%)	R(%)	F(%)	Gain(%)	P(%)	R(%)	F(%)	Gain(%)
-	Sub-P&K’s BD	36.16	79.77	49.77	-	34.74	81.62	48.74	-
-	+SQuAD	39.08	75.00	51.38	$\uparrow 1.61$	41.50	61.03	49.40	$\uparrow 0.66$
-	+MARCO	35.79	83.82	50.16	$\uparrow 0.39$	40.35	59.93	48.22	$\downarrow 0.52$
-	+NarrativeQA	34.23	79.41	47.84	$\downarrow 1.93$	34.61	76.47	47.65	$\downarrow 1.09$
MARCO	+SQuAD	40.04	70.95	51.19	$\uparrow 1.42$	36.41	65.07	46.70	$\downarrow 2.04$
NarrativeQA	+SQuAD	42.82	61.39	50.45	$\uparrow 0.68$	34.06	73.90	46.64	$\downarrow 2.11$
SQuAD	+MARCO	38.59	68.38	49.34	$\downarrow 0.43$	35.40	65.07	45.85	$\downarrow 2.89$
NarrativeQA	+MARCO	36.89	81.25	50.75	$\uparrow 0.98$	34.58	71.32	46.58	$\downarrow 2.16$
MARCO	+NarrativeQA	34.65	87.13	49.58	$\downarrow 0.19$	36.47	66.91	47.21	$\downarrow 1.53$
SQuAD	+NarrativeQA	36.93	76.84	49.88	$\uparrow 0.11$	38.54	56.25	45.74	$\downarrow 3.00$

Data Expansion: On the basis of QSC, we convert a question into a narrative sentence, which serves as a “*Result*” in the synthetic causal argument pair. Meanwhile, the answer of the question directly serves as a “*Reason*”. With these causally-related argument pairs, we expand the training set in the benchmark dataset. The following three QA KBs are used for expansion:

- **SQuAD** [23]: contains 107,785 QA pairs generated from 536 Wikipedia articles by trained annotators. Based on the answer spans, the questions are generated editorially. We extract 1,028 WHY-type QA pairs for use.
- **MARCO** [24]: consists of 100,000 queries issued to the *Bing* search engine by real users, and the corresponding answers are also free-form human generated text. There are 1,298 WHY-type QA pairs identified and taken for use.
- **NarrativeQA** [25]: includes 46,765 QA pairs generated from 1,572 stories by trained annotators. There are 4,015 WHY-type QA pairs utilized.

Based on the datasets mentioned above, we obtain three sets of synthetic argument pairs by RQSC. Therefore, three expanded versions of the benchmark dataset are generated by adding them to the Sub-P&K’s BD respectively. While for the PQSC, we first train the pointer-generator network on one of the QA KBs. Using the well-trained model, we generate synthetic causal argument pairs for another QA KB. The counterfeit argument pairs are added to the Sub-P&K’s BD to form an expanded version of the benchmark dataset. Running the procedure mentioned above for six times, we obtain six expanded versions of the Sub-P&K’s BD by PQSC.

Totally, we obtain nine expanded versions of the Sub-P&K’s BD by QSC. In the training set of each expanded Sub-P&K’s BD, the number of positive samples is much higher than the negative ones. This leads to unbalance between positive and negative samples, which has been widely recognized as one of the reasons for performance reduction [26]. Therefore, we randomly select negative samples from PDTB to maintain the balance.

B. Experimental Setup

We respectively train the neural classifiers on Sub-P&K’s BD and the nine expanded versions of it (See

Table III: Hyperparameter settings for neural models.

Hyperparameter	CNN	Bi-LSTM	Pointer-generator network
Learning rate	$1e - 3$	0.01	0.1
Batch size	64	30	16
Optimizer	Adam	Adam	Adagrad
Dropout rate	0.2	0.1	-
Filters number	1,024	-	-
Filter size	(2, 2, 2)	-	-

Section IV-A), and adopt Precision (P), Recall (R) and F1-score as the evaluation metrics. The hyperparameter settings are shown in Table III and the detailed model settings are as follows:

CNN [22] is used to recognize the causal relation. We combine word embeddings and POS embeddings to represent the arguments. The 300-dimensional word embeddings are initialized with pre-trained Word2Vec [27] vectors, and the 50-dimensional POS embeddings are initialized by random sampling in [-1,1].

Bi-LSTM [26] is adopted for argument modelling and relation recognition. We follow Guo et al. [26] to set the max sentence length as 50. The 50-dimensional word embeddings are initialized with GloVe [28] vectors. The size of the hidden state for LSTM is set to 100.

Pointer-generator network [17] is adopted for QSC. Adagrad [29] is used to optimize the learnable parameters. During training, the learning rate is set to 0.1 and the initial accumulator value is set as 0.1. The decoding beam size is set to 4 and the coverage loss weight γ is set to 1. We follow See et al. [17] to employ a vocabulary that is extracted from the training sets, and the size is limited to 50k. For both the encoder and decoder, we use 256-dimensional hidden states and 300-dimensional word embeddings initialized with Word2Vec [27] vectors.

C. Experimental Results and Analysis

As shown in Table II, when training on the Sub-P&K’s BD expanded by 1,028 synthetic causal argument pairs generated from SQuAD, the F1-scores are improved by 1.61% and 0.66% on CNN and Bi-LSTM respectively. However, utilizing QA pairs in NarrativeQA for expansion hurts the performance, thought there are 4,015 WHY-type QA pairs used. Undoubtedly, this raises an adaptation

Table IV: Causal classification performances of the retrained models when AL is used.

Datasets		CNN				Bi-LSTM			
Train	Estimate	P(%)	R(%)	F(%)	Gain(%)	P(%)	R(%)	F(%)	Gain(%)
-	Sub-P&K's BD	36.16	79.77	49.77	-	34.74	81.62	48.74	-
-	+SQuAD(AL)	41.52	70.22	52.19	$\uparrow 2.42$	39.35	71.32	50.72	$\uparrow 1.98$
-	+MARCO(AL)	39.50	76.10	52.01	$\uparrow 2.24$	37.55	72.06	49.37	$\uparrow 0.63$
-	+NarrativeQA(AL)	41.50	67.28	51.33	$\uparrow 1.56$	36.69	75.00	49.28	$\uparrow 0.54$
MARCO	+SQuAD(AL)	43.03	65.81	52.03	$\uparrow 2.26$	36.26	80.51	50.00	$\uparrow 1.26$
NarrativeQA	+SQuAD(AL)	40.63	71.69	51.86	$\uparrow 2.09$	36.69	75.00	49.28	$\uparrow 0.54$
SQuAD	+MARCO(AL)	40.39	68.01	50.68	$\uparrow 0.91$	37.17	73.53	49.38	$\uparrow 0.64$
NarrativeQA	+MARCO(AL)	38.71	75.00	51.06	$\uparrow 1.29$	35.43	74.63	48.05	$\downarrow 0.69$
MARCO	+NarrativeQA(AL)	40.08	71.32	51.32	$\uparrow 1.55$	37.88	69.49	49.03	$\uparrow 0.29$
SQuAD	+NarrativeQA(AL)	41.90	64.71	50.87	$\uparrow 1.10$	38.0	69.85	49.29	$\uparrow 0.55$

problem. Our survey illustrates that the inconsistency between domains most probably results in less adaptability.

The experimental results show that the domain inconsistency probably leads to less adaptability. The arguments in PDTB are manually extracted from news stories of the Wall Street Journal. While the QA KBs are established on encyclopedias, webpages, movie scripts and books. Thus the argument pairs in PDTB is rhetorically closer to those in SQuAD. This makes it easier for neural classifiers to learn isomorphic linguistic knowledge from SQuAD samples. By contrast, the rhetorically heterogeneous pragmatics are widely distributed in the NarrativeQA samples. Therefore, data expansion using these samples easily introduces unintelligible latent features for learning rather than referential ones. This can be illustrated with some representative samples akin to those in (4) and (5), where the PDTB argument pair in (4) are rhetorically dissimilar to the QA pair of NarrativeQA in (5). Note that they have the same topic of “*divulgung*”, though they are expressed in different rhetoric.

- (4) **Arg pair in PDTB:** [*One of them, 25-year-old Markus Hess of Hannover, allegedly used the international telecommunications network to break into more than 30 high-security computers in the U.S., searching for secrets.*]_{Arg1} [*He probably didn't penetrate any top-secret files, but the KGB in East Berlin was willing to pay two of his associates.*]_{Arg2}
- (5) **QA pair in NarrativeQA:** [*Why does Harry reveal his secret life?*]_{Question} [*He is given a truth serum.*]_{Answer}

Obviously, the retrained Bi-LSTM performs worse than CNN when the PQSC is used for expansion. Meanwhile, the classifiers perform slightly better when using the RQSC for expansion. These mainly result from that the synthetic argument pairs generated by PQSC include some less-ordered cases (See (6)).

- (6) **Question:** *Why was their use limited?*
Ground Truth: *their use was limited.*
Generated by PQSC: *as use limited was their.*

D. Discussion

In order to eliminate redundancy, we follow Xu et al. [8] to use active learning mechanism to purify the hold-up set for data expansion. As shown in Table IV, after using AL mechanism to select informative samples for expansion,

almost all neural classifiers have been improved further and performs better than only using the Sub-P&K's BD. In particular, the performance gains of CNN and Bi-LSTM have been increased to 2.42% and 1.98% respectively when the Sub-P&K's BD is expanded with SQuAD. The experimental results indicate that the classifiers are more adaptable, when the training set is expanded with samples selected by AL mechanism from various external datasets.

Table V: Causal classification performances of the retrained models when the benchmark dataset is expanded with samples selected by AL from different data sources.

Expansion Source	num (AL)/ALL	P(%)	R(%)	F(%)
Xu et al. [8]	291/2,240	44.79	58.46	50.72
Ours	269/1,028	41.52	70.22	52.19

In order to prove that using in-domain QA pairs to expand the benchmark dataset enhances the neural classifiers, we compare our method with the state-of-the-art AL based expansion approach [8] for implicit causal relation recognition. As show in Table V, our approach performs better than Xu et al. [8]'s when using the same neural classifier, AL algorithm and corpus (i.e., PDTB). While Xu et al. [8] use 2,240 explicit causal argument pairs in PDTB for expansion, and we use 1,028 QA pairs in SQuAD. With less candidates, we select 269 informative samples for expansion, almost the same number as Xu et al. [8] obtain.

V. CONCLUSION

We propose to use WHY-type QA pairs for data expansion, which enhances the performance of implicit causal relation recognition. In addition, we utilize active learning mechanism for redundancy elimination. In the future, we will further explore the usage of HOW-type and WHEN-type QA pairs, so as to help relation *Condition* and *Temporal* recognition.

ACKNOWLEDGMENT

This research work is supported by National Natural Science Foundation of China (Grants No.61672367, No.61672368, No.61751206.). The authors would like to thank the anonymous reviewers for their insightful comments and suggestions. Yu Hong, Professor in Soochow University, is the corresponding author of the paper, whose email address is tianxianer@gmail.com.

REFERENCES

- [1] Y. Yoshida, J. Suzuki, T. Hirao, and M. Nagata, “Dependency-based discourse parser for single-document summarization,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1834–1839.
- [2] P. Jansen, M. Surdeanu, and P. Clark, “Discourse complements lexical semantics for non-factoid answer reranking,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2014, pp. 977–986.
- [3] R. Higashinaka, K. Imamura, T. Meguro, C. Miyazaki, N. Kobayashi, H. Sugiyama, T. Hirano, T. Makino, and Y. Matsuo, “Towards an open-domain conversational system fully based on natural language processing,” in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014, pp. 928–939.
- [4] E. Pitler, A. Louis, and A. Nenkova, “Automatic sense prediction for implicit discourse relations in text,” in *ACL*. Association for Computational Linguistics, 2009, pp. 683–691.
- [5] R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, and B. Webber, “The penn discourse treebank 2.0,” in *LREC*, 2008.
- [6] H. Bai and H. Zhao, “Deep enhanced representation for implicit discourse relation recognition,” in *COLING*, 2018, pp. 571–583.
- [7] C. Wu, Y. Chen, Y. Huang *et al.*, “Bilingually-constrained synthetic data for implicit discourse relation recognition,” in *EMNLP*, 2016, pp. 2306–2312.
- [8] Y. Xu, Y. Hong, H. Ruan, J. Yao, M. Zhang, and G. Zhou, “Using active learning to expand training data for implicit discourse relation recognition,” in *EMNLP*, 2018, pp. 725–731.
- [9] Z. Lin, H. T. Ng, and M.-Y. Kan, “A pdtb-styled end-to-end discourse parser,” *Natural Language Engineering*, vol. 20, no. 2, pp. 151–184, 2014.
- [10] Q. X. Do, Y. S. Chan, and D. Roth, “Minimally supervised event causality identification,” in *EMNLP*. Association for Computational Linguistics, 2011, pp. 294–303.
- [11] C. Grivaz, “Automatic extraction of causal knowledge from natural language texts,” Ph.D. dissertation, University of Geneva, 2012.
- [12] P. Liu, X. Qiu, and X. Huang, “Adversarial multi-task learning for text classification,” *arXiv preprint arXiv:1704.05742*, 2017.
- [13] M. Lan, J. Wang, Y. Wu, Z.-Y. Niu, and H. Wang, “Multi-task attention-based neural networks for implicit discourse relationship representation and identification,” in *EMNLP*, 2017, pp. 1299–1308.
- [14] Z. Dai and R. Huang, “Improving implicit discourse relation classification by modeling inter-dependencies of discourse units in a paragraph,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 141–151.
- [15] D. Marcu and A. Echihabi, “An unsupervised approach to recognizing discourse relations,” in *ACL*. Association for Computational Linguistics, 2002, pp. 368–375.
- [16] A. Rutherford and N. Xue, “Improving the inference of implicit discourse relations via classifying explicit discourse connectives.” in *HLT-NAACL*, 2015, pp. 799–808.
- [17] A. See, P. J. Liu, and C. D. Manning, “Get to the point: Summarization with pointer-generator networks,” *arXiv preprint arXiv:1704.04368*, 2017.
- [18] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.
- [19] F. Qian, C. Gong, K. Sharma, and Y. Liu, “Neural user response generator: Fake news detection with collective user intelligence.” in *IJCAI*, 2018, pp. 3834–3840.
- [20] E. M. Ponti and A. Korhonen, “Event-related features in feedforward neural networks contribute to identifying causal relations in discourse,” *LSDSem 2017*, p. 25, 2017.
- [21] R. Prasad, E. Miltsakaki, N. Dinesh, A. Lee, A. Joshi, L. Robaldo, and B. L. Webber, “The penn discourse treebank 2.0 annotation manual,” 2007.
- [22] L. Qin, Z. Zhang, and H. Zhao, “A stacking gated neural architecture for implicit discourse relation classification,” in *EMNLP*, 2016, pp. 2263–2270.
- [23] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “Squad: 100,000+ questions for machine comprehension of text,” *arXiv preprint arXiv:1606.05250*, 2016.
- [24] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng, “Ms marco: A human generated machine reading comprehension dataset,” *arXiv preprint arXiv:1611.09268*, 2016.
- [25] T. Kočiský, J. Schwarz, P. Blunsom, C. Dyer, K. M. Hermann, G. Melis, and E. Grefenstette, “The narrativeqa reading comprehension challenge,” *arXiv preprint arXiv:1712.07040*, 2017.
- [26] F. Guo, R. He, D. Jin, J. Dang, L. Wang, and X. Li, “Implicit discourse relation recognition using neural tensor network with interactive attention and sparse learning,” in *COLING*, 2018, pp. 547–558.
- [27] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv preprint arXiv:1301.3781*, 2013.
- [28] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *EMNLP*, 2014, pp. 1532–1543.
- [29] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

On the Etymology of *he* ‘river’ in Chinese

Huibin Zhuang

Institute of Linguistics and Applied Linguistics
Henan University
Kaifeng, China
huibinzhuang@aliyun.com

Zhanting Bu

School of Foreign Languages
Qingdao University
Qingdao, China
harrybu@163.com

Abstract—In Chinese *he* 河 ‘river’ can be used as proper names (for the Yellow River), as well as a common word for rivers in North China. Based on linguistic data, ethnological evidence and historical documents, this paper argues against these leading hypotheses and proposes that *he* originated from the Old Yi language, entered Chinese through language contact, and replaced *shui* which was from Old Qiang and later became the only common noun for river in North China.

Key words—etymology; *he* ‘river’; *shui* 水 ‘river’; Old Yi; Chinese

I. INTRODUCTION

When it comes to the etymology of Chinese *he* 河, abundant literature can be found. Of them, there are three major approaches for etymological inquiry into *he*.

The First approach [1] is based on sound laws. Since *he*'s reconstructed form is **g'äl* [2-3] or **g'ar* [4-5], it is believed to be from Altaic (cf. *yool* in Mongolian). Unfortunately, this approach lacks evidence: there is no archaeological evidence that the Yellow River basin was at one time inhabited by Altaic speakers who left the name behind [6].

The second approach [7] is based on semantic laws. It is argued that the phonetic element of *he*, i.e., *ke* 可, has the etymonic meaning of “big”, specifically referring to the Yellow River. However, of these five example words provided, only four applicable — 柯 does not support what he wants to establish [6]. Besides, few of the following examples found in other archaic documents have the meaning “big”, e.g., 翳 ‘sleeve’, 箭 ‘shaft of arrow’, 鯀 ‘globefish’, 环 ‘a kind of jade’, etc. Some of them even have the meaning of “small”, such as 钥 ‘small wok’.

The third one [6] [8] is based on the belief that the Yellow River got its name *he* because of its shape. Tōdō [6] believes that it was bent like a hook, while Takashima [8] argues the part of the Yellow River with which the people living by it were familiar was like an adze handle. This approach, however, is also doubtful. The Yellow River is so big. Even though people could see the river every day, they did not know the shape of it since no one would walk along the river for thousands of miles to find out its shape (Even he did so, he was not able to draw a map and conclude that the river was bent like a hook or an adze handle).

Last but not less important, there is still one problem confronting the second and third approaches, that is, the representative scholars, while providing evidence to show the origin of *he*, did not tell us how *he* replaced *shui* and

later became the only common noun for river in North China. We will start with the argument that *he* came from a certain local aboriginal language, namely, Old Yi.

II. HE AS A COMMON NOUN IN OLD CHINESE

Many scholars believe that *he* was used as a proper name for the Yellow River first, and later developed into a common noun. For example, Wang [9] proposes explicitly that all *he*'s in ancient documents should be understood as the Yellow river or its tributaries. We do not agree with this and we would like to argue that *he* in Old Chinese could be used as a common noun, too. Evidence is shown below:

First, in *Shangshu* 《尚书·禹贡》(*Documents of the Elder*), the expression *jiu he* 九河 “nine rivers” can be found. The names of these nine rivers are presented in “*shi shui*” of *Erya* 《尔雅·释水》. They are *Tuhai* 徒骇, *Taishi* 太史, *Majia* 马颊, *Fufu* 覆瓠, *Husu* 胡苏, *Jian* 简, *Jie* 洁, *Goupan* 钩盘, *Gejin* 爾津. Six of them can still be found today, scattering on the North China plain.

Second, *he* is used in *Shijing* (*The Book of Odes*) as a common noun, as shown below:

委委佗佗，如山如河。(《鄘风·君子偕老》)

... *ru* *shan* *ru* *he*.

... like mountain like river

‘... Stately as a mountain, majestic as a river.’

Here, *he* appears in juxtaposition with *shan* ‘mountain’, thus forming a parallelism. This indicates that *he* here, just like *shan*, is not a proper name referring to the Yellow River, but a common word.

Last, in *Zhuangzi* 《庄子·外物》, the Qiantang River was called *Zhi he* 制河. If the above two clues are not definitive enough, this one surely is. The Qiantang River, being thousands of miles away from the Yellow River, obviously, has nothing to do with it. However, Zhuangcius (Master Zhuang) could address it as *he*. This obviously shows that at least in the Warring States period (475BC-221BC), *he* was not only a proper name for the Yellow River, but also a common noun (at least regionally).

When examining *Shijing*, it is more interesting to notice that only in *Qin feng* (Qin Ballads) can we find that *shui* is used separately as a common noun for river, as shown below:

所谓伊人，在水一方。(《秦风·蒹葭》)

... *zai* *shui* *yi* *fang*

... stay river one region

‘... is somewhere about the river.’

溯游从之，宛在水中央。（《秦风·蒹葭》）

... wan zai shui zhongyang

... as-if in river middle

‘... (She) is right in the midst of the river.’

While in the state of Yong, it is not *shui* but *he* that is used separately as a common noun for river, as shown above. This, clearly, indicates that in the era of *Shijing*, *shui* and *he* were distinctive geographically: in the state of Qin, river was called *shui*, while in the state of Yong, river was called *he*. Given the expression of *jiu he* (nine rivers) in *Shangshu* and *zhi he* from *Zhuangzi*, it is not adventuresome to draw the conclusion that *he* as a common noun was used only in the east of China.

This is supported by the later appearance of *he* in historical documents. For example, the *Historical Atlas of China* shows that two rivers were named *he*, *Kou he* 潼河 and *Huchi he* 壴池河 on the map of the Qin Dynasty [10]. On the map of the West Han Dynasty [10], more rivers were named with *he*, such as *Huchi bie he* 壴池别河 (*Tuhai he* 徒骇河), *Guzhang he* 故障河, *Tunshi he* 屯氏河, *Zhangjia he* 张甲河, *Tunshi bie he* 屯氏别河, *Mingdu he* 鸣犊河, *Duma he* 笃马河, *Gu he* 涡河, etc.. All of them were in East China. Why did *he* exist only in East China? This question cannot be answered unless the history of Chinese is taken into consideration.

III. ETYMOLOGY OF HE

In history (or legend), the Han people had two main origins, Old Qiang and Old Yi. According to *Shi ji* 史记, the Huangdi tribe was a branch of Qiang, living in the northwestern part of China; the Yandi tribe was another branch of Qiang, whose clan's name was *Jiang*. It was the Yandi tribe that first moved eastward and developed primitive agriculture. The Huangdi tribe followed them, and joined the Yandi tribe at last, forming the Yan-Huang

The front battlefield		Endings	The side battlefield		Endings
East	West		East	West	
Yi 夷-----	Xia 夏	Both sides had their victories and defeats. Yi once killed the king of Xia and Xia defeated Yi several times. But Xia never put down Yi.	Yin 殷-----	Guifang 鬼方	The east won.
Shang 商-----	Xia 夏	The east won.			
Yin 殷-----	Zhou 周	The west won.			
Six States-----	Qin 秦	The west won.			
Chen, Xiang---	Qin 秦	The east won.			
Chu 楚-----	Han 汉	The west won.			
			Huaiyi 淮夷----	Zhou 周	Huaiyi failed at last, although it threatened Zhou several times.

This history is actually reflected in languages. Those Qiang-originated tribes and their offsprings brought the Old Qiang language to the east, where it contacted the Old Yi language and initiated language integration. Old Chinese thus appeared.

However, the integration of the two languages has gone through a long process. At the very beginning, it was mainly through the spread of *Yayan* 雅言 which was formed on the basis of the clannish language of Huangdi tribe. “It was not until King Yu 禹 of Xia that the clannish language of Huangdi tribe gradually developed into a common language of Yan-Huang trial alliance, which laid the basis for the so-called *Yayan* in the Spring

tribal alliance 炎黄集团. Its successors, Xia, Zhou, Qin, etc., all had a close relationship with Qiang, too. According to *shiji* (《史记·六国年表》), “Yu 禹 rose from the West Qiang 西羌.” According to the *The Spring and Autumn of Wu and Yue* (《吴越春秋·越王无余外传》), Yu lived at a place call Shiniu 石钮 in West Qiang. In *Annals of Bamboo* (《竹书纪年》), Zhuanxu 颛顼’s family name was Gaoyang 高阳, whose mother was Nüshu 女枢, who gave birth to Zhuanxu at Ruoshui 若水.... Zhuanxu’s son was Gun 鯀, who gave birth to Yu at Shiniu. Even in the Shang and Zhou Dynasties many Qiang tribes joined the Xia people, such as *Zhou* and *Qin*. The Zhou people used to live in the region near Wugong of Shaanxi province today. their ancestress was Jiangyuan 姜原. Jiang descended from Qiang, too.

At that time, in the Yellow River Basin, besides the Yan-Huang tribal alliance, there was the Dong Yi tribal alliance 东夷集团 in the eastern part of China, which was powerful enough to found Shang Kingdom and unified the whole country later. The Yan-Huang tribal alliance, on their way moving eastward, encountered Dong Yi tribal alliance, and wars started. Through millennia of years’ confrontation and incorporation, the two groups unified into one, that is, the early Han people.

Before their final unification, Yan-Huang tribal alliance and its successors confronted with Dong Yi tribal alliance and its successors for millennia. Fu Sinian, therefore, proposes a hypothesis “Yi in the east and Xia in the west”. He argues that during the dynasties of Xia, Shang, Zhou and the following decades, the history of Ancient China has two systems, which are actually reflected by the confrontation between Yi and Xia, as shown in the table below [11]:

and Autumn period and the Han language (i.e., Chinese) ever since the Han Dynasty” [12]. The Old Yi language did not disappear easily but was continuously spoken by the local people in the East until very late. Wang [9] suggests that Old Yi became a dialect in the Spring and Autumn period when it shared most vocabulary with other dialects but owned a separate phonological system.

This confrontation between the two systems found expression in their languages, i.e., Old Yi and Old Qiang, and *he* and *shui* are exactly their representatives of their respective language systems. (Actually there are many more. For example, *ren* 人 and *min* 民, with their proto

forms as **njin* and **r-mi(y)*, and coming from Old Yi and Old Tibeto-Burman, respectively. [13]) We know the Yan-Huang tribal alliance came from the west where the languages are Altaic and Tibeto-Burman dominantly. However, Yi was in the east where it neighbors the Old Yue language(s). Therefore, it is not adventuresome to assume that Old Yi and Old Qiang were two separate languages. How can we demonstrate that *he* and *shui* are from two languages? Their reconstructed forms provide a piece of evidence.

In the literature, the proto-form of *shui* ‘water’ in Old Chinese is assumed to be **qhʷlji?* [14] or **eiwəi* [15], which, obviously, correspond to the river word in Modern Qiang *qhua*: (Mawo Qiang), *tsə* (Mawo Qiang) or *tsuə* (Taoping Qiang) [16]. In addition to *shui*, there are several basic Chinese words coming from Qiang evidently. One is *die* 爹 ‘dad’ from *Guangyun* 《广韵》: “爹，陟邪切，羌人呼父也”.

The east was the territory of Yi, and *he* should come from the Yi language. However, there is no Yi people as an ethnic group in the east nowadays, nor exists the Yi language. How can we tell that there existed a *he* in Old Yi? Actually, one descendant of this language can still be found today, that is, the Yi language spoken by the Yi minority living in the Southwest of China. According to some scholars [17-24], the Yi people in the Southwest today are the descendants of those who lived in the east in the Xia and Shang dynasties (or Dong Yi). Owing to the essential facts that the characters on Dinggong pottery scripts (丁公陶文) are close to the old characters of the Yi minority, and that the name of the Yi people is identical to that of Renfang 人方 (i.e., Dong Yi) in the Shang Dynasty,

It is reasonable for us to draw the conclusion that Dong Yi in Neolithic times and Xia and Shang dynasties are of the same race as the Yi people in the Southwest of China today. (In the classical documents of the Yi people, 夷 was the character used to refer to them themselves. This did not change until the foundation of Neo-china when people believed (incorrectly) that 夷 reflected racial discrimination, and replaced it with 舛. But until today the Yi people still have some objections to this change.) Besides, the fact that the Yi people today calls the Han people “Xia” also supports my argument. [2000]

In fact, Feng’s argument is supported by more evidence from historical documents, archaeology, linguistics (both cognate words and hieroglyphs), culture, etc.

First, according to historical records, as well as previous studies (e.g. [23]), there were two routes for the Old Yi people migrating from east to southwest. One moved first southwestwards from Shandong 山东 province to central China, then northwestwards to Sanwei 三危 and at last southwards to arrive Sichuan 四川 and Yunnan 云南 provinces; the other one moved first northwestwards from Shandong province to Shanxi 山西 province, then westwards to Shaanxi 陕西 province and at last southwards to arrive southeast of Gansu 甘肃 province. Many cultural heritages such as *bagua* 八卦 (the Eight Trigrams) and the ten-month solar calendar of Xia, indicate that ancestors of them used to live in Xia.

Therefore, the time when Modern Yi’s ancestors left East China was no later than the Shang dynasty (most probably, at the end of Xia) [23] [25].

Second, although the Old Yi people disappeared, the language they spoke has left certain traces in the Chinese language and in many historical documents. These traces also point to the Modern Yi Language in the southwest of China. For example, dozens of characters in some ancient Chinese documents have been recognized to be Yi words [26] and many ancient words from historical documents, such as *Shijing* and *Liji* (Rites of the Zhou) can be interpreted by Modern Yi [27-28].

Third, the Ancient hieroglyphs of Yi are helpful in interpreting some ancient hieroglyphs like the well-known Dinggong Pottery Scripts [18-22]. According to Liu [25], via references of hieroglyphs of the Yi language, many puzzling characters from oracle-bone inscriptions can get a better interpretation. For example, 卍 as one of the most controversial characters, has aroused much discussion. Yet until now, no satisfactory proposal has been put forward from the perspective of Modern Chinese and its writing system. However, with the help of hieroglyph of Yi, it can be recognized as 卍, which is pronounced as *khu* [29], meaning “how”. Then those phrases or sentences containing 卍 in oracle-bone inscriptions can get a better interpretation.

From the above discussion, it is not difficult to see that the Yi people in Southeast China today come from the Old Yi people who used to live in East China (Dongyi 东夷). When the Old Yi migrated southwestwards, they took their language and hieroglyphs (The hieroglyphs of Old Yi possibly were the official writing system of the Xia dynasty. This can explain why some scholars had the view that Old Yi was the ruler of Xia [30-31] to Southwest China, and handed them down from generation to generation for thousands of years.

As we have argued in last section, *he* was a common noun used only in East China, therefore, it is very possible that the common noun *he* came from Old Yi (as a substratal word), and that it still has certain connection with river word(s) of Modern Yi in the southwest. Interestingly, proto-form of *he* is similar to *water* of the Yi language. The proto-form of *he* is **ya* [15] which is very similar to *yuu* [32] which means *water* in the Yi language spoken by the Yi people today. Besides, some other scholars would argue that the proto-form of *he* is **GAR* (e.g. **g’ar* [33], **gar* [5], **gal* [1-3], **gaal* [14], which corresponds with the proto-form of *water* in the Proto-Yi Branch (of the Sino-Tibetan Language Family) **?-grakH* [34]. Thus, it is very possible that *he* came from the Old Yi language.

Now we can pose a hypothesis on how *he* came to be the proper name for the Yellow River. As discussed, many proper names for specific rivers in the Shang Dynasty, such as *jiang* 江, *he* 河, *fen* 汾, *wei* 渭, *ji* 济, *jing* 泾, *han* 汉, *luo* 洛, *ru* 汝, *huai* 淮, etc., may come from the languages of the local (aboriginal) tribes. For example, *Jiang* probably came from Old Yue in which it was a common noun, but became a proper name in Old Chinese. The same event might have happened to *he*, which was a common noun in Old Yi, and was borrowed by the Yan-Huang trial alliance to indicate the Yellow River solely. Since *Yayan* was developed on the basis of the

language of the Yan-Huang trial alliance, it kept *he* as a proper name for the Yellow River. But among Old Yi speakers, *he* was still a common name for (big) rivers. This explains the contrast between *shui* and *he* in *Shijing*.

One may ask why in ancient historical documents many rivers in the east were also named *shui*, for example, *Qi shui* 淇水 in *Wei feng* ‘Wei Ballads’ and *Wen shui* 汶水 in *Qi feng* ‘Qi Ballads’ of *Shijing*. How can we explain this? Before we get down to such cases, one concept must be addressed again — *Yayan*. As is mentioned earlier, *Yayan* was formed on the basis of the language system in the western part of China. Originally it was the language of Huangdi tribe. It survived the Xia and Shang dynasties, keeping in contact with many tribal languages, and finally came into being in the period of the West Zhou Dynasty. “The pronunciation in the capital was the most standard; therefore it is named *ya* (elegant). The pronunciations in the states were not that standard and thus named *feng* (folk)” [35]. In the Spring and Autumn period, *Yayan* already enjoyed a great popularity among nobilities. For example, according to *Lunyu* (*Confucian Analects*), Confucius used *Yayan* in his reading and formal activities (子所雅言). That is to say, *Yayan* at that time had spread as far as the State Lu. Since *Yayan* was formed on the basis of the language system in the west (Old Qiang), and established finally in the west (the capital of Zhou was Haojing, which is near to Xi'an of Shaanxi province today), naturally it would take *shui* as a common noun for river. *He*, which came from the Yi language, was not accepted by *Yayan*.

Now it is time to account for the appearance of *Qi shui* and *Wen shui*. *Shijing* was compiled in the Spring and Autumn period, when *Yayan* was the official language used by the ruling class for formal activities. It is not difficult to imagine that they were sure to use *Yayan* when they named rivers.

The popularity of *Yayan* among the upper class, however, did not mean that the language system in the east had disappeared. In *Liji*, it is recorded that “The people in five directions cannot understand each other and may have different desires (五方之民言语不通，嗜欲不同).” That is to say, although *Yayan* enjoyed popularity among the upper class, the folk, however, usually used their own languages/dialects. This is why *he* was still used then. *Shijing* exactly recorded that even the noble could not avoid speaking a certain local language or dialect. “Since Confucius grew up in the state of Lu, he could not avoid speaking the Lu language. Only when he chants *Shijing*, reads *Shangshu* and cites *Liji*, he must make his pronunciation standard” [35].

IV. THE RISE OF HE

When Old Qiang encountered Old Yi, their lexical items would inevitably compete to be the dominant ones. This is exactly demonstrated by the wars between *he* and *shui*. Below we will look into the historical documents to have a general overview of this war. Due to the fact that there is no faithful historical account before the Zhou Dynasty, we will start from Zhou.

A. *Shui advancing while he retreating*

As mentioned above, *shui* came from the language system in the western part of China (Some readers may point out that there was another river word, *chuan* 川, in the West China. But *chuan* actually was a variety of *shui*. [36]), and *he* came from the language system in the east. Ever since Xia and Shang, *Yayan* was the official language that was popular among the upper class. In the West Zhou Dynasty, especially, *shui* was overwhelming. From the *Historical Atlas of China* we can see that in the West Zhou Dynasty, the Spring and Autumn period, and the Warring period, all the rivers were named with *shui*. This does not necessarily mean that *he* was extinct. It is imaginable that *he* at that time must have certain kind of influence among the common people. Otherwise in the Warring period, Zhuangcius would not call Qiantang River as *Zhi he*.

B. *He replacing shui*

How could *he* come from behind and replace *shui* as the official form? The whole process can be divided into several phases:

First, *shui*'s losing its dominant status. Ever since 770BC, the strong states started to contend for hegemony, splitting the whole country into many small kingdoms, and creating many political centers. The change of the political center resulted in *Yayan*'s loss of the soil where it was rooted. The rising of the states means that it should be possible for some local words (e.g. *he*) to take this chance and sneak into *Yayan*, thus making local *Yayan* develop. This happened not only within the spoken system, but also in the written system [37].

Second, reanalysis of *Huang he* (Yellow River). In the Spring and Autumn period, *he* started to be used as a common word. It is interesting to notice that at the same time the Yellow River was also reanalyzed. For example, the lower reaches of the Yellow River (the part near Luoyang 洛阳 and Gongxian 巩县) was named *Nan he* 南河(the South River) [10]. In the Warring States period, the reach of the Yellow River in Inner Mongolia today was called *Bei he* 北河 (the North River), and the reach between *Bei he* and *Nan he* was called *Xi he* 西河 (literally, the West River) [10]. The division and reanalysis of the Yellow River in fact meant that the proper name *he* was not that proper any more. This inevitably helped the spread of *he* as a common word. It happened that at the very time, the number of people who used *shui* as a common noun dropped dramatically.

Third, rising of *he*. The rulers of Qin took pains and efforts to build up a big empire, but the warring in years resulted in a dramatic decrease in the number of *Yayan*-speakers. One is that many local people in the former Qin state who spoke *Yayan* died in the war; besides, the noble in the other states were killed or forced to flee from war (and became common people soon and picked up certain local language or dialect). Those lucky survivors, however, were forced to leave their home state in the east to reside in the capital of Qin. “(Emperor Qinshihuang) moves one hundred and twenty thousand powerful and wealthy people to Xianyang (徙天下豪富于咸阳十二万户)” (*Shiji-Qinshihuang Benji*). The fact that the population of *Yayan*-speakers reduced sharply

means that the dominance of *Yayan* was further going down. Taking advantage of this chance, many words of common people entered the written language and gradually became dominant. *He* was one of them.

Fourth, spreading of *he*. As mentioned above, in the Qin Dynasty, some river names with *he* like *Kou he* 滄河, *Huchi he* 虞池河, etc., appeared in the north on the map while in the West Han Dynasty, even more *he*-named rivers appeared, as shown above. From then on, the bounds of *he* kept enlarging. It even spread to the southwest. For example, in the East Han Dynasty, there was *Dieyu he* 楠榆河 [10]; in the Tang Dynasty, even Yarlung Zangbu River was named *Zang he* 藏河 [10].

Fifth, the establishing of *he*'s status. As more and more *he*-named rivers appeared and their bounds widened, Yellow River gave up its *shui* name and picked up a *he* name, *Huang he* 黄河, in the South and North Dynasty [10]. This event smashed the ‘firm front’ of *shui*. In the following hundreds of years, almost all the rivers in the north changed their names into *he*. Today, on the map of North China, few rivers with *shui* name are left.

V. CONCLUSION

In this paper, we have provided pieces of evidence from linguistics, ethnology, history, and geography to put forward the hypothesis on the etymology of river words in Chinese that *he* came from the Old Yi language, while *shui*, a word of the Old Qiang language, was dominant at the beginning of Early Old Chinese, but was gradually replaced in the long run of the history of Chinese. Much evidence from linguistics, anthropology and history shows that *he* came from Old Yi, which was incorporated into Proto-Chinese, and replaced *shui* which was a river word from Old Qiang. In this progress, the separatist regimes in the Spring and Autumn period and the Warring States period and the following unification of the Qin empire played an important role.

ACKNOWLEDGMENT

This research is supported by National Social Science Foundation of China (No. 19BYY001).

REFERENCES

- [1] J. Norman and T.-L. Mei, “The Austroasiatics in Ancient South China: Some Lexical Evidence,” *Monumenta Serica*, vol. 32, 1976, pp. 274-301, DOI: 10.2307/40726203
- [2] T. Rai [賴惟勤], “Joko Kango no koon inbi ni trs i te [上古汉语の喉音韵尾について],” *Ochanomizu joshi daigaku jinbun kagaku kiyo* [お茶の水女子大学人文科学纪要], vol. 3, 1953, pp. 51-64.
- [3] E. G. Pulleyblank, “The Consonantal System of Old Chinese,” *Asia Major*, vol. 9, 1962, pp. 58-144, 206-265.
- [4] B. Karlgren, “Grammata Serica Recensa,” *Bulletin of the Museum of Far Eastern Antiquities*, vol. 29, (1957) pp. 1-332.
- [5] F. K. Li [李方桂], “Shanggu yin yanjiu [上古音研究],” *Tsing Hua Xuebao* [清华学报], New vol. 9, 1971, pp. 1-61.
- [6] K. Takashima, “Etymology and Palaeogeography of the Yellow River hé 河,” *Journal of Chinese Linguistics*, vol. 40, 2012, pp. 269-306.
- [7] H. Zhang, “Chinese Etyma for River,” *Journal of Chinese Linguistics*, vol. 26, 1998, pp. 1-43.
- [8] A. Tōdō [藤堂明保], *Kanji gogen jiten* [汉字语源辞典]. Tokyo: Gakutōsha [学灯社], 1965.
- [9] L. Wang [王力], “Jianghe shiyi de tongxin [江河释义的通信],” in *Tantan Xuexi Gudai Hanyu* [谈谈学习古代汉语]. Jinan: Shandong Jiaoyu Chubanshe, 1984, pp. 210-222.
- [10] C. H. Tan [谭其骧], *Zhongguo Yuyan Lishi Ditu Ji* [中国历史地图集]. Beijing: Zhongguo Ditu Chubanshe, 1982b.
- [11] S. Fu [傅斯年], “Yi-Xia dongxi shuo [夷夏东西说],” in *Zhongyang Yanjiuyuan Lishi Yuyan Yanjiusuo Jikan* [中央研究院历史语言研究所集刊] (Waibian diyizhong [外编第一种]): *Qingzhu Tsai Yuan-pei Xiansheng Liushiwusui Lunwenji* [庆祝蔡元培先生六十五岁论文集]. Beijing: Guoli Zhongyang Yanjiuyuan, 1935, pp. 1093-1134.
- [12] Z. Pu [濮之珍], *Zhongguo Yuyanxue Shi* [中国语言学史]. Shanghai: Shanghai Guji Chubanshe, 1987.
- [13] S. Zhang [张树铮], “Shilun Hanyu ‘ren’ de laiyun [试论汉语‘人’的来源——兼谈汉族和汉语的来源],” *Shandong Shida Xuebao* [山东师大学报], issue 1, 1993a, pp. 93-96.
- [14] S. Zhengzhang [郑张尚芳], *Shangu Yinxi* [上古音系]. Shanghai: Shanghai Jiaoyu Chubanshe, 2003.
- [15] X. Guo [郭锡良], *Hanzi Guyin Shouce* [汉字古音手册]. Beijing: Beijing Daxue Chubanshe, 1986.
- [16] H. Sun [孙宏开], *Qiangyu Jianzhi* [羌语简志]. Beijing: Minzu Chubashe, 1981.
- [17] L. Duan [段连勤], “Guanyu Yizu de xiqian he Qinyin de qiyuandi, zushu wenti [关于夷族的西迁和秦嬴的起源地、族属问题].” *Renwen Zazhi* [人文杂志], special issue “Xianqinshi Lunwenji” [先秦史论文集], 1982.
- [18] S. Feng [冯时], “Longshan shidai tao wenhua yu gu yiwen [龙山时代陶文化与古彝文],” *Guangming Ribao* [光明日报], 6 June, 1993.
- [19] S. Feng [冯时], “Shandong Dinggong longshan shidai wenzi jiedu [山东丁公龙山时代文字解读],” *Kaogu* [考古], issue 1, 1994, pp. 37-54.
- [20] S. Feng [冯时], “Wenzi qiyuan yu Yi-Xia dongxi [文字起源与夷夏东西],” *Zhongguo Shehui Kexueyuan Gudai Wenming Yanjiu Zhongxin Tongxun* [中国社会科学院古代文明研究中心通讯], issue 3, 2002.
- [21] S. Feng [冯时], “Shilun zhongguo wenzi de qiyuan [试论中国文字的起源],” *Sichuan Wenwu* [四川文物], issue 3, 2008, pp. 46-49.
- [22] R. Bian [卞仁], “Guanyu ‘Dinggong Taowen’ de Taolun [关于“丁公陶文”的讨论],” *Kaogu* [考古], issue 9, 1994, pp. 825-831.
- [23] P. Chen [陈平], “Cong ‘Dinggong Taowen’ tan gu Dongyizu de xiqian [从“丁公陶文”谈古东夷族的西迁],” *Zhongguo Shi Yanjiu* [中国史研究], issue 1, 1998, pp. 3-13.
- [24] M. Yi [易谋远], *Yizu Shiyao* [彝族史要]. Beijing: Shehui Kexue Wenxian Chubanshe, 2000.
- [25] Z. Liu [刘志一], *Kaogu fajue chutu guyiwen zongshu* [考古发掘出土古夷(彝)文综述]. In: *Zhongguo Minzu Guwenzi Yanjiuhui Diqu ci Xueshu Yantaohui Lunwenji* [中国民族古文字研究会第七次学术研讨会论文集]. Beijing, 13-15 Octobor, 2004. *Zhongguo Minzu Guwenzi Yanjiuhui*.
- [26] Ap yup tiex ry [阿余铁日], *Gudai Yi-Han gongtongci* [古代彝汉共词]. *Xinan Minzu Xueyuan Xuebao* (Shehui Kexue ban) (6), 1998, pp. 52-53.
- [27] S. Huang [黄树先], *Guxun pangzheng* [古训旁证]. *Yuyan Yanjiu* [语言研究], issue 2, 1990, pp. 119-121.
- [28] S. Chen [陈世军], *Shijing yuyan yu Yiyu bijiao suyuan* [《诗经》语言与彝语比较溯源]. *Guizhou Minzu Xueyuan Xuebao* (Zhexue Shehui Kexue ban) [贵州民族学院学报(哲学社会科学版)], issue 3, 2004, pp. 108-110.
- [29] Guizhou Sheng Bijie Diqu Minwei Yiwen Fanyi Zu [贵州省毕节地区民委彝文翻译组] *Yiwen Zidian* [彝文字典]. Ms., 1978.
- [30] M. Zhong [钟鸣], “Sanxingdui kedouwen kaoju: Yizu shi xiadai de tongzhizhe [三星堆蝌蚪文考据:彝族是夏代的统治者],” *Nanfang Zhoumo* [南方周末], 10 August, 2009.
- [31] Z. Liu [刘志一], *Lun minzu wenzi de qiyuan, fanzhan yu xiaowang* [论民族文字的起源、发展与消亡]. *Zhongyao Minzu Xueyuan Xuebao* [中央民族学院学报], issue 1, 1988, pp. 62-67.
- [32] S. Chen, S. Bian and X. Li [陈士林、边仕明、李秀清], *Yiyu Jianzhi* [彝语简志]. Beijing: Minzu Chubanshe, 1985.

- [33] B. Karlgren, Word families in Chinese. Bulletin of the Museum of Far Eastern Antiquities, vol. 5, 1934, pp.9-120.
- [34] D. Bradley [布莱雷德], Yiyuzhi Yuanliu [彝语支源流]. Chengdu: Sichuan Renmin Chubanshe, 1991.
- [35] T. Liu [刘台拱], Lunyu Pianzhi [论语骈枝]. Shanghai: Shanghai Guji Chubanshe [上海古籍出版社], 1996.
- [36] S. Zhang [张树铮], “Heliu Mingcheng ‘chuan’ he ‘shui’ de dili fenbu jiqi yuyan beijing [河流名称“川”和“水”的地理分布及其语言背景].” Shandong Daxue Xuebao [山东大学学报], issue 2, 1993b, pp.68-72.
- [37] G. Wang [王国维], Handai Guwen Kao [汉代古文考]. Taipei: Yee Wen Publishing Company [艺文印书馆], 1964-1969.

Using Mention Segmentation to Improve Event Detection with Multi-head Attention

Jiali Chen, Yu Hong, Jingli Zhang, and Jianmin Yao

School of Computer Science and Technology, Soochow University

Suzhou, China

{ivycjl94, tianxianer, jlzhang05}@gmail.com, jyao@suda.edu.cn

Abstract—Sentence-level event detection (ED) is a task of detecting words that describe specific types of events, including the subtasks of trigger word identification and event type classification. Previous work straightforwardly inputs a sentence into neural classification models and analyzes deep semantics of words in the sentence one by one. Relying on the semantics, probabilities of event classes can be predicted for each word, including the carefully defined ACE event classes and a “N/A” class(i.e., non-trigger word). The models achieve remarkable successes nowadays. However, our findings show that a natural sentence may posses more than one trigger word and thus entail different types of events. In particular, the closely related information of each event only lies in a unique sentence segment but has nothing to do with other segments. In order to reduce negative influences from noises in other segments, we propose to perform semantics learning for event detection only in the scope of segment instead of the whole sentence. Accordingly, we develop a novel ED method which integrates sentence segmentation into the neural event classification architecture. Bidirectional Long Short-Term Memory (Bi-LSTM) with multi-head attention is used as the classification model. Sentence segmentation is boiled down to a sequence labeling problem, where BERT is used. We combine embeddings, and use them as the input of the neural classification model. The experimental results show that the performance of our method reaches 76.8% and 74.2% F_1 -scores for trigger identification and event type classification, which outperforms the state-of-the-art.

Keywords—Event Detection; Mention Segmentation; Multi-head Attention;

I. INTRODUCTION

Event detection (ED) is a crucial subtask of event extraction, which aims to identify triggers in a target sentence and assigns an event type for each trigger. For example, considering the sentence 1) on the ACE 2005¹, where “pay” is the trigger, and an ED system is expected to predict the “Fine” event triggered by “pay”.

- 1) *He also brought a check from Campbell to pay the fines and fees.*

Recently, neural networks have been widely used in ED [1], which can capture semantic information. However, they can't ignore the interference of redundant words. If the model can concentrate more on event-related content, the ED performance will improve. For example, in sentence 2), “discussions” triggers the “Meet” event and the underlined part is an event mention. The content before the underline in the sentence is a supplement to the

underlined content, rather than the principal content of the event. Thus, the content before the underline is redundant information for ED. If we consider more about the event-related information such as the underlined content, we would have more confidence in predicting the “Meet” event successfully. Manual annotation of event mention is time- and resource-expensive. Therefore, we propose a method for extracting event-related information. We adopt the idea of pointer networks [2] to get two pointers (indexes). One pointer points to the beginning of the event mention in the sentence, and the other pointer points to the end of the event mention. Then, the chunk divided from the sentence by two pointers is the used event mention in this paper. Finally, we use mention segmentation to improve ED performance.

- 2) *“It was useful to get it all out on the table and see where we go from here,” he said, referring to the April 23 to 25 discussions in Beijing.*

In addition, we note that the contribution degrees of all words in the sentence are different. For example, in sentence 1), “pay” provides more crucial clues than other words for ED and should be paid more attention. Fortunately, we find that multi-head attention mechanism [3] can compute the weights of different words in the sentence. And, it can help the Bidirectional Long Short-Term Memory (Bi-LSTM) [4] network concentrate on the important words in the sentence. Therefore, we add multi-head attention mechanism to the Bi-LSTM network with event mention for the ED task.

In summary, the contributions of this paper are as follows:

- We find that event mention is effective, and we also analyze the effects of event mention and use a model to predict event mention.
- We propose a novel method for improving ED that can exploit event mention via Bi-LSTM based on the multi-head attention. Additionally, we analyze the effects of multi-head attention.
- The experimental results demonstrate that our proposed method significantly outperforms the current state-of-the-art performance on the widely used ACE 2005 dataset.

II. RELATED WORK

Event detection which aims to identify triggers and classify event types has achieved considerable results.

¹<https://catalog.ldc.upenn.edu/Ldc2006t06>

Gupta et al. [5] propose cross-event inference to alleviate the problem of unknown time argument. Grishman and Ralph [6] achieve sentence-level event type classification with document-level information. Hong et al. [7] use cross-entity inference to achieve sentence-level trigger. Li et al. [8] incorporate global and local features via structured prediction. Liu et al. [9] propose a global inference method to implement event detection. The above methods are all feature-based methods.

At present, most studies have applied neural networks. Nguyen et al. [10] use Convolutional Neural Network (CNN) for event detection. Chen et al. [11] propose a dynamic multi-pooling convolutional neural network. Nguyen et al. [12] use a Bidirectional Recurrent Neural Network (Bi-RNN) to extract event triggers and arguments jointly. Duan et al. [1] exploit document-level information via Recurrent Neural Network (RNN). Feng et al. [13] propose a hybrid neural network. Chen et al. [14] propose a hierarchical and bias tagging networks with gated multi-level attention mechanisms. Zhao et al. [15] propose a novel document embedding enhanced Bi-RNN method. Hong et al. [16] propose a self-regulated learning method by exploiting a generative adversarial network to generate spurious features.

BERT [17] has emerged as an increasingly popular pre-trained language representation model for the tasks of natural language processing. For example, Xu et al. [18] use BERT post-training for review reading comprehension and aspect-based sentiment analysis. We follow the work to fine-tune BERT for extracting event mention. In addition, guided by the transformers [3] of the BERT encoder model, we note that the multi-head attention mechanism can capture important words in the sentence.

III. TASK DESCRIPTION

This paper focuses on the ED task defined in ACE evaluation [19]. We will briefly introduce related terms for the ED task in this section.

- **Entity:** An object or collection of objects in one of the semantic categories, such as human, object, location, etc;
- **Entity Mention:** A phrase for a specific type of entity;
- **Event Trigger:** A main word that expresses the occurrence of an event (often a noun or phrase) and consists of a single word or phrase;
- **Event Argument:** The participant of an event, which is the important part of the event, involving entities, time, and values;
- **Event Mention:** A phrase or sentence which includes trigger words and event arguments.

An event involves a specific type of event and one or more participants associated with the event. The ED task aims to identify triggers and classify event types. The ACE 2005 evaluation defines 8 event types and 33 event subtypes. Following Li et al. [8], we only focus on 33 event subtypes.

IV. METHODOLOGY

Following Nguyen and Grishman [12], event detection is considered as a multi-class classification problem. The aim of the ED task is to predict whether the token in a given sentence can trigger a specific type of event. In this section, we will describe the event segmentation and the multi-head attention-based Bi-LSTM.

A. Mention Segmentation

In this paper, we propose a method that extracts event mention to capture the most related words about events in the sentence and reduce the interference of redundant words.

With the development of deep learning, we adopt the end-to-end model to extract event mention by fine-tuning the pre-trained BERT. BERT [17] is a pre-trained language representation model which can break down records of multiple tasks with one additional fully-connected layer. Thus, we select fine-tuning the pre-trained BERT model² for extracting event mention. Given a sentence with n words, and we formulate the input of BERT as $S = \{[CLS], s_1, s_2, \dots, s_i, \dots, s_n, [SEP]\}$, where $[CLS]$ is a dummy token and $[SEP]$ marks the end of the sentence. Firstly, we obtain the hidden representation after BERT as $h_s = BERT(S)$. Then, we add two separate linear layers to the hidden representation followed by a softmax function to get two pointers (indexes). The two pointers can indicate the start and end position of the target chunk. The specific calculation formula is as follows:

$$L_{start,end} = \text{softmax}(W_{1,2}h_s + b_{1,2}) \quad (1)$$

where $W_{1,2}$ are two separate weight matrices and $b_{1,2}$ are two separate bias terms. We utilize the averaged cross-entropy on the two pointers as to the loss function and minimize it via Adam optimizer:

$$L_{BERT} = -\frac{\sum T(\text{start})\log L_{\text{start}} + \sum T(\text{end})\log L_{\text{end}}}{2} \quad (2)$$

where $T(\text{start})$ and $T(\text{end})$ are one-hot vectors representing the ground-truth pointers. Finally, the chunk divided from the sentence by two predicted pointers is regarded as the predicted event mention.

B. Multi-head attention-based Bi-LSTM

Figure 1 demonstrates the architecture of multi-head attention-based Bi-LSTM for ED. The model consists of the following components: (i) Embedding Layer; (ii) Multi-head Attention Mechanism; (iii) Bi-LSTM; (iv) Output; (v) Training.

1) *Embedding Layer:* Following Liu et al. [20], we take each token in the target sentence as the input of the network and transform them into a real-valued vector by looking up embedding tables.

- **Word Embedding** We utilize the Skip-gram model [21] to learn word embedding on the NYT corpus³. We present it as: $W = \{w_1, w_2, \dots, w_i, \dots, w_n\}$,

²<https://github.com/google-research/bert>

³<https://catalog.ldc.upenn.edu/LDC2008T19>

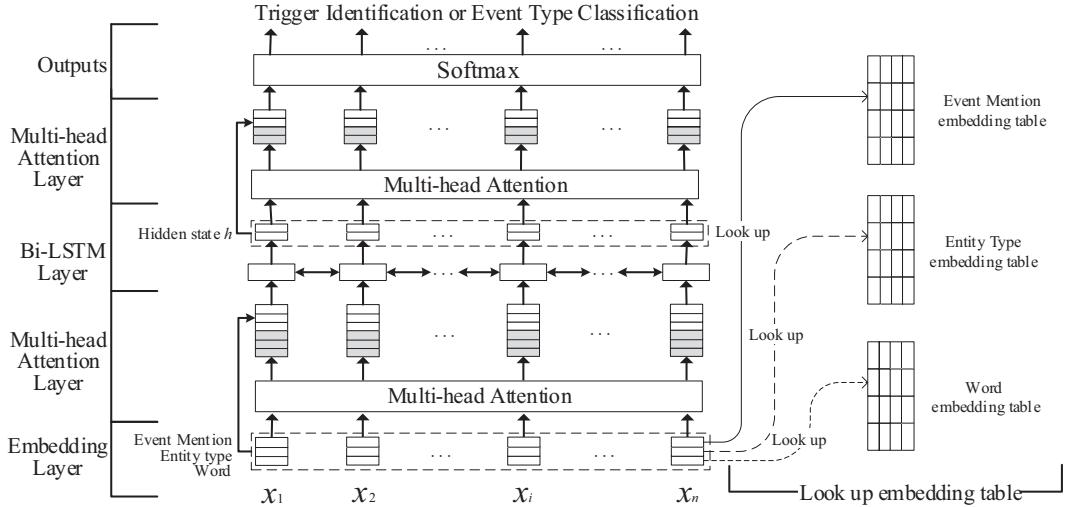


Figure 1. The architecture overview of multi-head attention-based Bi-LSTM with event mention

where w represents the hidden semantic properties of a token which is obtained by looking up a pre-trained word embedding table.

- **Entity Type Embedding** The representation of entity type in this sentence is: $E = \{0, \dots, e_i, \dots, e_j, 0, \dots\}$, where “0” represents that the current word is not an entity, and e_i or e_j indicates that the i th or j th word is a specific type of entity. Following Hong et al. [16], we look up entity type embedding table which is randomly initialized embedding vector to get the entity type embedding W_e .
- **Event Mention Embedding** We present the event mention label as: $M = \{0, \dots, 1, \dots, 1, 0, \dots\}$, where “1” indicates event mention in a sentence and “0” indicates other words of the sentence. Similarly, we encode the event mention as a fixed-dimensional real-valued vector W_{men} by looking up randomly initialized event mention embedding table.

Finally, we concatenate the three embeddings into the final representation denoted as $X = \{W, W_e, W_{men}\}$, where $X \in \mathbb{R}^{n \times m}$, n is the length of the sentence and m is the dimensionality of each token.

2) *Multi-head Attention Mechanism*: For paying more attention to keywords in a given sentence and reducing the interference of meaningless words, we exploit the multi-head attention mechanism [3] before and after Bi-LSTM respectively. In this section, we will introduce the details of multi-head attention.

The structure of multi-head attention is shown in Figure 2, the three inputs of multi-head attention are “Query”, “Key”, “Value”, which are denoted as Q , K , V . Since we adopt the self-attention in one sentence, we set “ $Q = K = V = X$ ”. Firstly, we add three separate linear layers to the Q , K , V , and denote the outputs as \tilde{Q} , \tilde{K} , $\tilde{V} \in \mathbb{R}^{n \times m}$:

$$\begin{bmatrix} \tilde{Q} \\ \tilde{K} \\ \tilde{V} \end{bmatrix} = W \begin{bmatrix} Q \\ K \\ V \end{bmatrix} + b \quad (3)$$

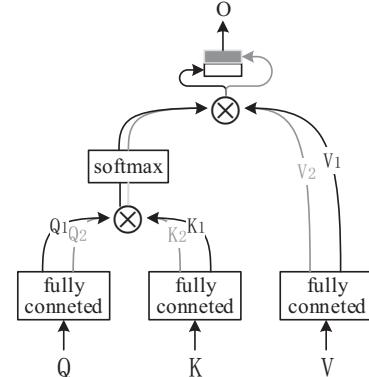


Figure 2. Multi-head Attention Mechanism

Then, we divide the last dimensionality of \tilde{Q} , \tilde{K} , \tilde{V} into two segments $\bar{Q}_{1,2}$, $\bar{K}_{1,2}$, $\bar{V}_{1,2} \in \mathbb{R}^{n \times \frac{m}{2}}$. We compute the dot products of the candidate token $\bar{Q}_{1,2}$ with other tokens $\bar{K}_{1,2}$, divide each by $\sqrt{d_k}$ (d_k is the dimensionality of \tilde{K}) and apply a softmax function to obtain the weights on $\bar{V}_{1,2}$:

$$\bar{O}_{1,2} = \text{softmax}\left(\frac{\bar{Q}_{1,2} \bar{K}_{1,2}^T}{\sqrt{d_k}}\right) \bar{V}_{1,2} \quad (4)$$

Finally, we concatenate the outputs $\bar{O}_{1,2}$ into a new representation $O = [\bar{O}_1, \bar{O}_2]$ ($O \in \mathbb{R}^{n \times m}$), and concatenate O with the input representation X into the final output $T = [O, X]$ ($T \in \mathbb{R}^{n \times 2m}$). T is as the input of Bi-LSTM.

3) *Bi-LSTM*: Bi-LSTM [4] is a network that combines the forward Long Short-Term Memory (LSTM) [22] and the backward LSTM. It can capture bidirectional semantic dependencies, and combine context information effectively.

At every time step t , we set an input gate i_t , forget gate f_t , output gate o_t , and cell memory unit c_t . The LSTM unit obtains a distributed representation according to the current input T_t , the previous hidden layer state h_{t-1} , and the previous cell state c_{t-1} . The detailed operations of

LSTM are as follows:

$$\begin{bmatrix} i_t \\ f_t \\ o_t \\ \tilde{c}_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} \left(W \begin{bmatrix} T_t \\ h_{t-1} \end{bmatrix} + b \right) \quad (5)$$

$$c_t = i_t \odot \tilde{c}_t + f_t \odot c_{t-1}, h_t = o_t \odot \tanh(c_t) \quad (6)$$

where $W \in \mathbb{R}^{2m \times d}$ is the weight matrix, $b \in \mathbb{R}^{n \times d}$ is the bias term, d is the size of the hidden units, σ refers to the sigmoid function, and \odot denotes element-wise multiplication. Finally, the forward hidden state \overrightarrow{h}_t and the backward hidden state \overleftarrow{h}_t are concatenated into a single vector $h_t = [\overrightarrow{h}_t, \overleftarrow{h}_t]$ ($h_t \in \mathbb{R}^{2d}$). Then, we implement the multi-head attention on h_t and get a new sequence of vectors A . Finally, we concatenate H and A into a matrix $\tilde{H} = [H, A]$ ($\tilde{H} \in \mathbb{R}^{n \times 4d}$) as the input of a fully-connected layer.

4) *Output*: When identifying triggers, we use each token in the target sentence as the candidate trigger and calculate scores of event type for each token. Following Hong et al. [16], we use a fully-connected layer followed by a softmax function to obtain the predicted conditional probability $P(y|x, \theta)$ of different types:

$$P(y|x, \theta) = \text{softmax}(W\tilde{H} + b) \quad (7)$$

where $W \in \mathbb{R}^{4d \times t}$ is the weight matrix, $b \in \mathbb{R}^{n \times t}$ is the bias term, t represents the number of event types including one non-trigger type, and θ represents all parameters of the model.

5) *Training*: We minimize the objective function to reduce the cross-entropy loss of our model:

$$L(\Theta) = - \sum_{i=1}^l \sum_{j=1}^n \hat{y}_{ij} \log P(y_j|x_i, \theta) \quad (8)$$

where \hat{y}_{ij} represents that the i th token x_i triggers the j th real event type, $P(y_j|x_i, \theta)$ represents scores of the j th predicted event type for the i th token x_i . We minimize the log-likelihood $L(\Theta)$ through Stochastic Gradient Descent (SGD) [10] optimizer to compute all the parameters θ .

V. EXPERIMENTS

A. Experiment Settings

1) *Dataset and Evaluation Metrics*: Following the previous works [8], [11], [12], [23], we split ACE 2005 dataset into 40 documents for the test set, 30 documents for the development set, and 529 documents for the training set. Additionally, we utilize Precision (P), Recall (R), and F_1 -score (F_1) as the evaluation metrics.

2) *Hyper-parameters of Mention Segmentation*: We adopt $BERT_{Large}$ (uncased)⁴ for fine-tuning the pre-trained BERT model. The fixed length of sentences is set to 128, the batch size is set to 1, and the learning rate is set to 3e-5.

⁴https://storage.googleapis.com/bert_models/2018_10_18/uncased_L-24_H-1024_A-16.zip

3) *Hyper-parameters of ED model*: We use 300 dimensions for word embedding, 50 dimensions for entity type embedding, and 6 dimensions for event mention embedding. We set the fixed length of sentences as 80 by padding shorter sentences and cutting off longer sentences. We utilize LSTM with 100 hidden units. We also set the head number of multi-head attention as 2, and the hidden units of each-head attention as 100. The dropout rate is 0.2, the batch size is 10, and the learning rate is 0.3.

B. Overall Performance

We compare our performance with the following state-of-the-art methods:

- **Cross-Entity** [7] is a feature-based model based on cross-entity inference.
- **Cross-Event** [24] exploits relations among event types from the training set to help predict events in the test set.
- **DMCNN** [11] uses dynamic multi-pooling CNN to capture features.
- **Bi-RNN** [12] utilizes Bi-RNN to capture the dependencies of sequence.
- **PSL** [25] uses probabilistic soft logic to exploit latent and global information.
- **ANN** [20] uses argument information via supervised attention mechanisms.
- **GMLATT** [26] exploits cross-lingual attention to solve ambiguity problems.
- **HBTNGMA** [14] uses hierarchical and bias tagging networks with gated multi-level attention mechanisms.
- **HNN** [13] combines Bi-LSTM and CNN networks.

Table I shows the overall performance compared with the above state-of-the-art methods. These results demonstrate that our method is effective and the performance is better than all other state-of-the-art methods. F_1 -scores of trigger identification and event type classification can reach 76.8% and 74.2% respectively, with a gain of 0.9% and 0.8% higher than the highest performance in other models. Furthermore, compared with other methods, the precision rate and recall rate of our proposed model is relatively balanced. The following reasons can explain why our method is better:

- Neural network-based models perform better than feature-based models because neural network-based models mitigate error propagation problems.
- In this task, Bi-LSTM can capture the context features and long-distance features of sentences more effective than Bi-RNN and CNN.
- Mention Segmentation can concentrate more on event-related information and reduce the interference of redundant words.
- Multi-head attention can get more attention to key-words in sentences and reduce the interference of meaningless words.

C. Effect of Event Mention

To confirm event mention help improve ED, we verify whether ground-truth event mention (ground-truth event

Table I
PERFORMANCE OF ALL ED SYSTEMS ON ACE 2005. (N/A: THE PAPER DIDN'T LIST RESULTS OF THIS TASK)

Methods	Trigger Identification			Type Classification		
	P	R	F ₁	P	R	F ₁
Cross-Entity [7]	n/a	n/a	n/a	72.9	64.3	68.3
Cross-Event [24]	n/a	n/a	n/a	68.7	68.9	68.8
DMCNN [11]	80.4	67.7	73.5	75.6	63.6	69.1
Bi-RNN [12]	68.5	75.7	71.9	66.0	73.0	69.3
PSL [25]	n/a	n/a	n/a	75.3	64.4	69.4
ANN [20]	n/a	n/a	n/a	76.8	67.5	71.9
GMLATT [26]	80.9	68.1	74.1	78.9	66.9	72.4
HBTNGMA [14]	n/a	n/a	n/a	77.9	69.1	73.3
HNN [13]	80.8	71.5	75.9	84.6	64.9	73.4
our model	77.2	76.5	76.8	74.1	74.3	74.2

Table II
PERFORMANCE OF GROUND-TRUTH EVENT MENTION AND PREDICTED EVENT MENTION IN DIFFERENT MODELS. * DENOTES THAT MENTION DERIVES FROM ACE 2005.

Methods	Identify Trigger			Classify Type		
	P	R	F ₁	P	R	F ₁
Bi-LSTM	72.0	77.4	74.6	68.6	73.8	71.1
+ground-truth mention*	87.2	86.2	86.7	82.8	81.8	82.3
Bi-LSTM+Att [27]	74.5	75.1	74.7	72.1	72.6	72.3
reproduced Bi-LSTM+Att	73.4	76.9	75.1	70.8	72.3	71.5
reproduced + mention	76.6	76.5	76.5	74.2	74.0	74.1
GAN [16]	75.3	78.8	77.0	71.3	74.7	73.0
reproduced GAN	75.2	76.5	75.8	72.1	73.3	72.7
reproduced + mention	79.2	75.0	77.1	76.7	72.6	74.6

mention is referred to annotated event mention in ACE 2005) is effective, and add predicted event mention to the reproduced previous works respectively.

1) *Ground-truth Event Mention*: Ground-truth event mention is concatenated into the input of Bi-LSTM. We set the Bi-LSTM as the baseline model, and add the ground-truth event mention as the feature to the Bi-LSTM. Experimental results in the top of Table II show that ground-truth event mention provides a large F_1 improvement of 12.1% and 11.2% over the baseline on ED task, and can fully indicate the effectiveness of ground-truth event mention.

2) *Predicted Event Mention*: Predicted event mention is added as a feature to the input of other models. We reproduce Bi-LSTM+Att [27] and GAN[16] model. Bi-LSTM+Att model uses entity relations, but we reproduce the model without using entity relations. The middle of Table II illustrates that predicted event mention on the Bi-LSTM+Att plays a critical role, yielding a 1.8% F_1 improvement for event type classification and trigger identification respectively. Furthermore, we conduct experiments on GAN model with predicted event mention, and the bottom of Table II shows that F_1 -scores can reach 74.6% on type classification and 77.1% on trigger identification. Both can prove that the event mention is effective.

D. Effect of Multi-head Attention Mechanism

We utilize Bi-LSTM as the baseline model and add the multi-head attention mechanism to the Bi-LSTM. The result is shown in Table III, the F_1 -scores of trigger

Table III
PERFORMANCE OF BI-LSTM, BI-LSTM WITH MULTI-HEAD ATTENTION, BI-LSTM WITH PREDICTED EVENT MENTION, AND MULTI-HEAD ATTENTION-BASED BI-LSTM WITH PREDICTED EVENT MENTION.

Methods	Trigger Identification			Type Classification		
	P	R	F ₁	P	R	F ₁
Bi-LSTM	72.0	77.4	74.6	68.6	73.8	71.1
+multi-head att	73.8	76.5	75.1	69.6	73.8	71.6
+mention	75.0	76.5	75.7	71.5	74.3	72.9
+multi-head att +mention	77.2	76.5	76.8	74.1	74.3	74.2

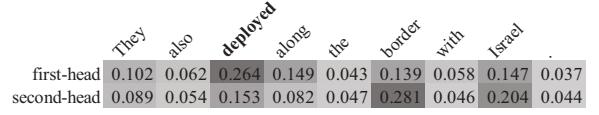


Figure 3. Visualization of the two-head attention results of “deployed”

identification and event type classification are both 0.5% higher than Bi-LSTM, indicating that the Bi-LSTM based on the multi-head attention is effective. Additionally, after incorporating predicted event mention and multi-head attention into the baseline, the F_1 -scores of trigger identification and event type classification are 2.2% and 3.1% higher than the baseline, 1.1% and 1.3% higher than Bi-LSTM with predicted event mention. This indicates that multi-head attention is effective.

We exploit two-head attention before and after the Bi-LSTM. Here, we will introduce how two-head attention works. As shown in sentence 3), the trigger word is “deployed”, entities are “They”, “border” and “Israel”. Figure 3 shows the visualization of the two-head attention results of “deployed” in 3). As can be seen from Figure 3, the score of “deployed” in the first-head attention is the highest, indicating that the keyword “deployed” in the sentence is paid more attention by the attention mechanism. Furthermore, the scores of entities “They”, “border” and “Israel” are higher than other words, indicating that the correlation between entities and triggers is relatively large and two-head attention can notice it. Thus, this also verifies the effectiveness of multi-head attention.

3) *They also deployed along the border with Israel*.

VI. CONCLUSION

In this paper, we propose a novel method for ED task, which extracts event mention and then uses mention segmentation to improve ED with multi-head attention-based Bi-LSTM. The purpose is to concentrate more on event-related information in sentences and avoid the interference of meaningless words. The experimental results demonstrate the effectiveness of our proposed method.

ACKNOWLEDGMENT

This work was supported by National Natural Science Foundation of China (Grants No. 61672368, 61672367, and 2017YFB1002104). The authors would like to thank the anonymous reviewers for their insightful comments

and suggestions. Yu Hong, Professor in Soochow University, is the corresponding author of the paper, whose email address is tianxianer@gmail.com.

REFERENCES

- [1] S. Duan, R. He, and W. Zhao, “Exploiting document level information to improve event detection via recurrent neural networks,” in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, 2017, pp. 352–361.
- [2] O. Vinyals, M. Fortunato, and N. Jaitly, “Pointer networks,” in *Advances in Neural Information Processing Systems*, 2015, pp. 2692–2700.
- [3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [4] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.
- [5] P. Gupta and H. Ji, “Predicting unknown time arguments based on cross-event propagation,” in *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, 2009, pp. 369–372.
- [6] R. Grishman, “The impact of task and corpus on event extraction systems.” in *LREC*, 2010.
- [7] Y. Hong, J. Zhang, B. Ma, J. Yao, G. Zhou, and Q. Zhu, “Using cross-entity inference to improve event extraction,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 1127–1136.
- [8] Q. Li, H. Ji, and L. Huang, “Joint event extraction via structured prediction with global features,” in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2013, pp. 73–82.
- [9] S. Liu, Y. Chen, S. He, K. Liu, and J. Zhao, “Leveraging framenet to improve automatic event detection,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2016, pp. 2134–2143.
- [10] T. H. Nguyen and R. Grishman, “Event detection and domain adaptation with convolutional neural networks,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, vol. 2, 2015, pp. 365–371.
- [11] Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao, “Event extraction via dynamic multi-pooling convolutional neural networks,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, 2015, pp. 167–176.
- [12] T. H. Nguyen, K. Cho, and R. Grishman, “Joint event extraction via recurrent neural networks,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 300–309.
- [13] X. Feng, B. Qin, and T. Liu, “A language-independent neural network for event detection,” *Science China Information Sciences*, vol. 61, no. 9, p. 092106, 2018.
- [14] Y. Chen, H. Yang, K. Liu, J. Zhao, and Y. Jia, “Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 1267–1276.
- [15] Y. Zhao, X. Jin, Y. Wang, and X. Cheng, “Document embedding enhanced event detection with hierarchical and supervised attention,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, 2018, pp. 414–419.
- [16] Y. Hong, W. Zhou, G. Zhou, Q. Zhu *et al.*, “Self-regulation: Employing a generative adversarial network to improve event detection,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2018, pp. 515–526.
- [17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [18] H. Xu, B. Liu, L. Shu, and P. S. Yu, “Bert post-training for review reading comprehension and aspect-based sentiment analysis,” *arXiv preprint arXiv:1904.02232*, 2019.
- [19] G. R. Doddington, A. Mitchell, M. A. Przybocki, L. A. Ramshaw, S. M. Strassel, and R. M. Weischedel, “The automatic content extraction (ace) program-tasks, data, and evaluation.” in *Lrec*, vol. 2, 2004, p. 1.
- [20] S. Liu, Y. Chen, K. Liu, J. Zhao *et al.*, “Exploiting argument information to improve event detection via supervised attention mechanisms,” 2017.
- [21] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv preprint arXiv:1301.3781*, 2013.
- [22] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [23] S. Liu, R. Cheng, X. Yu, and X. Cheng, “Exploiting contextual information via dynamic memory network for event detection,” *arXiv preprint arXiv:1810.03449*, 2018.
- [24] S. Liao and R. Grishman, “Using document level cross-event inference to improve event extraction,” in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 789–797.
- [25] S. Liu, K. Liu, S. He, and J. Zhao, “A probabilistic soft logic based approach to exploiting latent and global information in event classification,” in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [26] J. Liu, Y. Chen, K. Liu, and J. Zhao, “Event detection via gated multilingual attention mechanism,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [27] J. Zhang, W. Zhou, Y. Hong, J. Yao, and M. Zhang, “Using entity relation to improve event detection via attention mechanism,” in *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer, 2018, pp. 171–183.

Chinese Spelling Check based on Sequence Labeling

Zijia Han

School of Computer Science and Technology,
Heilongjiang University
Harbin, China
e-mail: hanzjianlp@hotmail.com

Qiansheng Wang

School of Computer Science and Technology,
Heilongjiang University
Harbin, China
e-mail: chncwang@gmail.com

Abstract—Chinese has been widely used by people from all over the world. Various Chinese spelling errors may occur inevitably from Chinese as Foreign Language (CFL) beginners. This paper presents a method for Chinese spelling check to detect and correct spelling errors in a sentence. Our approach is based on the sequence labeling model using the bi-direction LSTM network. We also utilize the Pinyin feature and Chinese strokes feature to improve performance. The evaluation on the SIGHAN-8 shows that our approach gets good performance on both detection and correction tasks.

Keywords—Chinese spelling check, Bi-LSTM, sequence labeling, Pinyin, Chinese strokes

I. INTRODUCTION

With the development of China's economy and the popularity of Chinese, booming people are learning Chinese as a Foreign Language (CFL). The number of CFL learners is expected to become larger for the years to come. First CFL learners tend to find Chinese is a difficult language for its confusing tones and complicated characters, and they are more prone to misspelling Chinese words or characters than native users. In this situation, a spelling check system is necessary for these people.

In this paper, we present a Chinese spelling check method based on a sequence labeling model. A sentence with potential typos is corrected by the model and a new sentence without typo is re-generated character by character. In a sentence, most characters are correct and origin-correction sentence pairs are equal in length, which means the origin-correction sentence pairs have a strong correlation. However, the neural machine translation (NMT) model is designed to translate one language into another language even if they are totally different. Intuitively, compared to the NMT model, the sequence labeling model is more suitable to CSC task. Besides, the sequence labeling model has the strength of lightweight, so it can save the computing resource and makes the model trained faster.

There are two main types of spelling errors: pronunciation similarity errors and shape similarity errors[1]. For example, a sentence with a typo like “我覺得你們會好好的碗。”(I think you will bowl yourselves.),

Chengguo Lv

School of Computer Science and Technology,
Heilongjiang University
Harbin, China
e-mail: 13936160042@163.com

Guohong Fu

School of Computer Science and Technology,
Soochow University
Suzhou, China
e-mail: ghfu@hotmail.com

where the typo “碗” (bowl) should be corrected to “玩” (enjoy). That is a pronunciation similarity error. For another sentence “我覺得你們會好好的元。”(I think you will meta yourselves.), where the typo “元”(meta) should be corrected to “玩”(enjoy). That is a shape similarity error.

To address both of these errors, we incorporate the Pinyin feature and the strokes feature into our sequence labeling model separately. Pinyin is widely used as a phonetic system. Strokes, also call CJK strokes, are the smallest unit of Chinese characters. The characters with similar pronunciations have similar Pinyin while the characters with similar shapes have similar strokes.

Experiments show that the performance of the sequence labeling model is better than the NMT model. Our model performs even better utilizing the Pinyin feature and the strokes feature.

The rest of the paper is organized as follows. We review the related work in Section II. Then we introduce our method based on sequence labeling with utilizing the Pinyin feature and strokes feature to improve the performance of the model in Section III. In section IV, we detail the evaluation metrics and experimental performance of our models. Finally, we conclude the paper and explore future work in Section V.

II. RELATED WORK

Chinese Spelling Check (CSC)[2] has been active research in recent years. It can be divided into two sub-tasks: spelling error detection and spelling error correction. There other similar tasks like Chinese Grammatical Error Diagnosis (CGED)[3] and Spelling Error Correction (SEC)[4]. In our work, we address the task of correcting Chinese spelling errors for CFL learners and raise an effective model with linguistic features to improve the performance.

Early works on spelling correction tasks are mainly two kinds of methods: rule-based methods and statistical-based methods. Mangu and Brill[5] proposed a rule-based approach for automatically acquiring linguistic knowledge to help correct spelling errors. Mays et al.[6] proposed a statistical-based model on spelling error correction tasks,

where context information is incorporated into the detection and correct spelling errors.

Gao et al.[6] introduced statistical machine translation to search query spelling correction. Lopez et al.[7] improved the statistical machine translation model to build a data-driven system. Recently, neural machine translation (NMT) has been applied to spelling check task. Yuan and Briscoe[8] first present NMT model for English grammatical error correction.

Rei and Yannakoudakis[9] first applied the sequence labeling model on error detection task. Their models are based on bi-direction LSTM and are able to outperform other participants on detecting errors in learner writing.

As to the Chinese spelling check task, Xie et al.[10] utilized an N-gram language model on CSC. Their models are based on the bi-gram and tri-gram language model as well as Chinese word segmentation. Wang and Liao[11] proposed an approach based on word vector and conditional random field.

Jin et al.[12] utilized a hybrid approach to Chinese spelling correction. They integrated three models including the n-gram language model, Pinyin-based language model and tone-based language model to improve the performance of Chinese checking spelling error system.

Generating artificial misspelled sentences is a practical way to help train models, which can address the problem of corpus lack. Li et al.[13] added artificial error data to expand the dataset and applied the NMT model on the Chinese spelling check task. The result shows their approach is able to significantly improves the performance.

A united framework for Chinese spelling check called HANSpeller was proposed by Zhang et al [14]. The framework is based on extended HMM and ranker-based models, together with a rule-based model for further polishing. They afterward proposed HANSpeller++[15] based on their previous work. The improvements including candidates generating, candidates re-ranking and final global decision and the result show the state-of-the-art performance on CSC tasks.

Compared to previous research on Chinese spelling check, we apply the sequence labeling model on Chinese spelling check and it shows that our approach performs better than the NMT model. Furthermore, we utilize the Pinyin feature and the strokes feature to enhance performance on typo detection and correction.

III. METHOD

We consider the Chinese spelling check as a sequence labeling task. A sentence with potential misspelling characters will be re-generated as an error-free sentence character by character through a sequence labeling neural network.

In our approach, we prepare training data from SIGHAN successive CSC tasks (Section A). We then introduce the architecture of our sequence labeling based neural network model (Section B) and incorporate the Pinyin feature 0(Section C) along with the strokes feature (Section D).

A. Data Processing

We use SIGHAN-2013 CSC Datasets, CLP-2014 CSC datasets, and SIGHAN-2015 CSC training data as our training data. The source texts are a set of the sentence with mistakes and corrections. We extract the typos in the source texts and replace them with correction characters. For example, the sentence is in SGML format

```
<ESSAY title="上學遲到">
<TEXT>
<PASSAGE id="A2-3006-1">他快進座下，下課以後跟老師說「不好意思！昨晚很晚才睡覺。」
</PASSAGE>
</TEXT>
<MISTAKE id="A2-3006-1" location="4">
<WRONG>座下</WRONG>
<CORRECTION>坐下</CORRECTION>
</MISTAKE>
</ESSAY>
```

is converted to

“他快進座下，下課以後跟老師說「不好意思！昨晚很晚才睡覺。」”

“他快進坐下，下課以後跟老師說「不好意思！昨晚很晚才睡覺。」”

The upper one is the original sentence with potential errors and the one below is the error-free sentence after correction. We replace the character “座” (seat) with the right character “坐” (sit).

B. Sequence Labeling Model

We train a character-based sequence labeling model, to translate a sentence with potential errors to an error-free sentence. The source sentence $X = [x_1, x_2, \dots, x_J]$ and the target sentence $Y = [y_1, y_2, \dots, y_K]$. We use origin-and-correction sentence pairs as source-and-target training sentence pairs.

It has been shown that word embedding plays an important role to improve sequence labeling performance [16]. We use Chinese word2vec to convert $X = [x_1, x_2, \dots, x_J]$ into vectors $E = [e_{x1}, e_{x2}, \dots, e_{xJ}]$, which is fed into a bi-direction long-short memory (LSTM) [17] model. Each character will be predicted to a new character after LSTM hidden layer via a softmax classifier.

Recurrent neural networks (RNN)[18] are a promising model to capture long-distance dependencies while in practice they fail due to gradient vanishing or exploding problems.[19] LSTMs are designed to solve these problems. It is proved that LSTMs have better performance than RNNs dealing with long sequences. An LSTM unit is composed of three gates including input gate, forget gate and output gate, which control the cell to forget or pass on the information to the next time step.

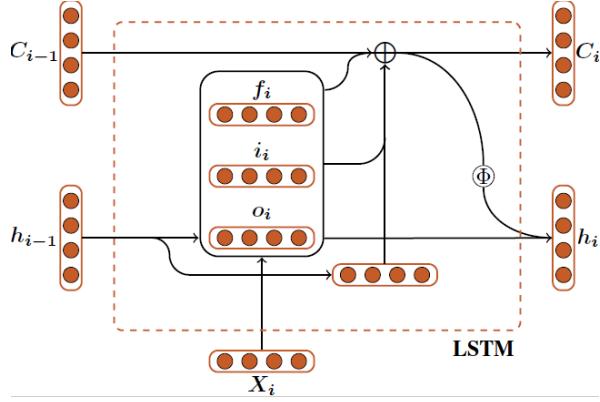


Figure 1. The architecture of LSTM

Figure 1 shows the architecture of an LSTM unit. The formulas of an LSTM unit are below:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (1)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (2)$$

$$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C) \quad (3)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (4)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t \odot \tanh(C_t) \quad (6)$$

where x_t is the input vector with word embedding representation and h_t is the hidden state at time t. W_f, W_i, W_C, W_o is the weight matrices for hidden state h_t and b_f, b_i, b_C, b_o is the bias vectors. σ denotes sigmoid function.

In order to get both past and future contexts, we use bi-direction LSTM (Bi-LSTM)[20] to improve our model. The input vectors are fed to two LSTMs in opposite directions. Then two hidden states of LSTM are concatenated to form the final outputs of the hidden layers.

C. Incorporating Pinyin Feature

Pinyin is a scheme of using Latin letters as the phonetic symbol of Mandarin Chinese. It includes four diacritics denoting tones. We use Pinyin as the additional feature to improve the performance of the spelling check task.

First, we obtain the Pinyin of every character. Both with-tone and without-tone Pinyin are incorporated into our model. Pinyin input vectors are concatenated with Chinese character input vectors, and we feed them into Bi-LSTM network. Figure 2 shows the network we use to extract Pinyin feature together with Chinese character.

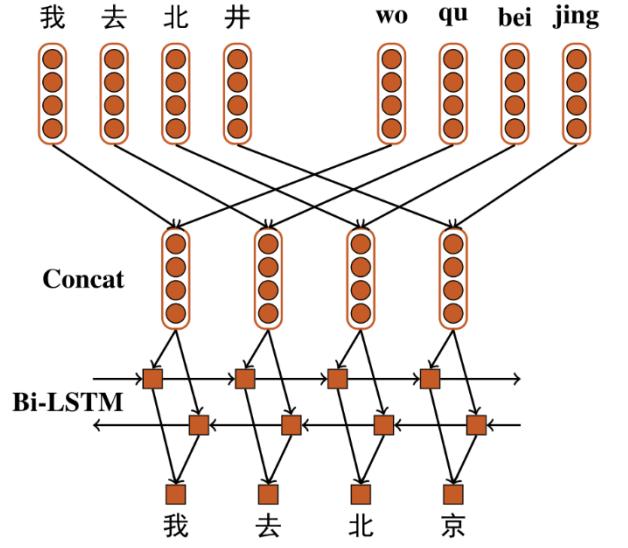


Figure 2. The architecture of our network using the Pinyin feature. The character representation is converted to word embedding. Then the character vector is concatenated with the Pinyin vector before feeding into the Bi-LSTM network. The outputs are the correction character after a softmax layer.

D. Incorporating Strokes Feature

Chinese characters have various strokes like Dot (“丶”, “点”), Horizontal (“—”, “横”), Vertical (“|”, “竖”) and so on. These strokes make up all kinds of complex Chinese characters. First, we break up characters and extract strokes from all the characters in training data. Then we gather and number them, obtain the stroke indexes of each character. For example, the stroke indexes of character “才” (ability; talent; just) are [2,6,4], where ‘2’ denotes Horizontal (“—”), ‘0’ denotes J hook (“J”), ‘4’ denotes Throw (“J”).

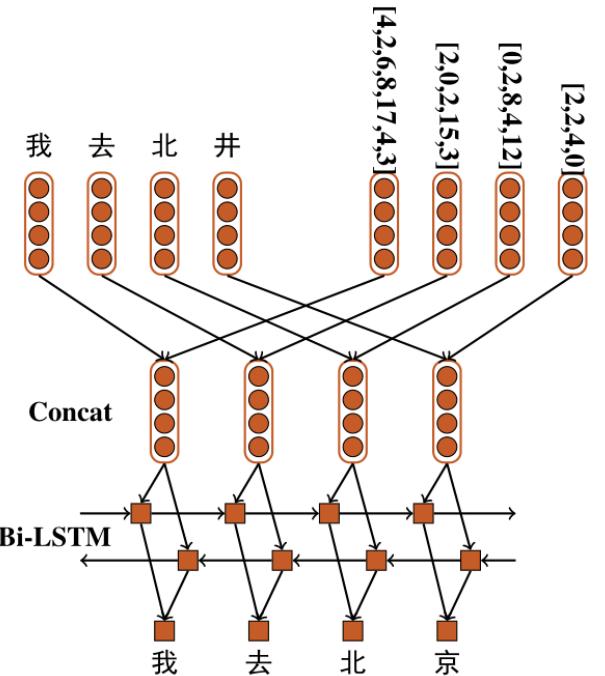


Figure 3. The architecture of our model with strokes feature. The character representation is converted to word embedding. The character vector is concatenated with the strokes indexes vector before feeding

into the Bi-LSTM network. The outputs is the correction character after a softmax layer.

After vectorization, the stroke feature is concatenated with character input vectors. We feed them into Bi-LSTM and get the prediction outputs. Figure 3 shows the model of Bi-LSTM with strokes feature.

IV. EXPERIMENT

In this section, we introduce the dataset used in Section A, and the evaluations of our task in Section B. Then we detail the hyper-parameters of sequence labeling model and NMT model in Section C. Finally, in Section D, we present the performance of these models.

A. Dataset

Training Data: We use CLP-2014 CSC Datasets[21] and SIGHAN-2013 CSC Datasets[2] as well as SIGHAN-2015 CSC training data as our training data[22]. There are 897 sentences without error and there are 7641 sentences with one or more errors. The total number of sentences in training data is 8538.

Test Data: We use SIGHAN-2015 CSC test data[22] as our test data. There are 897 sentences without error and there are 7641 sentences with one or more errors. The total number of sentences in training data is 8538.

B. Evaluation

The criteria we use for judging correctness is divided into two subtasks. One is the detection level and the other is the correction level. For the detection level, all locations of incorrect characters in a given passage should be completely identical with the gold standard. For detection level, locations and corresponding corrections of incorrect characters should be completely identical with the gold standard.

Table I shows the Confusion matrix. The following metrics are measured at both levels with the help of the confusion matrix.

TABLE I. CONFUSION MATRIX FOR EVALUATION

Confusion Matrix		System Results	
		Positive	Negative
Gold Standard	(Error)	(No Error)	
	Positive	TP	FN
	Negative	FP	TN

As is shown in the following, the Evaluation metric consists of false positive rate, accurate rate, precision rate, recall rate and F1-score.

$$\text{False Positive Rate (FPR)} = \frac{FR}{FP + TN} \quad (7)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (9)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (10)$$

$$F1 - score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (11)$$

C. Hyper-parameters of our Model

The hyper-parameters of our sequence labeling models are the same among these models. We use traditional Chinese word2vec with $d = 200$ dimensions as our word embedding, which is trained with CTB-60 dataset.

We set our input vector dimension as 200. The dimension of Pinyin input vectors and stroke input vectors are 200 as well. Our model contains a 1-layer Bi-LSTM network with 400 hidden nodes. We use Adam Algorithm[23] as the optimizer to train the model with learning rate 0.001. The Learning rate decay method is applied in our model with 0.05 decay each epoch. The batch size is set as 10 and the dropout is 0.5.

We train an NMT model as our baseline model. We use a sequence-to-sequence model and both encoder and decoder are a bi-direction gated recurrent unit (GRU)[24]. We set the input vector dimension as 512 and the hidden layer dimension is 512. We use Adam with learning rate 0.0001.

D. Result

Table II shows the evaluation on the detection level, table III shows the evaluation on the correction level. Seq2Seq denotes the model based on the sequence-to-sequence approach, which is the NMT model. SL denotes the model based on the sequence labeling approach, following with which is the sequence labeling model incorporating the Pinyin feature and the strokes feature.

As we can see, models based on sequence labeling, which is Bi-LSTM, get much better performances than the sequence-to-sequence model on both evaluations. When we using the Pinyin feature or strokes feature, the performances are better than Bi-LSTM only models. The combined models with the Pinyin feature and the strokes feature performs are superior to Bi-LSTM only models but inferior to using the Pinyin feature only models.

We consider that the excessive features are not favorable to our model. In addition, most of the Chinese typos are pronunciation similar error. That's the reason of why the Individual Pinyin feature model performed better than the strokes feature and the combine feature.

Experiments show that the model trained with the Pinyin feature gets the best recall and F1-score on the detection level and correction level. While the model trained with the strokes feature gets the best precision and accuracy on both levels.

Moreover, the combined model trained with the Pinyin feature and the strokes feature can be regarded as a compromise model between the above two models.

V. CONCLUSION

This paper proposes a spelling check approach based on the sequence labeling model. We also incorporate the Pinyin feature and strokes feature to improve the

evaluation performance. Our approach achieved a considerable result at SIGHAN-2015 Chinese spelling check task.

Future works on Chinese spelling check (CSC) are as follows: (1) The critical problem for CSC task is the lack of corpus source. Sentences with potential errors are difficult to collect or generate. Ample corpus is also a necessary condition to train a fancy model. Finding a way to solve the corpus deficiency problem is key to the CSC task. (2) Transfer learning ought to happen used in CSC. We can use transfer learning to enhance the proximity of origin corpus and artificial corpus

ACKNOWLEDGMENT

The authors would like to thank the organizers and the reviewers of IALP-2019 for their dedicated and professional works. This research was supported by the National Natural Science Foundation of China No. 61672211).

TABLE II. PERFORMANCE OF OUR MODEL AND SEQ2SEQ MODEL ON CORRECTION LEVEL

	Seq2Seq	SL	SL+Pinyin	SL+strokes	SL+Pinyin+strokes
FPR	0.1309	0.1064	0.1119	0.0881	0.1064
Accuracy	0.5645	0.5976	0.6076	0.6104	0.6085
Precision	0.6651	0.7467	0.752	0.7848	0.7593
Recall	0.26	0.3076	0.3327	0.3147	0.3291
F1-score	0.3739	0.4357	0.4613	0.4493	0.4592

TABLE III. PERFORMANCE OF OUR MODEL AND SEQ2SEQ MODEL ON DETECTION LEVEL

	Seq2Seq	SL	SL+Pinyin	SL+strokes	SL+Pinyin+strokes
FPR	0.1611	0.1014	0.1119	0.0807	0.1064
Accuracy	0.5845	0.6240	0.6231	0.6267	0.6231
Precision	0.6500	0.7615	0.7681	0.8112	0.7743
Recall	0.3159	0.3377	0.3633	0.3399	0.3579
F1-score	0.4252	0.4679	0.4933	0.4791	0.4895

REFERENCES

- [1] C.-L. Liu, M.-H. Lai, K.-W. Tien, Y.-H. Chuang, S.-H. Wu, and C.-Y. Lee, "Visually and phonologically similar characters in incorrect Chinese words: Analyses, identification, and applications," ACM Transactions on Asian Language Information Processing (TALIP), vol. 10, no. 2, pp. 10, 2011.
- [2] S.-H. Wu, C.-L. Liu, and L.-H. Lee, "Chinese spelling check evaluation at SIGHAN Bake-off 2013." pp. 35-42, 2013.
- [3] Lee L H, Gaoqi R A O, Yu L C, et al. Overview of nlp-tea 2016 shared task for chinese grammatical error diagnosis[C]/Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA2016). 2016: 40-48.
- [4] K. Oflazer, "Error-tolerant finite-state recognition with applications to morphological analysis and spelling correction," Computational Linguistics, vol. 22, no. 1, pp. 73-89, 1996.
- [5] L. Mangu, and E. Brill, "Automatic rule acquisition for spelling correction." pp. 187-194, 1997.
- [6] Gao J, Hwang M, Huang X D, et al. Statistical Machine Translation Based Search Query Spelling Correction: U.S. Patent Application 13/296,640[P]. 2013-5-16.
- [7] Lopez Ludeña V, San Segundo Hernández R, Montero Martínez J M, et al. Architecture for text normalization using statistical machine translation techniques[J]. 2012.
- [8] Z. Yuan, and T. Briscoe, "Grammatical error correction using neural machine translation." pp. 380-386, 2016.
- [9] M. Rei, and H. Yannakoudakis, "Compositional sequence labeling models for error detection in learner writing," arXiv preprint arXiv:1607.06153, 2016.
- [10] W. Xie, P. Huang, X. Zhang, K. Hong, Q. Huang, B. Chen, and L. Huang, "Chinese spelling check system based on n-gram model." pp. 128-136, 2016.
- [11] Y.-R. Wang, and Y.-F. Liao, "Word Vector/Conditional Random Field-based Chinese Spelling Error Detection for SIGHAN-2015 Evaluation," Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing. pp. 46-49, 2015.
- [12] P. Jin, X. Chen, Z. Guo, and P. Liu, "Integrating pinyin to improve spelling errors detection for Chinese language." pp. 455-458, 2014.
- [13] Li C W, Chen J J, Chang J S. Chinese spelling check based on neural machine translation[C]/Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation. 2018.
- [14] Zhang Q, Zhang S, Hou J, et al. HANSPELLer: A Unified Framework for Chinese Spelling Correction[C]/International Journal of Computational Linguistics & Chinese Language Processing, Volume 20, Number 1, June 2015-Special Issue on Chinese as a Foreign Language. 2015.
- [15] S. Zhang, J. Xiong, J. Hou, Q. Zhang, and X. Cheng, "HANSPELLer++: A Unified Framework for Chinese Spelling Correction," Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing. pp. 38-45, 2015.
- [16] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," Journal of machine learning research, vol. 12, no. Aug, pp. 2493-2537, 2011.
- [17] S. Hochreiter, and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735-1780, 1997.

- [18] C. Goller, and A. Kuchler, "Learning task-dependent distributed representations by backpropagation through structure." pp. 347-352, 1996.
- [19] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," IEEE transactions on neural networks, vol. 5, no. 2, pp. 157-166, 1994.
- [20] C. Dyer, M. Ballesteros, W. Ling, A. Matthews, and N. A. Smith, "Transition-based dependency parsing with stack long short-term memory," arXiv preprint arXiv:1505.08075, 2015.
- [21] Yu L C, Lee L H, Tseng Y H, et al. Overview of SIGHAN 2014 bake-off for Chinese spelling check. 126-132, 2014.
- [22] Y.-H. Tseng, L.-H. Lee, L.-P. Chang, and H.-H. Chen, "Introduction to sighan 2015 bake-off for chinese spelling check." pp. 32-37, 2015.
- [23] D. P. Kingma, and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.
- [24] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," arXiv preprint arXiv:1409.1259, 2014.

Research on Tibetan Text Classification Method Based on Neural Network

Zhensong Li^{1,2}, Jie Zhu^{1,2}, Zhixiang Luo^{1,2}, Saihu Liu^{1,2}

¹ School of Electronics Engineering and Computer Science Tibet University

² Tibetan Information Technology National Joint Engineering Research Center
Lhasa, China

{lizhensong_3.14159, rocky_tibet, 2328866327, 593731006}@qq.com

Abstract—Text categorization is an important task in natural language processing, and it has a wide range of applications in real life. In this paper, two N-Gram feature models (MLP, FastText) and two sequential models (sepCNN, Bi-LSTM) are used to study the automatic classification for Tibetan text based on syllables and vocabulary. The experiment on Tibetan language data collected by China Tibet News Network shows that the classification accuracy is about 85%.

Keywords- *Tibetan text classification; neural network; N-Gram model; sequence model*

I. INTRODUCTION

Text classification is to classify and mark text information according to certain rules and standards. Tibetan text categorization is an important research task of Tibetan information processing technology, which plays a fundamental role in search, public opinion analysis, document retrieval, and other application fields.

With the development of Tibetan information processing technology and the arrival of the era of big data, the Tibetan text data in websites, emails, forums, comments, and real-time communications are growing rapidly. These data contain abundant knowledge and bad information. To classify these data, we can analyze and process the information. And use brings great convenience.

At present, the text categorization process can be divided into feature engineering and classification model construction. Feature engineering can be divided into text preprocessing, feature extraction and text representation, to convert the original text into structured information that can be recognized and processed by computer. According to the difference of text preprocessing, the classification model can be divided into the model using a word set (bag of words) and the model using a word order (sequence model). Among them, the bag of words model includes logistic regression, support vector machine (SVM), Bayesian and Multi-Layer Perception, and the sequence model includes convolutional neural network (CNN) and cyclic neural network (RNN).

In this paper, we will use Multi-Layer Perception (MLP), fast text classifier (Fast Text), depthwise separable convolutional network (SepCNN) and Bi-directional Long Short-Term Memory network (Bi-LSTM) to study the Tibetan text classification method based on syllables and vocabulary.

II. RELATED WORK

In the early stage of text categorization, people used the rules constructed by professionals to mark the text and developed a classification method based on knowledge engineering. This method relies heavily on rules and has

poor generalization [1]. Later, the emergence and development of machine learning replaced the rule classification method and became the mainstream method of text classification. The main machine learning methods in text classification are Bayesian classification [2], KNN classification [3], Logistic classification [4], SVM classification [5], neural network classification [6].

With the development of word embedding [7,8], the neural network has achieved better results than other machine learning methods in many aspects of natural language processing [9]. It has also become the preferred classification method and an important research direction for text categorization tasks.

In 2016, Facebook opened a text categorization tool, FastText [10], which uses only shallow neural networks. Compared with other categorization tools, it keeps the categorization effect while greatly shortening the training time. Convolutional neural networks have achieved great success in image processing. In 2014, Yoon Kim applied a convolutional neural network to text categorization task [11]. It used the convolution window to obtain local information of sequential text and achieved good classification results in many text categorization tasks. Cyclic neural networks have great advantages when dealing with variable length sequences. In many text classification tasks, variants of the cyclic neural network Bi-LSTM [12] can be found, which can acquire the relationship between context and context well, and the use of gate mechanism can control the length of memory and avoid the problem of long-term dependence.

At present, there are less research results in the Tibetan text classification. In 2011, Chuncheng Xiang [13] and others studied the automatic text classification of Tibetan web pages based on rules and implemented the classification tag of Tibetan web pages by establishing the Tibetan category feature vocabulary and comparing the entries of Tibetan web pages. Huiqiang Jia [14] used the KNN algorithm to classify Tibetan texts. In 2014, Renqing Nobu [15] used the maximum entropy model design to implement the Tibetan bad text recognition system. In 2018, Hongyun Jia [16,17] studied the Tibetan text classification method using Logistic and SVM models. It can be found that in the current Tibetan classification research, there are only a small number of classification studies based on rules and using traditional machine learning methods, and no literature on applying neural network models to Tibetan text classification research has been found. Because Tibetan lacks open classification test corpus, and the corpus used by each researcher is different, the experimental results vary greatly, and the accuracy is difficult to evaluate and analyze.

This paper experiments on the Tibetan corpus collected by China Tibet News Network, and compares the four neural network models of MLP, FastText, sepCNN and Bi-LSTM, based on the syllable and vocabulary classification of Tibetan text.

III. NEURAL NETWORK MODEL

A. N-Gram feature model

MLP and FastText belong to the N-Gram feature model. They use N-Gram of lexical granularity and N-Gram of syllable granularity in preprocessing. The original sample can be regarded as a sequence of Tibetan syllables or Tibetan vocabulary $T_i = [w_1, w_2 \dots w_n]$, where T_i represents the nth original sample and w_n is the nth syllable or vocabulary in text T_i . The N-Gram operation is performed on the text to generate the corresponding N-Gram feature set $G_i = [x_1, x_2 \dots x_m]$, where G_i is the N-Gram feature set generated by text T_i , and x_m is an N-Gram feature in the G_i feature set. In this paper, three N-Gram methods, unigram, bigram, and trigram, are used to generate the N-Gram feature set of Tibetan syllables and the N-Gram feature set of Tibetan vocabulary. Specific Tibetan N-Gram feature generation is shown in TABLE I.

TABLE I. RESULTS OF N-GRAM FEATURE GENERATION IN TIBETAN

sentence	ཨୱାଆ ཀ୍ୱାଳ ཁେନ୍ ཁେନ୍ (The students are playing.)
syllable	ཨୱା ଆ ອ୍ୱାଳ ແେ ແ
N-Gram Characteristics of Syllable Granularity	‘ୱା’, ‘ନ୍ତିନ କ’’, ‘ନ୍ତିନ କ ອ୍ୱାଳ’’, ‘କ’’, ‘ନ ອ୍ୱାଳ’’, ‘ଭ୍ୱାଳ ແେ’’, ‘ଭ୍ୱାଳ ແେ କ’’, ‘ଭ୍ୱାଳ ແେ ແ’’, ‘ଭ୍ୱାଳ ແେ ແ କ’’, ‘ଭ୍ୱାଳ ແେ ແ କ କ’’, ‘ଭ୍ୱାଳ ແେ ແ କ କ କ’’, ‘ଭ୍ୱାଳ ແେ ແ କ କ କ କ’’
vocabulary	ཨୱାଆ ཀ୍ୱାଳ ཁେନ୍ ཁେନ୍
N-Gram Characteristics of Lexical Granularity	‘ୱାଆ’’, ‘ନ୍ତିନ ଭ୍ୱାଳ’’, ‘ନ୍ତିନ ଭ୍ୱାଳ ແେନ୍ ແେନ୍’’, ‘ଭ୍ୱାଳ’’, ‘ଭ୍ୱାଳ ແେନ୍ ແେନ୍’’, ‘ଭ୍ୱାଳ ແେନ୍ ແେନ୍ ແେନ୍’’

In-text representation, all N-Gram feature sets G are merged to remove repetitive generated list $L = [g_1, g_2 \dots g_l]$, and all N-Gram features in list L are numbered, where g_l is the lth N-Gram feature in list L .

In MLP model, TF-IDF algorithm is used to process text into TF-IDF vector $V(MLP)_i = [t_1, t_2 \dots t_l]$, where $V(MLP)_i$ represents the TF-IDF vector of N-gram feature set G_i and t_l represents the TF-IDF value of the lth feature in list L in the T_i text. The formula for calculating t_l value is (1):

$$t_l = \frac{fnum(G_i, g_l)}{fnum(G_i)} \cdot \log \frac{dnum(all)}{dnum(all, g_l) + 1} \quad (1)$$

Among them, $fnum(G_i, g_l)$ represents the number of occurrences of g_l features in N-Gram feature set G_i in list L , $fnum(G_i)$ is the total number of occurrences of all features in N-Gram feature set G_i , and the value of the front part of multiplication sign is Term Frequency (TF). The rear part of the multiplication sign is Inverse Document Frequency (IDF), $dnum(all)$ represents the total number of documents, and $dnum(all, g_l)$ represents the number of documents containing g_l features, and denominator plus 1 prevents the document set from containing g_l features.

MLP classifier uses a fully connected layer and softmax classification layer. The calculation formula between the input layer and the fully connected layer is as follows (2):

$$a_i^{(1)} = f(W^{(1)}V(MLP)_i + b^{(1)}) \quad (2)$$

$V(MLP)_i$ is the input of text T_i in the neural network. $a_i^{(1)}$ is the value of all neurons in the first layer of the neural network after input text T_i ; f is the activation function of the first layer; $W^{(1)}$ is the weight matrix of all neurons in the first layer and the input layer; and $b^{(1)}$ is the bias unit of all neurons in the first layer.

The word vector is used in the FastText model. First, each feature in list L is coded with one-hot. Then, according to the one-hot vector encoded by list L , the sample is transformed into one-hot vector expression $O(FT)_i = [o_1, o_2 \dots o_m]$, where $O(FT)_i$ represents one-hot vector set of sample T_i and o_m represents one-hot vector corresponding to feature x_m in N-Gram feature set G_i . Finally, the one-hot vector set is transformed into a dense vector through the word vector layer, and the corresponding set of word vectors $V(FT)_i = [v_1, v_2 \dots v_m]$ is obtained. $V(FT)_i$ represents the set of word vectors of sample T_i . v_m is the word vector corresponding to o_m in one-hot vector set $O(FT)_i$.

FastText uses the average pooling layer and the softmax classification layer, and the formula between the input layer and the average pooling layer is (3):

$$aver_i = \frac{1}{m} W \cdot \sum_{j=1}^m v_j^i \quad (3)$$

Where $aver_i$ represents the weighted average of all word vectors of the text T_i , the weight is W , m represents the number of text T_i word vectors, and v_j^i is the jth word vector in the word vector set $O(FT)_i$.

B. Sequence model

The sepCNN and Bi-LSTM models used in this paper belong to the sequence model. The original sample can be regarded as the sequence $T_i = [w_1, w_2 \dots w_n]$ of Tibetan syllables or Tibetan vocabulary. w_n is the nth syllable or vocabulary in this text, and it is also the nth feature. T_i can also be regarded as a feature sequence. The feature sequence is transformed into the word vector sequence $V(sequence)_i = [v_1, v_2 \dots v_n]$ by word vector operation. $V(sequence)_i$ is the corresponding word vector sequence of T_i . v_n represents the word vector of w_n feature in T_i after training.

SepCNN is a variant of CNN that mentioned in the conference paper of CVPR (Computer Vision and Pattern Recognition Conference) in 2017 [18]. It decomposes ordinary convolution operation into a depthwise process and a pointwise process. Compared with conventional CNN, the number of parameters of sepCNN is reduced and the training efficiency is improved.

In this paper, sepCNN classifier uses deep separable convolution layer, maximum pooling layer, average pooling layer, and softmax classification layer, in which the input word vector sequence is convoluted, such as formula (4):

$$A_i^{[1]} = g^{[1]}(W^{[1]} * V(\text{sequence})_i + b^{[1]}) \quad (4)$$

Among them, $V(\text{sequence})_i$ is the input of text T_i in the neural network; $A_i^{[1]}$ is the value of all neurons in the first convolution layer after input text T_i ; $g^{[1]}$ is the activation function of the first convolution layer; $W^{[1]}$ is the convolution core used by the first convolution layer; $*$ is the convolution operation; and $b^{[1]}$ is the bias term.

Bi-LSTM classifier uses Bi-direction Long Short-Term Memory layer, fully connected layer and soft Max classification layer. The Bi-direction Long Short-Term Memory layer has one update gate, one forget gate, and one output gate, wherein the formula of update gate is as shown in (5):

$$\Gamma_u^{<n>} = \sigma(W_u[a^{<n-1>}, v^{<n>}] + b_u) \quad (5)$$

Among them, $a^{<n-1>}$ is the activation value of $n-1$ time; $v^{<n>}$ is the input of n time; W_u is the weight matrix of the update gate; b_u is the bias unit of the update gate. The formula of forget gate is as shown in (6):

$$\Gamma_f^{<n>} = \sigma(W_f[a^{<n-1>}, v^{<n>}] + b_f) \quad (6)$$

W_f is the weight matrix of the forget gate and b_f is the bias unit of the forget gate. The formula of output gate is as shown in (7):

$$\Gamma_o^{<n>} = \sigma(W_o[a^{<n-1>}, v^{<n>}] + b_o) \quad (7)$$

W_o is the weight matrix of the output gate and b_o is the bias unit of the output gate. The formula of the memory cell is as shown in (8), (9):

$$\tilde{c}^{<n>} = \tanh(W_c[a^{<n-1>}, v^{<n>}] + b_c) \quad (8)$$

$$c^{<n>} = \Gamma_f^{<n>} * c^{<n-1>} + \Gamma_u^{<n>} * \tilde{c}^{<n>} \quad (9)$$

W_c is the weight matrix of memory cells and b_c is the biased unit of memory cells. The formula of activation value is as shown in (10):

$$a^{<n>} = \Gamma_o^{<n>} * \tanh(c^{<n>}) \quad (10)$$

The activation value of the current neuron is determined by the memory cell value and the output gate. In this paper, Bi-direction time series is used, and its formula is (11):

$$y^{<n>} = g(w_y[\vec{a}^{<n>}, \tilde{a}^{<n>}] + b_y) \quad (11)$$

$\vec{a}^{<n>}$ is the positive activation value of the neural network at time n , and $\tilde{a}^{<n>}$ is the reverse activation value of the neural network at time n .

IV. EXPERIMENTS

A. Experimental Data

Because there is no open Tibetan text categorization corpus, Python is used to implement the crawler system, and

the news corpus is crawled from "China Tibet News Network", with a total of 66 622 articles in 9 categories. The data set was cleaned to remove the text whose length was too short to express the clear meaning, and the final number of texts was 66310. Because the number of texts is different, each type is divided into 80%, 10% and 10% data sets, which are used as the training set, the validation set, and the test set respectively. The specific number is shown in TABLE II.

TABLE II. DATA SET DISTRIBUTION STATISTICS

categories	training	validation	test
<i>government affairs</i>	15000	1875	1875
<i>farming</i>	1648	206	206
<i>legal</i>	15992	1999	1999
<i>scientific education</i>	6880	860	860
<i>Literature and Art</i>	6816	852	852
<i>religion</i>	1784	223	223
<i>medicine</i>	1224	153	153
<i>Historical geography</i>	568	71	71
<i>Eco-tourism</i>	3136	392	392
<i>total</i>	53048	6631	6631

The non-Tibetan characters are cleaned when the text is processed. In the syllable level experiment, the Tibetan syllable-dividing mark is used as the segmentation point, and then the syllable-dividing mark is removed, and the syllables are separated by spaces. In the vocabulary level experiment, the data were segmented using the Tibetan automatic word segmentation system of Tibetan University. After processing, the words were separated by spaces, and the syllables contained in each word are connected by Tibetan syllable-dividing mark.

B. Parameter Settings and Evaluation Metrics

In this experiment, the neural network is constructed using the TensorFlow framework. The batch size is set to 1024; the learning rate is set to 0.001 using Adam optimizer; the loss function uses Multi-class log loss.

In MLP classifier, the number of neurons in the fully connected layer is set to 64, and the activation function uses ReLU.

The average pooling layer in FastText classifier uses GlobalAveragePooling1D.

Four deep separable convolution layers are set in sepCNN, the length of the convolution window is set to 3, the number of output channels of the first two convolution layers is set to 64, and the number of output channels of the last two layers is set to 128.

In Bi-LSTM, the output dimension of Bi-direction Long Short-Term Memory layer is set to 64, and the number of neurons in the fully connected layer is set to 64.

In this experiment, accuracy, precision, recall, and F1-Measure were used as evaluation criteria.

C. Experimental Results and Discussions

TABLE III shows the accuracy of MLP, FastText, sepCNN and Bi-LSTM based on Tibetan syllables or vocabularies in the training set, the validation set, and the test set. It can be found from the comparison table that the accuracy of each model based on vocabulary classification is higher than that based on syllable classification in the test set. This is because Tibetan words are the smallest unit of semantic load rather than syllables.

TABLE III. ACCURACY RESULTS BASED ON SYLLABLE AND VOCABULARY MODELS

models		training (%)	validation (%)	test (%)
<i>MLP</i>	<i>syllable</i>	93.77	84.63	84.15
	<i>vocabulary</i>	94.64	85.07	84.68
<i>FastText</i>	<i>syllable</i>	91.91	86.65	86.25
	<i>vocabulary</i>	91.92	86.58	86.38
<i>sepCNN</i>	<i>syllable</i>	87.09	83.67	83.09
	<i>vocabulary</i>	91.60	83.37	83.56
<i>Bi-LSTM</i>	<i>syllable</i>	88.83	84.77	84.20
	<i>vocabulary</i>	91.98	85.97	85.55

In TABLE III, by comparing the accuracy of the training set, the validation set and the test set, it can be found that the accuracy of the training set is significantly higher than that of the other two sets, which is due to the existence of overfitting phenomenon in training. Some problems can be found in the analysis of samples. Samples vary in length.

TABLE IV. PERFORMANCE COMPARISON OF SYLLABLE-BASED MODELS ON TEST SETS

	government affairs	farming	legal	scientific education	Literature and Art	religion	medicine	Historical geography	Eco-tourism
<i>MLP</i>	<i>precision (%)</i>	90.52	68.66	87.95	71.00	81.03	73.86	89.38	54.84
	<i>recall (%)</i>	97.28	44.66	90.95	74.30	83.22	58.30	66.01	23.94
	<i>F1-Measure(%)</i>	93.78	54.12	89.42	72.61	82.11	65.16	75.94	33.33
<i>FastText</i>	<i>precision (%)</i>	93.91	64.58	90.73	78.01	78.41	74.19	87.80	64.52
	<i>recall (%)</i>	96.21	60.19	91.55	78.37	86.97	61.88	70.59	28.17
	<i>F1-Measure(%)</i>	95.05	62.31	91.14	78.19	82.47	67.48	78.26	39.22
<i>sepCNN</i>	<i>precision (%)</i>	93.79	68.66	86.00	70.90	79.31	73.94	88.18	46.67
	<i>recall (%)</i>	91.04	44.66	91.00	77.33	83.22	54.71	63.40	9.86
	<i>F1-Measure(%)</i>	92.40	54.12	88.43	73.97	81.21	62.89	73.76	70.19
<i>Bi-LSTM</i>	<i>precision (%)</i>	92.59	58.13	87.70	75.03	86.19	63.24	78.57	36.59
	<i>recall (%)</i>	93.92	57.28	91.00	73.72	79.11	71.75	71.90	21.13
	<i>F1-Measure(%)</i>	93.25	57.70	89.32	74.37	82.50	67.23	75.09	26.79

TABLE V. PERFORMANCE COMPARISON OF VOCABULARY-BASED MODELS ON TEST SETS

	government affairs	farming	legal	scientific education	Literature and Art	religion	medicine	Historical geography	Eco-tourism
<i>MLP</i>	<i>precision (%)</i>	93.50	80.28	88.62	70.02	79.43	69.68	85.83	59.57
	<i>recall (%)</i>	96.59	55.34	91.95	75.23	81.10	58.74	71.24	39.44
	<i>F1-Measure(%)</i>	95.02	65.52	90.25	72.53	80.26	63.75	77.86	47.46
<i>FastText</i>	<i>precision (%)</i>	94.62	75.00	89.69	78.02	80.42	71.28	88.35	69.70
	<i>recall (%)</i>	95.68	61.17	93.15	78.02	86.27	60.09	59.48	32.39
	<i>F1-Measure(%)</i>	95.15	67.38	91.39	78.02	83.24	65.21	71.09	44.23
<i>sepCNN</i>	<i>precision (%)</i>	96.52	77.14	90.50	72.03	73.68	62.26	74.65	56.67
	<i>recall (%)</i>	88.64	52.43	90.50	76.98	87.09	59.19	69.28	23.94
	<i>F1-Measure(%)</i>	92.41	62.43	90.50	74.42	79.83	60.69	71.86	33.66
<i>Bi-LSTM</i>	<i>precision (%)</i>	91.07	66.50	91.91	80.08	79.61	67.13	80.00	60.98
	<i>recall (%)</i>	96.27	63.59	90.95	71.05	85.68	65.02	67.97	35.21
	<i>F1-Measure(%)</i>	93.60	65.01	91.43	75.29	82.53	66.06	73.50	44.64

From TABLE II, TABLE IV and TABLE V, all kinds of F1-Measures are closely related to the number of samples it has. In all classification models, the F1-Measures of the category of "historical geography" are the lowest, while the F1 values of the two categories of "government affairs" and "legal" which have the largest number of samples are very high. Comparing their F1-Measures based on the syllable and vocabulary in each model, we can find that in most categories, and the F1-Measures are improved after having finished word segmentation. This is because of the structural characteristics of Tibetan; the use of word segmentation is better than syllable-based classification. Because the accuracy of the Tibetan word segmentation system is not too high, the F1-Measure has not been significantly improved after the existence of some categories of word segmentation.

The shortest sample has only one sentence with four syllables, while the longer sample has thousands of sentences. The distribution of samples is uneven. TABLE II shows that the largest number of legal samples is 19 990, while the smallest number of historical and geographical samples is 710, accounting for only 1/28 of legal samples. The number of samples is relatively small. In this experiment, there are only 60,000 Tibetan News Texts in 9 categories, with an average of 7,000 in one category. It is difficult to compare the number of data sets with those of Chinese and English news classifications. The sample quality is not good; some samples have little information content; the classification of the original corpus is mixed. So, it is easy to have over-fitting problems in training.

TABLE IV and TABLE V shows the detailed precision, recall, and F1-Measure of each model in the test set.

Compared with the MLP model and FastText model in TABLE IV and TABLE V, FastText is superior to the MLP model in accuracy, precision, recall, and F1-Measure. These two models belong to shallow neural networks in structure. The difference is that MLP uses TF-IDF coding while FastText uses word embedding. Word embedding has more advantages than TF-IDF in terms of meaning expression and meaning association. It can be found that the FastText model has a good classification effect, and because of its simple structure, it is also fast in training.

Observation TABLE IV and TABLE V show that compared with other models, the accuracy, recall and F1 values of sepCNN are not very high. Compared with the advantage of convolutional neural networks in image processing, it has no great advantage in text processing. The reason for this may be related to the data samples used in

this experiment, or it may be because only the basic convolutional network is used without improvement.

In TABLE IV and TABLE V, the cyclic neural network model Bi-LSTM performs well, because the Bi-LSTM model can get the context information of words in text very well and is good at long-distance information transmission of text. However, due to its complex structure and the problems of the small amount of data and uneven distribution in the Tibetan text set, it is very easy to produce over-fitting when training the Bi-LSTM model. Moreover, the Bi-LSTM model is difficult to parallelize in training due to its cyclic structure. Therefore, compared with other neural network models, the training time is too long, and the training cost is very high.

V. CONCLUSION AND FUTURE WORK

This research selects the typical models of full-connected neural network, convolution neural network, and cyclic neural network and apply them to Tibetan text categorization. The classification results based on Tibetan syllables and vocabulary were summarized and analyzed. The classification effect based on Tibetan vocabulary is better than that based on Tibetan syllables, and the use of neural networks can better solve the problem of Tibetan text classification. Because there is no open standard classification sample set in Tibetan corpus, the crawler technology is used to collect data in this study. There are some problems in the sample set, such as the small amount of data and the unbalanced distribution of data. After that, we will expand and improve the sample set, and further study the application of neural network in Tibetan information processing, to improve the effect of Tibetan text classification.

ACKNOWLEDGMENT

This paper is supported by The National Natural Science Foundation of China (61751216), the National Team of Computer and Tibetan Information Technology and the Construction of Key Laboratories (Tibetan Finance and Education Instruction [2018] No. 81).

REFERENCES

- [1] M. Sasaki, and K. Kita, "Rule-based text categorization using hierarchical categories," SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218), San Diego, CA, USA, 1998, pp. 2827-2830 vol.3.
- [2] M. E. Maron, and J. L. Kuhns, "On Relevance, Probabilistic Indexing and Information Retrieval," *J. ACM*, vol. 7, no. 3, pp.216-244, Jul. 1960.
- [3] B. Trstenjak, S. Mikac, and D. Donko, "KNN with TF-IDF based Framework for Text Categorization," *Procedia Engineering*, vol. 69, pp.1356-1364, 2014.
- [4] D. Liu, T. Li, and D. Liang, "Incorporating logistic regression to decision-theoretic rough sets for classifications," *International Journal of Approximate Reasoning*, vol. 55, pp.197-210, Jan. 2014.
- [5] L. Demidova, I. Klyueva, Y. Sokolova, N. Stepanov, N. Tyart, "Intellectual Approaches to Improvement of the Classification Decisions Quality on the Base of the SVM Classifier," *Procedia Computer Science*, vol. 103, pp. 222-230, 2017.
- [6] G. Vinodhini, and R. M. Chandrasekaran, "A comparative performance evaluation of neural network based approach for sentiment classification of online reviews," *Journal of King Saud University - Computer and Information Sciences*, vol. 28, pp. 2-12, Jan. 2016.
- [7] G. E. Hinton, "Learning distributed representations of concepts," Eighth Conference of the Cognitive Science Society, 1989, pp. 1-12.
- [8] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv preprint, arXiv:1301.3781, 2013.
- [9] J. Hartmann, J. Huppertz, C. Schamp, and M. Heitmann, "Comparing automated text classification methods," *International Journal of Research in Marketing*, vol. 36, pp. 20-38, Mar. 2019.
- [10] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," arXiv preprint, arXiv:1607.04606, 2016.
- [11] Y. Kim, "Convolutional Neural Networks for Sentence Classification," arXiv preprint, arXiv:1408.5882, 2014.
- [12] M. Schuster, and K. K. Paliwal, "Bidirectional recurrent neural networks," in *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673-2681, Nov. 1997.
- [13] C. Xiang, Y. Weng, X. Zhao, G. Yang, "Automatic Text Classification of Tibetan Web Pages Based on Column," *Journal of Chinese Information Processing*, vol. 25, no. 04, pp. 20-23, 2011.
- [14] H. Jia, "Research on Key Technologies of Tibetan Text Classification Based on KNN Algorithms," *Journal of Northwest University for Nationalities (Natural Science)*, vol. 32, no. 03, pp. 24-29, 2011.
- [15] R. Nobu, Y. Su, Y. Sun, "Design and Implementation of Tibetan Bad Text Recognition System Based on Maximum Entropy Model," *Tibet's Science and Technology*, no. 03, pp. 77-78, 2014.
- [16] N. Qun, H. Jia, "Research and Implementation of Based on the Logistic Regression Model for Tibetan Text Classification," *China Computer & Communication*, no. 05, pp. 70-73, 2018.
- [17] H. Jia, N. Qun, H. Su, L. Ciren, Z. Basang, "Research and Implementation of Tibetan Text Classification Based on SVM," *Electronic Technology & Software Engineering*, no. 09, pp.144-146, 2018.
- [18] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 1800-1807.

BERT with Enhanced Layer for Assistant Diagnosis Based on Chinese Obstetric EMRs

Kunli Zhang^{*†}, Chuang Liu^{*}, Xuemin Duan^{*†}, Lijuan Zhou^{*}, Yueshu Zhao^{‡†} and Hongying Zan^{*†}

^{*}School of Information Engineering

Zhengzhou University

[†]Peng Cheng Laboratory

[‡]The Third Affiliated Hospital of Zhengzhou University

Email : ieklzhang@zzu.edu.cn ; 214674227@qq.com ; xueminduan@stu.zzu.edu.cn ;

ieljzhou@zzu.edu.cn ; zyswr@163.com ; iehyzan@zzu.edu.cn

Abstract—This paper proposes a novel method based on the language representation model called BERT (Bidirectional Encoder Representations from Transformers) for Obstetric assistant diagnosis on Chinese obstetric EMRs (Electronic Medical Records). To aggregate more information for final output, an enhanced layer is augmented to the BERT model. In particular, the enhanced layer in this paper is constructed based on strategy 1(A strategy) and/or strategy 2(A-AP strategy). The proposed method is evaluated on two datasets including Chinese Obstetric EMRs dataset and Arxiv Academic Paper Dataset (AAPD). The experimental results show that the proposed method based on BERT improves the F1 value by 19.58% and 2.71% over the state-of-the-art methods, and the proposed method based on BERT and the enhanced layer by strategy 2 improves the F1 value by 0.7% and 0.3% (strategy 1 improves the F1 value by 0.68% and 0.1%) over the method without adding enhanced layer respectively on Obstetric EMRs dataset and AAPD dataset.

Keywords-EMRs; assistant diagnosis; BERT; the enhanced layer;

I. INTRODUCTION

EMRs(Electronic Medical Records) are detailed records of medical activities by medical personnel. The most important form of EMRs is free text data. With the development of medical informatization, hospitals have accumulated massive amounts of EMRs. These EMRs contain a lot of medical knowledge and patients health information. It is one of the most important tasks in the medical field to use NLP(Natural Language Processing) technology for assistant diagnosis based on EMRs. Since family planning was issued as one of the fundamental state policies in China, the policy of late marriage and late childbearing has brought many benefits. However, it has also led to an increase in the proportion of older pregnant women over 35 years of age [1]. After the implementation of China's Universal Two-child Policy in 2016, the proportion of older pregnant women will has become greater. The incidence of dystocia, fetal malformations and complications among older pregnant women is higher than that of normal pregnant women, it will be a great challenge for obstetrics in medical institutions to solve this problem. EMRs not only records the patient's complaint, physical examination, auxiliary examination and other information, but also records the doctor's initial diagnosis, diagnosis based on differential diagnosis and treatment plan.

Usually, ad-mission diagnosis includes normal diagnosis, pathological diagnosis and description of complications, rather than a single diagnosis. In this paper, we treat the obstetric diagnostic task of EMRs as text multi-label classification task. To solve this problem [2] proposed the BERT model, which not only improves greatly on multiple data sets of different tasks, but also adapts to different tasks only by fine-tuning the pre-training version of BERT. On the basis of real EMRs, through screening and processing the original medical records, we transform tasks into multi-label text classification tasks. The contribution of this paper are as follows.

- To the best of our knowledge, BERT was firstly applied to the auxiliary diagnosis of Chinese EMRs.
- An enhanced layer was augmented to the BERT model based on two strategies for further improvement of the diagnosis effect.
- The enhanced layer works equally well on other domains of dataset.

Experiments on EMRs datasets and public datasets of text multi-label classification show that the results of Bert with enhanced layer model have been improved, which demonstrates the effectiveness and generality of the enhanced layer.

II. RELATED WORKS

Traditional multi-label classification mainly transforms multi-label classification tasks into multi-classification problem. In the neural network learning, changing the deep learning model and the loss function in order to improve the effect of multi-label classification [3]–[10]. Pre-training technology has become the corner-stone of NLP task. Pre-training technology can effectively improve the performance of NLP task results. Among them, word embedding technology [11], [12] has also made considerable progress, and has become a standard technology in different tasks. However, word embedding technology also has some drawbacks, such as the inability to distinguish polysemy. For example, the inaccuracy of word segmentation and the inability to distinguish polysemy will affect the quality of word or word encoding. In recent studies, pre-trained language models can effectively address the above shortcomings, such as ELMO [13], OpenAI GPT [14], and BERT [2]. In addition to using word embedding

technology, these language models also use different deep learning coders for context encoding, in which ELMO uses Bi-LSTM [15], GPT uses one-way Transformer [16]. Coder, BERT uses bidirectional Transformer encoder. The above language model has obvious improvement on different NLP tasks, and can be applied to different NLP tasks only by fine-tuning the output of different NLP tasks. In the diagnosis of Obstetric based on EMRs, [17] proposed a multi-label classification method to study the problem of assistant diagnosis based on Chinese obstetric EMR. [18] used vector stitching method to fuse the numerical characteristics of EMRs for experiments, which improved the effect of assistant diagnosis. The traditional multi-label classification model is used in both tasks, and the experimental dataset is about 10,000. On the basis of expanding the dataset, this paper will use the pre-training model to study the problem of assistant diagnosis based on Chinese obstetric EMRs.

III. MODEL AND METHOD

This section presents the details of the proposed model. Firstly, we will give the overall structure of the model, which is divided into three parts: encoding layer, enhanced layer and output layer. Then the three parts are detailed in the following.

A. Overview

In this paper, we treat the assistant diagnosis task of obstetric EMRs as multi-label text classification task. Let $\chi = R^d$ be a d dimensional instance space, $\mathbf{y} = y_1, y_2, \dots, y_q$ is a set of q categories. Given a training set, where each instance $T = (x_1, Y_1), (x_2, Y_2), \dots, (x_m, Y_m)$ ($x_i \in \chi, Y_i \subseteq \mathbf{y}$) is a d dimensional feature vector, the goal of multi-label learning is to learn a multi-label classifier which satisfies some evaluation criteria. The model consists of Encoding layer, Enhanced layer and Output layer. The encoding layer uses BERT to obtain all the hidden layer representations of input sequences and hidden representation [C] for classification tasks (represented by C shown in Figure1). The design goal of enhanced layer is to further enhance the [C] hidden layer representations to cover as much sequence information as possible. Detailed content is described in enhanced layer. The main structure of the model is shown in Figure1.

B. BERT

BERT is a encoding structure composed of bidirectional transformer model, in which the transformer model is the attention encoding model proposed by [18]. Transformer uses a multi-head attention mechanism and each head calculates the attention weight independently. Then the model splices the results of each head, so the multi-head attention mechanism can be represented at different levels of the sequence. Because location information cannot be obtained in simple attention calculation, Transformer uses special position vector encoding for sequence location information. Due to the performance advantages of transformer model, it has recently become one of the most important

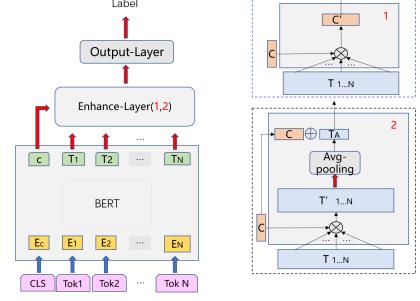


Figure 1. The left side of the figure is the framework of the proposed model. After sequence input, it first generates one additional output representation which is represented by C shown in Figure1 through BERT model, then generates a better hidden layer representation by enhanced layer. The right side of the figure illustrates two strategies of enhanced layer. Among them, red number 1 represents attention strategy (namely, A strategy) and red number 2 represents Attention-average pooling strategy (namely, A-AP strategy).

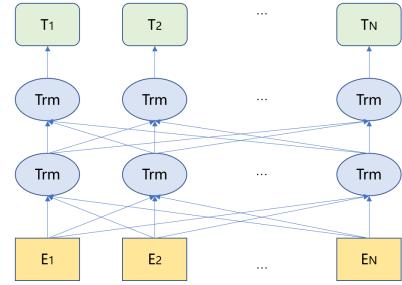


Figure 2. BERT uses a bidirectional transformer structure (represented by a blue elliptical Trm), and BERT encoding connects information in both directions of the context. Figure adapted from [2].

models in NLP field, and has been widely used in various sub-tasks, so the details of transformer model are not elaborated in this paper. In BERT, the input of the model can be a single sequence or a sequence pair. In this paper, we regard EMRs as single sequence to do experiments. In the Chinese task, a sequence consists of three parts: embedding representation of each independent Chinese character, location embedding and input marking. In the output of BERT, in addition to the classified representation embedding [C] mentioned in the previous section, all the representations of the sequence are also recorded as h_i , where i is the position of each word in the input sequence x , all the sequence representations are defined as H_m , and m represents the number of words in the text. In order to obtain better classification representation for final classification, the self-attention mechanism is introduced to further enhance the output representation of encoding. Because BERT has been encoded by several transformer blocks, each position of each layer sequence is encoded by transformer blocks, we can see the construct of BERT in Figure2. But it is noteworthy that when the original BERT model is output, only one position encoding information is used as the input of the latter classifier, which is the position of [C] in Figure1. Although this position is a

representation of the sequence, other positions in the final output still contain the information of the sequence which can also be used to enhance the representation of the output vector. So this paper designs enhanced layer after BERT to enhance the representation of output. Details of enhanced layer are discussed below.

C. Enhanced Layer

In this paper, we proposed two enhanced layer strategies. Two strategies are used to enhance the classification representation of the original BERT $h[C]$. Details of the two strategies are described below.

1) *Strategy 1(namely, A strategy)*: In the strategy 1, the enhanced layer uses the information on the position of [C] after the last encoding to calculate the attention of all the position information in this layer sequence. In this case, the output vector of [C] position not only contains the whole sequence information of the upper layer, but also interacts with the information of all positions of this layer. As shown in the upper right of Figure1, the hidden layer representation $h[C]$ and the whole sequence representation T_n are used to calculate the attention equation (1) (2). After using equation (3), the hidden layer representation containing all sequence information is obtained which is written as $h[C]$, and then $h[C]$ is used as the input of the classifier.

$$e = v_a^T \tan(W_a h_{[c]} + U_t T_n) \quad (1)$$

$$a = \text{Softmax}(e) \quad (2)$$

$$h'_{[c]} = a T_n \quad (3)$$

2) *Strategy 2(namely, A-AP strategy)*: In the strategy 2, all the sequences will be generated. The information is computed with the information at [C] position, and the enhanced representation of all sequences is obtained. Inspired by [19] and [20], simple pooling operation extracts the effective information of the sequence. Therefore, this paper adopts the pooling operation to extract the information, and then output the information after splicing with the original [C]. This paper uses Concat Attention [21] mechanism to make an attention interaction between the representation $h[C]$ of [C] and the representation of all sequences. As shown in the lower right of Figure1, we first exchange the calculation order of hidden layer representation $h[C]$ and the representation of T_n of all sequences, then use the same attention equation as strategy 1 to calculate, and use equation (4) to get the new representation of T_n of all sequences. We use average pooling (5) operation to obtain a new representation of T_{avg} . The new representation T_{avg} and original classification representation $h[C]$ are joined together (6) as input of classifier.

$$T'_n = \sum_{i=1}^m a h_{[c]} \quad (4)$$

$$T_{avg} = \text{Avg - pooling}(T'_n) \quad (5)$$

$$h'_{[c]} = \text{Concat}(h_{[c]}, T_{avg}) \quad (6)$$

Table I
SUMMARY OF DATASETS. TOTAL SAMPLES, LABEL SETS DENOTE THE TOTAL NUMBER OF SAMPLES AND LABELS, RESPECTIVELY. LABELS/SAMPLE IS THE AVERAGE NUMBER OF LABELS PER SAMPLE.

	Total Samples	Label Sets	Labels/Sample
Obstetrics EMRs	24,339	73	2.49
AAPD	55,840	54	2.41

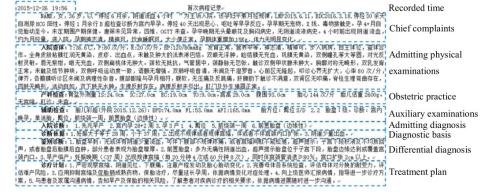


Figure 3. The example of the first course of disease record. Figure adapted from [12].

It should be noted that the classification representation $h[C]$ obtained in Strategy 1 is a d-by-T matrix and T is the size of batch size. In strategy 2, the classification representation $h[C]$ is a 2d-by-T matrix.

D. Output Layer

We optimize the entire model end-to-end, with the additional softmax classifier parameters $W \in R^{k \times H}$ where H is the dimension of the hidden state vectors and K is the number of classes. In this paper, K is also the number of diseases to be classified. The sigmoid function is used to classify each label into 0 or 1, where the threshold is set to 0.5. And we minimize the BCEWithLogitsLoss for multi-label tasks.

IV. EXPERIMENT

A. Datasets

In this paper, the experiment was carried out on the Obstetrics EMRs and the open multi-label data set Arxiv Academic Paper Dataset (AAPD) [10]. The scale information of the data set is shown in Table I.

1) *AAPD Dataset*: Arxiv Academic Paper Dataset (AAPD) is a large multi-label text categorization dataset. Totally, it includes 55,480 abstracts of papers collected from computer science websites. An academic paper may contain a variety of topics. There are 54 topics in this dataset. The task of data sets is to predict the corresponding topics according to the abstracts of the papers.

2) *Chinese Obstetric EMRs Dataset*: The Chinese obstetric EMRs datasets used in this paper is randomly extracted from 15 hospitals. A total of 24,339 EMRs were pre-processed. The experiment in this paper mainly uses the information of chief complaint, admission examination, obstetric examination and auxiliary examination in the first course record to give a preliminary diagnosis. Therefore, the structure of the first course is mainly analyzed. The first course record sample is shown in Figure3.

Table II
EXPERIMENTAL RESULTS OF OBSTETRICS EMRS. A REPRESENT A STRATEGY, A-AP REPRESENT A-AP STRATEGY.

Model	F1(%)	Average Precision(%)	Hamming Loss	One Error
SGM	60.00	39.00	0.0200	0.0630
BERT	79.58	84.97	0.0132	0.0961
BERT+A	80.26	85.42	0.0129	0.0863
BERT+A-AP	80.28	85.74	0.0129	0.0891

Table III
EXPERIMENTAL RESULTS OF AAPD. A REPRESENT A STRATEGY, A-AP REPRESENT A-AP STRATEGY.

Model	F1(%)	Average Precision(%)	Hamming Loss	One Error
CNN	66.40	-	0.0256	-
CNN-RNN	66.40	-	0.0278	-
SGM	71.00	-	0.0245	-
BERT[24]	73.40	-	-	-
BERT	73.71	79.89	0.0227	0.022
BERT+A	73.81	79.51	0.0225	0.023
BERT+A-AP	74.01	79.74	0.0225	0.024

Table IV
COMPARISON OF OBSTETRICS EMRS OF DIFFERENT POOLING MECHANISMS IN STRATEGY 2(A-AP STRATEGY). A-AP REPRESENT A-AP STRATEGY.

Pooling	F1(%)	Average Precision(%)	Hamming Loss	One Error
Avg-pooling(A-AP)	80.28	85.74	0.0129	0.0891
Max-pooling	80.07	85.69	0.0130	0.0933
(Max+Avg)-pooling	80.16	85.55	0.0130	0.0941

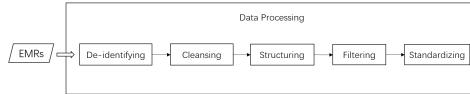


Figure 4. Anonymous identification process

B. Experimental Setup

The collected electronic medical records are pretreated by de-identifying [22], data cleaning, structuring, data filtering and standardization of diagnostic labels. The process is shown in Figure4. Data filtering is the filtering of repeated and similar information contained in the first course of disease, not the decisive factor of filtering diagnosis. It filters by calculating sentence similarity, and only retains personality information. The standardization of diagnostic labels uses different descriptions of the same diagnosis for different doctors, and standardizes the diagnostic results according to ICD10 under the guidance of doctors. In this paper, BERT-base-Chinese is chosen as the language model version of electronic medical record diagnosis in obstetrics and gynecology, and BERT-base is used as the model language model version of AAPD dataset. The parameters are set by default. The main parameters are: hidden_size 768, max_position_embedding 512, num_attention_heads 12, num_hidden_layers 12, maximum input length 512, optimizer Adam [23], learning rate 5e-5, batch_size 2, training epoch 20. We run the rest of the experiments on a GTX 1080 GPU.

C. Results

According to the distribution of diagnostic results, 21,905 of them were used as training set and 2,434 as

test set. The experimental results of EMRs diagnosis in obstetrics and gynecology are shown in Table II. Four evaluation indexes [24] (hamming loss, F1-micro, One-error, Average precision) are used to test the effects of BERT and the model with enhanced layer strategy. In the experiment, the Sequence Generation Model (SGM) proposed by [10] is used as the experimental contrast model in this paper. The AAPD dataset used in this experiment is also from SGM paper [10]. BERT represents the result of using the BERT-base-Chinese version alone which is one of the BERT models by [2], and BERT + represents the author's introduction of a mechanism to enhance the original BERT model after the original BERT. The experimental results of AAPD dataset are shown in Table III. The evaluation indicators also use the above four indicators. The results of CNN, CNN-RNN and SGM are from [10], which are shown in the first to third rows of Table III respectively. The result of BERT are from [25], which are shown in the four row of Table III respectively. In this experiment, BERT represents the result of using the BERT-base version alone, and BERT + also represents the result of using the corresponding model layer to enhance BERT.

D. Analysis By Different Pooling Mechanisms in strategy 2

Strategy 1(strategy A) plays an important role in enhancing classification representation through sequence information. The experimental results further prove its effectiveness. But in strategy 2(A-AP strategy), why choose average pooling mechanism instead of other pooling mechanism? This paper then gives a more detailed experimental comparison on Chinese Obstetric EMRs, through the experimental results to analyze the reasons. In Table

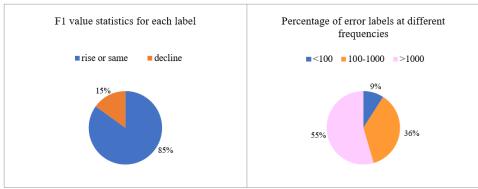


Figure 5. F1 value statistics of each diagnostic label for A strategy. The left chart shows that the F1 value of 85% tags increases or remains unchanged after adding A strategy, while only 15% of tags have a F1 value decline. The right chart shows that among all the wrong labels, labels with frequencies higher than 1000 account for 55% of the total, labels with frequencies between 100 and 1000 account for 36% of the total, while labels with frequencies lower than 100 account for only 9%.

IV, the author compares the pooling mechanism used in strategy 2(A-AP strategy) with other pooling mechanisms on Chinese Obstetric EMRs.

The experimental results show that the average pooling mechanism used in strategy 2(A-AP strategy) is the best based on all the indicators. Besides, the other indicators of maximum pooling are better than the stitching pooling except that the F1 value is lower. This is because the average pooling mechanism is used to get the representation that contains all sequence information. In the maximum pooling mechanism, only the most important representation is obtained. Result from the rich information in gynecological electronic medical records, there are more than one information conducive to diagnosis, and only using the maximum pooling mechanism will cause a certain degree of information loss. In stitching pooling, the use of both pooling methods will get more abundant information theoretically, but due to the existence of maximum pooling, it will still interfere with the final tag prediction to a certain extent. Therefore, the average pooling mechanism is adopted in strategy 2(A-AP strategy) rather than others.

E. Analysis By Different Strategy

In order to analyze the effects of the two enhanced layer strategies proposed in this paper more intuitively, we have made fine-grained statistics on the F1 values of each tag. The experiment found that after adding strategy 1(A strategy), the F1 values of 62 tags in all 73 tags were increased or unchanged, and only 11 Tags showed a decrease. The author further counted the wrong labels, and found that the F1 value of only one label decreased in labels with frequency less than 100. Figure5 is a visual representation of the results. With the addition of strategy 2(A-AP strategy), the F1 value of 34 Tags was increased in all 73 tags, the F1 value of 27 Tags remained unchanged, and the F1 value of only 12 Tags decreased. Among them, none of the labels with frequencies less than 100 has a F1 value decrease. Figure6 is a visual representation of the results. The above analysis shows that the strategy 1(A strategy) and strategy 2(A-AP strategy) in enhanced layer can increase the F1 value of labels by 84% and 85% respectively. According to the frequency of labels appearing, the experimental results show that the two strategies are more effective for low-frequency labels. This also shows that in the case of sufficient data size, BERT

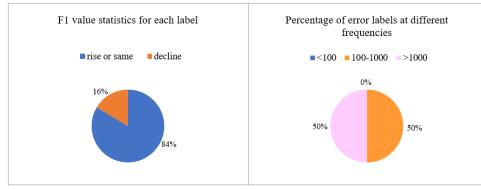


Figure 6. F1 value statistics of each diagnostic label for A-AP strategy. The left chart shows that the F1 value of 84% tags increases or remains unchanged after adding A-AP strategy, while only 16% of tags have a F1 value decline. The right chart shows that among all the wrong labels, labels with frequencies higher than 1000 account for 50% of the total, labels with frequencies between 100 and 1000 account for 50% of the total, while no labels with frequencies lower than 100.

itself has been excellent enough, but in the data set with balanced label distribution, the effect of low-frequency label still has great room for improvement.

V. CONCLUSION

In this experiment, we regard the diagnosis of electronic medical records of gynecology and obstetrics as the task of text multi-label classification. We propose two strategies based on the current advanced language model (BERT) and verify its effectiveness. Of course, in the electronic medical records of obstetrics, besides the existence of information in the form of text, the inspection indicators are also important diagnostic basis. In the next step, the introduction of more domain knowledge and information of inspection indicators deserves attention too. On the other hand, even fine-tuning basic version of BERT-base model will bring a huge amount of computation cost, which limits the deployment and utilization of language models. Therefore, in future work, how to reduce the computation cost of the language models without affecting their performance is equally important.

REFERENCES

- [1] Y. Z. Yang Y.L., "Effect of older pregnancy on maternal and fetal outcomes," *Chin J Obstet Emerg (Electronic Edition)*, pp. 129–135, 2016.
- [2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.
- [3] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1338–1351, 2006.
- [4] J. Nam, J. Kim, E. L. Mencía, I. Gurevych, and J. Fürnkranz, "Large-scale multi-label text classification revisiting neural networks," in *Joint european conference on machine learning and knowledge discovery in databases*. Springer, 2014, pp. 437–452.

- [5] F. Benites and E. Sapozhnikova, “Haram: a hierarchical aram neural network for large-scale text classification,” in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. IEEE, 2015, pp. 847–854.
- [6] G. Kurata, B. Xiang, and B. Zhou, “Improved neural network-based multi-label classification with better initialization leveraging label co-occurrence,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 521–526.
- [7] G. Chen, D. Ye, Z. Xing, J. Chen, and E. Cambria, “En-semble application of convolutional and recurrent neural networks for multi-label text categorization,” in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 2377–2383.
- [8] S. Baker and A. Korhonen, “Initializing neural networks for hierarchical multi-label text classification,” in *BioNLP 2017*, 2017, pp. 307–315.
- [9] S. Ma, X. Sun, Y. Wang, and J. Lin, “Bag-of-words as target for neural machine translation,” *arXiv preprint arXiv:1805.04871*, 2018.
- [10] P. Yang, X. Sun, W. Li, S. Ma, W. Wu, and H. Wang, “Sgm: sequence generation model for multi-label classification,” *arXiv preprint arXiv:1806.04822*, 2018.
- [11] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.
- [12] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in neural information processing systems*, 2013, pp. 3111–3119.
- [13] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” *arXiv preprint arXiv:1802.05365*, 2018.
- [14] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving language understanding with unsupervised learning,” Technical report, OpenAI, Tech. Rep., 2018.
- [15] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [17] K. Zhang, H. Ma, Y. Zhao, H. Zan, and L. Zhuang, “The comparative experimental study of multilabel classification for diagnosis assistant based on chinese obstetric emrs,” *Journal of healthcare engineering*, vol. 2018, 2018.
- [18] Y. Z. H. Z. L. Z. Hongchao Ma, Kunli Zhang, “The study of multi-label assistant diagnosis of obstetrics based on feature fusion,” *Journal of Chinese Information Processing*, pp. 128–136, 2018.
- [19] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, “Bag of tricks for efficient text classification,” *arXiv preprint arXiv:1607.01759*, 2016.
- [20] D. Shen, G. Wang, W. Wang, M. R. Min, Q. Su, Y. Zhang, C. Li, R. Henao, and L. Carin, “Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms,” *arXiv preprint arXiv:1805.09843*, 2018.
- [21] T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kočiský, and P. Blunsom, “Reasoning about entailment with neural attention,” *arXiv preprint arXiv:1509.06664*, 2015.
- [22] Y.-S. Zhao, K.-L. Zhang, H.-C. Ma, and K. Li, “Leveraging text skeleton for de-identification of electronic medical records,” *BMC medical informatics and decision making*, vol. 18, no. 1, p. 18, 2018.
- [23] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [24] M. L. Zhang and Z. H. Zhou, “Ml-knn: A lazy learning approach to multi-label learning,” *Pattern Recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.
- [25] A. Adhikari, A. Ram, R. Tang, and J. Lin, “Docbert: Bert for document classification,” *arXiv preprint arXiv:1904.08398*, 2019.

A Systematic Investigation of Neural Models for Chinese Implicit Discourse Relationship Recognition

Dejian Li, Man Lan, Yuanbin Wu
*School of Computer Science and Technology
 East China Normal University, Shanghai, P.R.China
 51194506071@stu.ecnu.edu.cn; mlan,ybwu@cs.ecnu.edu.cn*

Abstract—The Chinese implicit discourse relationship recognition is more challenging than English due to the lack of discourse connectives and high frequency in the text. So far, there is no systematical investigation into the neural components for Chinese implicit discourse relationship. To fill this gap, in this work we present a component-based neural framework to systematically study the Chinese implicit discourse relationship. Experimental results showed that our proposed neural Chinese implicit discourse parser achieves the SOTA performance in CoNLL-2016 corpus.

Keywords-deep learning; Chinese implicit discourse relation recognition; word embedding;

I. INTRODUCTION

Implicit discourse relationship recognition aims to detect the semantic logic relationship (e.g., *Contrast*, *Conjunction*) between consecutive textual units (e.g., clauses, sentences or paragraphs), which is the main challenge of discourse parsing and benefits many downstream NLP tasks such as Sentiment Analysis [1], Machine Translation [2] and Summarization [3], etc. This task is quite challenging due to two reasons: there is no explicit discourse connective (e.g., *because*, *however*) between textual units (i.e., arguments denoted as Arg1 and Arg2) and implicit discourse relation often occurs in text. For example, almost 40% of the sentences in Penn Discourse Treebank (PDTB)[4] held implicit discourse relations and over 65% in Chinese Discourse Treebank (CDTB)[5].

From the linguistics perspective aspect, the annotation of Chinese discourse relationship differs quite a lot from that in English. Firstly, PDTB has a 3-level hierarchy of multiple relation senses but CDTB has 10 relation senses without hierarchy (as listed in Table I). Secondly, in PDTB Arg2 is always the text to which the connective is syntactically bound, but in CDTB the text order is dependent on the relation sense rather than the discourse connective. As specified in Table I, in *Causation* relation, the argument of effect is always annotated as Arg2 no matter the discourse connective is “因为 (because)” or “所以 (thus)”. Thirdly, in PDTB the discourse relations are annotated within one paragraph while in CDTB the implicit relation can be hold across paragraphs. Fourthly, since the sentences in Chinese are often short (may share the same subject), leading to a large proportion of *Conjunction* relations in Chinese.

Table I
 THE ORDER OF ARGUMENTS FOR EACH SENSE OF DISCOURSE RELATIONS IN CDTB.

Sense	Arg1	Arg2
Alternative		或者/or
Causation	因为/because	所以/so
Conditional	如果/if	就/then
Conjunction		而且/and
Contrast	虽然/although	但是/but
Expansion	综上所述/in conclusion	例如/for example
Progression	不仅/not only	还/but also
Progression	通过/through	还/-
Restatement		换言之/in other words
Temporal	在...之后/text order	在...之前/text order

Recent studies on Chinese implicit discourse relationship adopted deep learning methods. [6] first examined several unsupervised word representations (e.g., one-hot word pair, Brown word clusters and simple word embedding) and confirmed the effectiveness of word embeddings. Later, [7] and [8] explored neural models by adopting word2vec embedding and element-wise *pooling* functions (i.e., *max*, *sum* and *mean*) for sentence representation but they neglected the relationship interaction between two arguments. Furthermore, [8] and [9] used an self-attention BiLSTM to derive sentence representation and demonstrated that modeling two arguments as a joint sequence outperforms previous word order-agnostic approaches.

However, the studies described above leave two open questions. First, several recent word embeddings (e.g., GloVe, ELMo, BERT) have been reported supreme performance in many NLP tasks. We would like to examine their performance for Chinese implicit discourse recognition. Second, the discourse relationship between two arguments is supposed to be more complicated than simple concatenation or self-attention operations on two arguments. We state that the discourse relation between arguments is represented by the interaction between two arguments rather than only simple or separate operations of two arguments.

To address these questions, we present a systematic investigation work to deeply analyze the influence of neural models. Our main contributions are summarized as follows.

- To our knowledge, this is the first framework to systematically investigate neural models for Chinese implicit discourse relationship recogni-

- tion. We present a component-based deep learning architecture, which consists of four independent components and each of them have multiple implementations.
- Extensive experiments in benchmark CoNLL-2016 corpus are conducted to demonstrate the efficacy and effectiveness of our model. Our proposed neural model outperforms the state-of-the-art models by average 1% in accuracy.

II. RELATED WORK

Discourse relationship recognition has attracted a lot of research interests in these years. The recognition of explicit discourse relationship reaches 93% accuracy only by using discourse connectives [10], but the performance of implicit discourse relationship recognition is always poor due to the lack of discourse connectives, which is the bottleneck of the whole discourse parser. Earlier Researchers adopted traditional NLP methods to design and extract complex features with expert knowledge. [11] adopted an aggregated approach to word pairs and [12] employed Brown word clusters. These methods perform badly in generalization.

With the development of deep learning in NLP, researchers began to use the deep learning method to recognize implicit discourse relationship. For example, [6] first compared different unsupervised word representations including standard one-hot word pair representations, low-dimensional representations based on Brown clusters and word embedding. They demonstrated the effectiveness of the word embedding. The studies using deep learning methods are divided into two lines in general. One research line is to learn from explicit discourse relationship or other languages. [13], [14], [15] tried to expand the implicit training dataset with the help of discourse connectives. In order to make full use of the connectives in explicit data, [16] used connective-based word representations and [17] learned discourse-specific word embedding from massive explicit data. [18] presented their implicit network to learn from another neural network which has access to connectives. Unlike those above, [19] used bilingually-constrained synthetic implicit data for implicit discourse relation recognition. The other line focuses on the expression of words and the structure of the model. For example, [20], [21] used word2vec word embedding and Convolutional Neural Network (CNN) to determine the senses. [22] used CNN to model argument pairs with GloVe word embedding and multi-task learning system. [23] combined the word2vec and their proposed event embedding. [24] combined the context information into word embedding which is context-aware character-enhanced embeddings. Regarding to model structure, [25], [26] adopted gated network to calculate the relevance score between two arguments. [27] employed new network structure TreeLSTM to model the sentences.

However, all above studies focused on English corpus and there is not much studies on Chinese corpus. [7], [8] explored feedforward and LSTM for this task. [9] used BiLSTM to model the sentences. To alleviate the shortage of labeled data, [19] designed a multi-task neural network model to use their bilingually-constrained synthetic implicit data as additional data.

III. CHINESE IMPLICIT DISCOURSE RELATIONSHIP PARSER

We present a component-based neural framework for Chinese implicit discourse relationship recognition, consisting of four independent components. Figure 1 depicts the architecture of Chinese implicit discourse relationship parser.

A. Word Embedding Layer

Word embedding is the first and crucial step in deep learning framework, which transforms the natural language into word vector as the input of the neural network. To do so, we convert each word w into a word vector $\mathbf{x} \in \mathbb{R}^{d_w}$, where the d_w is the dimension of the word vector. Let \mathbf{x}_i^1 (\mathbf{x}_i^2) be the i -th word vector in Arg-1(Arg-2), then the two discourse arguments are represented as:

$$\text{Arg-1} : [\mathbf{x}_1^1, \mathbf{x}_2^1, \dots, \mathbf{x}_{L_1}^1] \quad (1)$$

$$\text{Arg-2} : [\mathbf{x}_1^2, \mathbf{x}_2^2, \dots, \mathbf{x}_{L_2}^2] \quad (2)$$

where Arg-1(Arg-2) has $L_1(L_2)$ words.

Generally, the word embeddings are pre-trained on large corpus and supposed to contain latent semantic and syntactic information. In recent years several supreme word embeddings have been presented by researchers. To examine their different effectiveness in word conversion, we choose two types of pre-trained word vector models, i.e., context-free models and contextual models.

Context-free models generate a word embedding representation for each word in the vocabulary, without regard to the context of this word in specific arguments. Word2vec [28] and GloVe [29] are two widely used context-free models. Word2vec uses local text controlled by small window size from large corpus to train the word vector. While GloVe trains on aggregated global word co-occurrence statistics from the corpus.

Contextual models generate word representation for each word based on its context words in the sentence. Usually contextual models aims to obtain language model rather than word embedding. For specific sentence, it gets contextual word representation base on language model. Here we choose ELMo and BERT models as follows.

ELMo [30] is a deep contextualized word representation, which is learned as the internal states of a deep bidirectional language model (biLM).

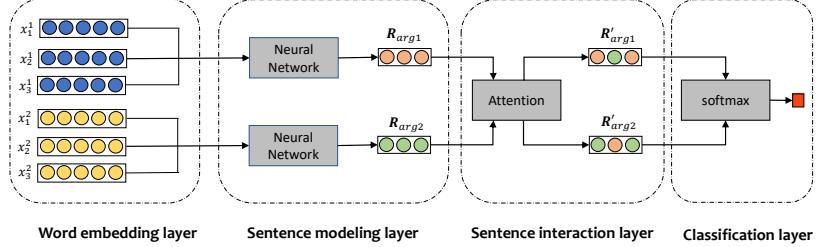


Figure 1. Architecture of our implicit discourse relationship parser system.

BERT¹ means Bidirectional Encoder Representations from Transformers, which is learned from unlabeled text by jointly conditioning on both left and right context in all layers [31].

The two above contextual models are different in some aspects. Firstly, ELMo uses LSTM to encode word while BERT uses transformers. Besides, they have different bi-direction implementations. In ELMo, the forward and the backward direction are simply aggregated and the bidirectional is separate from training. But the bidirectional of the BERT is integrated into the training process.

B. Sentence Modeling Layer

Each argument is transformed into a word vector matrix as shown in Formula (1) and (2) from the word embedding layer. To achieve the semantic representation for each argument, we adopt three sentence modeling methods, i.e., Long Short Term Memory (LSTM)[32], Bi-directional Long Short Term Memory (BiLSTM)[33] and Convolutional Neural Network (CNN).

Given the two argument representations as shown in Formula (1) and (2), the LSTM computes the state sequence $[\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_L]$ for each time step i using the following formulas:

$$\mathbf{i}_i = \sigma(\mathbf{W}_i[\mathbf{x}_i, \mathbf{h}_{i-1}] + \mathbf{b}_i) \quad (3)$$

$$\mathbf{f}_i = \sigma(\mathbf{W}_f[\mathbf{x}_i, \mathbf{h}_{i-1}] + \mathbf{b}_f) \quad (4)$$

$$\mathbf{o}_i = \sigma(\mathbf{W}_o[\mathbf{x}_i, \mathbf{h}_{i-1}] + \mathbf{b}_o) \quad (5)$$

$$\tilde{\mathbf{c}}_i = \tanh(\mathbf{W}_c[\mathbf{x}_i, \mathbf{h}_{i-1}] + \mathbf{b}_c) \quad (6)$$

$$\mathbf{c}_i = \mathbf{i}_i \odot \tilde{\mathbf{c}}_i + \mathbf{f}_i \odot \mathbf{c}_{i-1} \quad (7)$$

$$\mathbf{h}_i = \mathbf{o}_i \odot \tanh(\mathbf{c}_i) \quad (8)$$

where σ denotes the *sigmoid* function and \odot denotes element-wise multiplication.

Unlike LSTM using information only from past, BiLSTM gets the information from both past and future directions. At each position i of the sequence, we obtain two states $\overrightarrow{\mathbf{h}}_i$ and $\overleftarrow{\mathbf{h}}_i$, where $\overrightarrow{\mathbf{h}}_i, \overleftarrow{\mathbf{h}}_i \in \mathbb{R}^{d_h}$. Then we concatenate them to get the intermediate state, i.e. $\mathbf{h}_i = [\overrightarrow{\mathbf{h}}_i, \overleftarrow{\mathbf{h}}_i]$. After that, we sum up the

¹<https://github.com/google-research/BERT/#pre-trained-models>

states in sequence $[\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_L]$ to get the representations of *Arg-1* and *Arg-2* as follows:

$$\mathbf{R}_{\text{Arg}_1} = \sum_{i=1}^{L_1} \mathbf{h}_i^1 \quad (9)$$

$$\mathbf{R}_{\text{Arg}_2} = \sum_{i=1}^{L_2} \mathbf{h}_i^2 \quad (10)$$

In CNN model, we use $\mathbf{Arg}[i : j]$ to represent the sub-matrix of \mathbf{Arg} from row i to row j . A convolution involves a filter $\mathbf{w} \in \mathbb{R}^{h \times d}$ (h is the height of filter and d is the dimensionality of the word vector). The output sequence \mathbf{o}_i of the convolution operator is obtained by repeatedly applying the filter on sub-matrices of \mathbf{Arg} as follows:

$$\mathbf{o}_i = \mathbf{w} \cdot \mathbf{Arg}[i : i + h - 1] \quad (11)$$

where $i=1\dots s-h+1$. A bias term $b \in \mathbb{R}$ and an activation function f are added to each \mathbf{o}_i to compute the feature map c_i for this filter:

$$c_i = f(\mathbf{o}_i + b) \quad (12)$$

Then we use *max pooling* operation to get the representation of the argument:

$$\mathbf{R}_{\text{arg}} = \max\{c_i\} \quad (13)$$

C. Sentence Interaction Layer

Unlike sentence modeling to get semantic representation for each sentence, the sentence interaction aims to learn the relationship representation between two arguments rather than one single argument. We choose four ways to make arguments concatenate with each other, resulting in an interrelated representation of the two arguments. These methods based on the attention and the self-attention mechanism as follows.

- Attention: Perform attention operations on two argument vectors respectively and then concatenate them together
- Con-self-attention: Concatenate two argument vectors first and then perform self-attention operation on the concatenated vector
- Self-attention-con: Perform self-attention operation on two argument vectors respectively and then concatenate them together
- Attention-mlp: Perform attention interact operations on two sentence vectors respectively and

then feed their concatenation into Multi-Layer Perceptron (MLP)

Through the above four interactions, the separate \mathbf{R}_{Arg1} , \mathbf{R}_{Arg2} become joint pair representation \mathbf{R}_{pair} which contains the overall information of the two arguments.

D. Classification Layer

Finally we feed the result of joint representation of the arguments \mathbf{R}_{pair} into a full-connected softmax layer to predict the implicit discourse sense.

IV. EXPERIMENT

A. Dataset

We perform our experiments on the CDTB corpus². To make comparison with previous work, we use the data provided by the CoNLL-2016, which is adapted from the CDTB corpus and has been a benchmark corpus for study. Following previous work, we use the accuracy to evaluate the performance of models. Table II shows the distributions of Chinese and English corpus in CoNLL-2016.

Table II

THE DISTRIBUTIONS OF DISCOURSE RELATIONSHIP TYPES IN CONLL-2016 ENGLISH AND CHINESE CORPUS.

	English		Chinese	
	amount	percent(%)	amount	percent(%)
Explicit	18,459	45.5	2,398	21.75
Implicit	16,053	39.5	7,238	65.66
EntRel	5,210	12.8	1,219	11.06
AltLex	624	1.5	223	2.02
NoRel	254	0.6	0	0
Total	40,600	100	11,023	100

Clearly, there is more implicit data in Chinese than in English. We follow the previous work in [20] and combine the non-Explicit (i.e., Implicit, EntRel and AltLex) dataset as implicit samples, which makes the implicit discourse relationship recognition more challenging. The Chinese discourse relationship is divided into 9 categories. Table III lists the sense distribution breakdown of Chinese non-Explicit discourse.

Table III

CONLL-2016 CHINESE NON-EXPLICIT SENSE DISTRIBUTION.

Sense Label	Training	Development	Test
Conjunction	5,196	189	228
Expansion	1,228	49	40
EntRel	1,098	50	71
Causation	260	12	11
Purpose	79	2	6
Contrast	72	3	1
Temporal	36	0	1
Conditional	32	1	1
Progression	14	0	0

²<https://catalog.ldc.upenn.edu/LDC2013T21>

Table V
COMPARISONS OF ACCURACY(%) FOR SENTENCE INTERACTIONS.

	word2vec +BiLSTM	ELMo ₂ + BiLSTM	BERT _{single} + CNN
without Attention	70.19	72.70	74.09
Attention	68.80	68.52	70.75
Con-self-attention	70.75	65.74	71.30
Self-attention-con	72.98	70.20	70.75
Attention-mlp	70.47	64.90	69.63

B. Experiment Setup

We employ Adam optimization [34] using the cross-entropy loss function. In CNN model, [20] choose filter window size (1, 3, 5) to represent the *unigram*, *trigram* and *5-gram* features in sentence. We follow their choice because we test all the sub-set of (1, 3, 5, 7) and found (1, 3, 5) achieves the optimal performance. Following [35], we set hidden size as 50 in LSTM and BiLSTM. We set epochs as 50, batch size as 64, learning rate as 0.001 and dropout as 0.5.

In the word embedding layer, learning from the positive correlation between the vector dimension and expression ability in English, we train the 300 dimensions of word2vec and GloVe vector on Tagged Chinese Gigaword³. In the contextual embedding model, we get the three-layers ELMo representations by the tools provided by allenlp⁴ to train the contextual representation of words. As for BERT, we use the pre-trained Chinese model offered by Google, which is 12-layer, 768-hidden, 12-heads, 110M parameters. Both single sentence model and sentence pairs model are used in our experiment for BERT.

C. Results and Discussion

We evaluate our component-based model from different aspects, i.e., the contextual and context-free word embeddings, three sentence modeling methods and two relationship interaction strategies. Table IV and Table V reported the experimental results on test dataset. Next we analyze the performance of different components.

Table IV
COMPARISONS OF ACCURACY(%) FOR DIFFERENT WORD EMBEDDINGS AND SENTENCE MODELING METHODS.

	word2vecGloVe	ELMo			BERT	
		1	2	3	single	pairs
LSTM	67.68	70.47	70.31	71.59	67.41	67.69
BiLSTM	70.19	71.30	71.03	72.70	70.47	66.30
CNN	70.75	70.20	69.92	71.59	72.42	74.09

1) **word embeddings:** To examine the impact of different word embeddings and sentence modeling, we did not involve the operation of sentence interaction. From Table IV, we find BERT_{single} with CNN achieves the best performance (74.09% in accuracy). However,

³<https://catalog.ldc.upenn.edu/LDC2007T03>

⁴<https://github.com/allenai/allennlp>

comparing other embeddings, BERT performs not stable enough considering the worst is BERT_{single} with BiLSTM (66.30% in accuracy). The possible reason may be that BERT generates the single Chinese character embeddings rather than the word embeddings and thus the simple addition of two characters cannot be equal to the real word embedding. Furthermore, the ELMo_{second} performs well and stable comparing other embeddings.

2) sentence modeling: The performance of different sentence modeling methods cannot be summarized in one sentence. When in combination with different word embeddings, the performance of sentence modeling methods changes a lot. For example, comparing CNN with BiLSTM and LSTM, we find the bidirectional information is helpful in discourse relationship recognition when in combination with most word embeddings. However, CNN with BERT_{single} achieves the best performance among all setting. This indicates that not only word embedding but with their combination makes contribution to performance.

3) sentence interactions: Furthermore, we examine the performance of different sentence interactions along with the selected combination of word embedding and sentence modeling as shown in Table IV. From Table V we see that the sentence interactions did not perform well as we expected. Given word embedding and sentence modeling, the sentence interaction without attention outperforms other attention strategies. This is surprising as we think the diverse discourse relations is complex and supposed to be represented by complex operations rather than simple concatenation. To give a deep analysis of this phenomenon, we dive in to the Chinese implicit discourse corpus. We find that among all senses, *Conjunction* is a very common category (63.5%) as it is the default category when the relationship is hard to judge[36]. Since the two arguments in *Conjunction* are often flexible and varied in structure and content, the interaction relationship between the two arguments may not be effectively captured by the attention mechanisms proposed in this work. This opens a future study for interaction relationship representation.

Finally, Table VI shows the comparison of our best model with the recent state-of-the-art systems on CoNLL-2016 for multi-class classification. All these systems use gold standard argument pairs. Again, our model BERT_{single} with CNN achieves the best performance (74.09% in accuracy) and outperforms the state-of-the-art performance. This indicates the effectiveness of our proposed model.

V. CONCLUSION

In this paper, we present a component-based neural framework to investigate the neural components for Chinese implicit discourse relationship recognition. Different word embeddings, sentence modeling methods and relationship interaction strategies are

Table VI
COMPARISONS OF OUR BEST MODEL WITH RECENT SYSTEMS ON CoNLL-2016 CHINESE NON-EXPLICIT DATASET, ACCURACY(%).

	Development Set	Test Set
Wang and Lan (2016) [20]	73.53	72.42
Rutherford and Xue (2016)[37]	71.57	67.41
Schenk et al. (2016)[7]	70.59	71.87
Rönnqvist et al. (2017)[9]	-	73.01
Ours(BERT _{single} +CNN)	72.54	74.09

extensively explored. The experimental results showed that it is not one component does matter but their combination makes contribution to performance improvement. Besides, our proposed model achieves the SOTA performance in CoNLL-2016 Chinese corpus.

REFERENCES

- [1] B. Yang and C. Cardie, “Context-aware learning for sentence-level sentiment analysis with posterior regularization,” in *Proceedings of the 52nd ACL*, 2014, pp. 325–335.
- [2] J. J. Li, M. Carpuat, and A. Nenkova, “Assessing the discourse factors that influence the quality of machine translation,” in *Proceedings of the 52nd ACL*, 2014, pp. 283–288.
- [3] X. Wang, Y. Yoshida, T. Hirao, K. Sudoh, and M. Nagata, “Summarization based on task-oriented discourse parsing,” *IEEE Transactions on Audio, Speech, and Language Processing*, pp. 1358–1367, 2015.
- [4] R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. K. Joshi, and B. L. Webber, “The penn discourse treebank 2.0.” in *LREC*, 2008.
- [5] Y. Zhou and N. Xue, “Pdtb-style discourse annotation of chinese text,” in *Proceedings of the 50th ACL: Long Papers-Volume 1*, 2012, pp. 69–77.
- [6] C. Braud and P. Denis, “Comparing word representations for implicit discourse relation classification,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 2201–2211.
- [7] N. Schenk, C. Chiarcos, K. Donandt, S. Rönnqvist, E. Stepanov, and G. Riccardi, “Do we really need all those rich linguistic features? a neural network-based approach to implicit sense labeling,” in *Proceedings of the CoNLL-16 shared task*, 2016, pp. 41–49.
- [8] A. Rutherford, V. Demberg, and N. Xue, “A systematic study of neural discourse models for implicit discourse relation,” in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 2017, pp. 281–291.
- [9] S. Rönnqvist, N. Schenk, and C. Chiarcos, “A recurrent neural model with attention for the recognition of Chinese implicit discourse relations,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017, pp. 256–262.
- [10] E. Pitler and A. Nenkova, “Using syntax to disambiguate explicit discourse connectives in text,” in *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, 2009, pp. 13–16.

- [11] O. Biran and K. McKeown, “Aggregated word pair features for implicit discourse relation disambiguation,” 2013.
- [12] A. Rutherford and N. Xue, “Discovering implicit discourse relations through brown cluster pair representation and coreference patterns,” in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, 2014, pp. 645–654.
- [13] C. Braud and P. Denis, “Combining natural and artificial examples to improve implicit discourse relation identification,” in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014, pp. 1694–1705.
- [14] A. Rutherford and N. Xue, “Improving the inference of implicit discourse relations via classifying explicit discourse connectives,” in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015, pp. 799–808.
- [15] Y. Ji, G. Zhang, and J. Eisenstein, “Closing the gap: Domain adaptation from explicit to implicit discourse relations,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 2219–2224.
- [16] C. Braud and P. Denis, “Learning connective-based word representations for implicit discourse relation identification,” 2016.
- [17] C. Wu, X. Shi, Y. Chen, J. Su, and B. Wang, “Improving implicit discourse relation recognition with discourse-specific word embeddings,” in *Proceedings of the 55th ACL, year=2017*.
- [18] L. Qin, Z. Zhang, H. Zhao, Z. Hu, and E. Xing, “Adversarial connective-exploiting networks for implicit discourse relation classification,” in *Proceedings of the 55th Annual Meeting of ACL*, 2017.
- [19] C. Wu, Y. Chen, Y. Huang *et al.*, “Bilingually-constrained synthetic data for implicit discourse relation recognition,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 2306–2312.
- [20] J. Wang and M. Lan, “Two end-to-end shallow discourse parsers for english and chinese in conll-2016 shared task,” in *Proceedings of the CoNLL-16 shared task*, 2016.
- [21] L. Qin, Z. Zhang, and H. Zhao, “Shallow discourse parsing using convolutional neural network,” *Proceedings of the CoNLL-16 shared task*, pp. 70–77, 2016.
- [22] Y. Liu, S. Li, X. Zhang, and Z. Sui, “Implicit discourse relation classification via multi-task neural networks,” in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [23] M. L. Pacheco, I.-T. Lee, X. Zhang, A. K. Zehady, P. Daga, D. Jin, A. Parolia, and D. Goldwasser, “Adapting event embedding for implicit discourse relation recognition,” *Proceedings of the CoNLL-16 shared task*, pp. 136–142, 2016.
- [24] L. Qin, Z. Zhang, and H. Zhao, “Implicit discourse relation recognition with context-aware character-enhanced embeddings,” in *Proceedings of COLING 2016: Technical Papers*, pp. 1914–1924.
- [25] ———, “A stacking gated neural architecture for implicit discourse relation classification,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 2263–2270.
- [26] J. Chen, Q. Zhang, P. Liu, X. Qiu, and X. Huang, “Implicit discourse relation detection via a deep architecture with gated relevance network,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016, pp. 1726–1735.
- [27] Y. Wang, S. Li, J. Yang, X. Sun, and H. Wang, “Tag-enhanced tree-structured neural networks for implicit discourse relation classification,” in *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, 2017, pp. 496–505.
- [28] T. Mikolov, I. Sutskever, K. Chen, Corrado, and G. S., “Distributed representations of words and phrases and their compositionality,” in *Advances in neural information processing systems*, 2013.
- [29] Pennington, Jeffrey, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on EMNLP*, 2014.
- [30] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” in *Proc. of NAACL*, 2018.
- [31] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 4171–4186.
- [32] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, pp. 1735–1780.
- [33] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional lstm and other neural network architectures,” *Neural networks*, pp. 602–610.
- [34] K. D. P and B. J. Adam, “Adam: A method for stochastic optimization,” 2014.
- [35] M. Lan, J. Wang, Y. Wu, Z.-Y. Niu, and H. Wang, “Multi-task attention-based neural networks for implicit discourse relationship representation and identification,” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 1299–1308.
- [36] e. a. Xue, Nianwen, “Conll 2016 shared task on multilingual shallow discourse parsing.” in *Proceedings of the CoNLL-16 shared task (2016)*, 2016, pp. 1–19.
- [37] A. Rutherford and N. Xue, “Robust non-explicit neural discourse parser in english and chinese,” *Proceedings of the CoNLL-16 shared task*, pp. 55–59, 2016.

Syntax-aware Transformer Encoder for Neural Machine Translation

Sufeng Duan¹, Hai Zhao^{1,*}, Junru Zhou¹ and Rui Wang²

¹ Department of Computer Science and Engineering,

Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering,

MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University,
Shanghai, China

² National Institute of Information and Communications Technology (NICT),
Kyoto, Japan

1140339019dsf@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn, zhoujunru@sjtu.edu.cn, wangrui@nict.go.jp

Abstract—Syntax has been shown a helpful clue in various natural language processing tasks including previous statistical machine translation and recurrent neural network based machine translation. However, since the state-of-the-art neural machine translation (NMT) has to be built on the Transformer based encoder, few attempts are found on such a syntax enhancement. Thus in this paper, we explore effective ways to introduce syntax into Transformer for better machine translation. We empirically compare two ways, positional encoding and input embedding, to exploit syntactic clues from dependency tree over source sentence. Our proposed methods have a merit keeping the architecture of Transformer unchanged, thus the efficiency of Transformer can be kept. The experimental results on IWSLT’14 German-to-English and WMT14 English-to-German show that our method can yield advanced results over strong Transformer baselines.

Keywords-Neural Machine Translation, dependency parsing, POS Tagging

1

I. INTRODUCTION

Neural machine translation (NMT) models are based on a sequence-to-sequence (seq2seq) architecture, which uses an encoder to create a vector of source sequence and a decoder to predict target sequence, usually with an attention mechanism [1, 2, 3]. Recently, a series of seq2seq NMT models, such as recurrent neural network (RNN) model [2, 3, 4, 5], convolutional neural network [6] model and the Transformer which is empowered by self-attention mechanism [7], have been proposed and get good performance.

Seq2seq models have to take source input as sequence which makes the tree-based syntactic information difficult to add to the model directly. Inspired by incorporating syntactic information in statistical machine translation (SMT) [8, 9], syntactic information has been incorporated into NMT model for better performance [10, 11, 12, 13] in the similar ways of SMT, in which two types of approaches have been proposed, one is to still introduce tree structure directly [10], like tree long-short term

^{1*}* Corresponding author. This paper was partially supported by National Key Research and Development Program of China (No. 2017YFB0304100) and Key Projects of National Natural Science Foundation of China (No. U1836222 and No. 61733011).

memory (Tree-LSTM) and tree-attention, and the other turns to linearization of syntactic tree [11, 12, 13].

However, existing works using syntactic information are almost all for RNN-based NMT. For Transformer, its self-attention mechanism has been capable of capturing general relationship among words which is supposed to be delivered by syntax from a view of linguistics. Thus using a syntax integration method like [12], it is quite possible that no significant performance improvement from the syntactic clues will be observed. For those models introducing syntax in terms of tree structure, model architecture has to be modified and more parameters will be introduced, which brings the drawback of slowing down computational efficiency. At last, the Transformer encoder adopts scaled dot-product attention to finalize representation, which is quite different from what RNN-based encoder does, thus it is not so convenient to modify the Transformer architecture to accommodate syntactic information as the same integration way in RNN-based NMT.

In this work, we propose using syntax to enhance the Transformer based NMT in terms of input embedding and positional encoding. The first method which is inspired by the work [12] using independent syntactic embedding to enhance RNN-based NMT views the tree structure as one label and encodes it by embedding. The second method transfers tree features into positional encoding, and may conveniently keep the model architecture unchanged. Our experiments on German-to-English (De-En) and English-to-German (En-De) translation show that our proposed syntax-enhanced Transformer indeed obtains better translation performance.

II. BACKGROUND

A. Syntactic Information for NMT

There are two ways to incorporate syntactic information into NMT model. The first is to use tree structure such as Tree-LSTM to directly encode syntactic information. Eriguchi et al. [10] propose a tree-to-sequence model with a tree-based encoder, which encodes the phrase structure of a sentence as vectors.

The second is to represent syntax through linearization. Linearized syntactic tree is advantageous because it is

easily added into models with minor modifications of the model architecture. Aharoni and Goldberg [11] design a sequence-to-tree model which translates source sentence to a linearized constituency tree. Sennrich and Haddow [14] propose a feature method which will be followed by this work,

$$\vec{h}_j = \tanh(\vec{W}(\parallel_{k=1}^{|F|} E_k x_{jk}) + \vec{U} \vec{h}_{j-1}), \quad (1)$$

where m is the word embedding size, $\vec{U} \in \mathbb{R}^{n \times n}$ are weight matrices, n is the number of hidden units, $|F|$ is the number of additional features, \parallel is the operator of concatenation, $E_k \in \mathbb{R}^{m_k \times K_k}$ are the feature embedding matrices, with $\sum_{k=1}^{|F|} m_k = m$, \vec{h}_{j-1} is the hidden states of the $(j-1)$ -th words and K_k is the vocabulary size of the k -th feature.

B. Transformer

Transformer is introduced by [7] which is based on self-attention network and good at getting the relationship between words in sentence. The encoder of Transformer is composed of a stack of N layers, which has two sub-layers in one layer. The first sub-layer is a multi-head self-attention structure to generate the attention of input. Transformer computes the matrix of outputs as

$$\text{Attention}(H) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2)$$

where Q is the matrix of queries, K is the matrix of keys and V is the matrix of values. Queries, keys and values are transferred from the input H . Q , K and V are all generated from input representations. The second sub-layer is a position-wise fully connected feed-forward network. The Transformer employs a residual connection [15] which is followed by layer normalization [16]. The output of sub-layers in encoder is

$$\begin{aligned} \mathbf{C}^k &= \text{LAYERNORM}(\text{ATTENTION}(\mathbf{H}^{k-1}) + \mathbf{H}^{k-1}) \\ \mathbf{H}^k &= \text{LAYERNORM}(\text{FFN}(\mathbf{C}^k) + \mathbf{C}^k) \end{aligned} \quad (3)$$

where C^k is the output of first sub-layer in the k -th layer, H^k is the output of second sub-layer in the k -th layer. The decoder is composed of a stack of N layers. Different from encoder, decoder layer adds another sub-layer to perform attention over the output of encoder.

Several variants of Transformer have been proposed. Shaw et al. [17] extend the self-attention mechanism to efficiently consider representations of the relative positions or distances between sequence elements. Xiao et al. [18] propose a lattice-based encoder to explore effective word or subword representation in an automatic way during training.

As few existing studies consider using syntax to enhance the state-of-the-art Transformer though syntax itself has been shown helpful for previous SMT and RNN-based NMT, we thus make an attempt to filling the gap by exploring a syntax-enhanced Transformer.

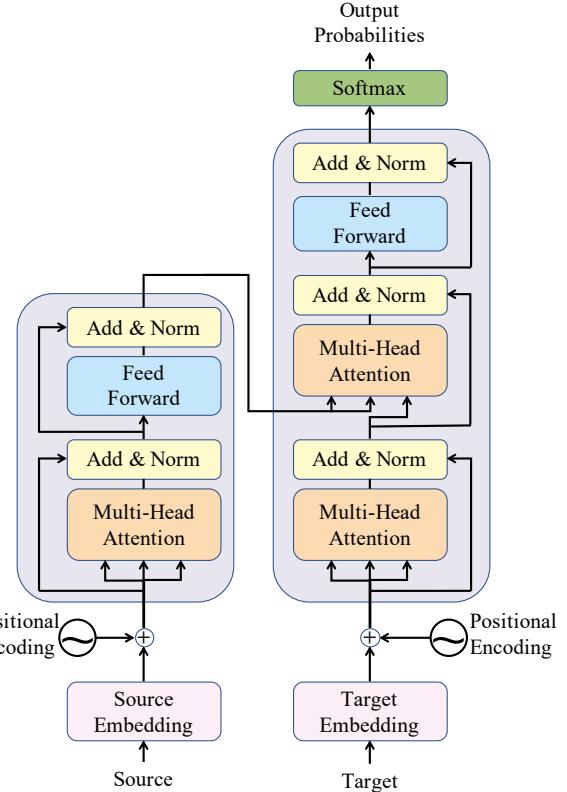


Figure 1. Architecture of Transformer

C. Dependency Parsing

Dependency parsing aims to predict the existence and type of linguistic dependency relations between words. It is a fundamental task in natural language processing (NLP) [19, 20, 21, 22, 23, 24, 25]. Dependency parsing is often used for other NLP tasks such as semantic role labeling [23, 24, 26, 27, 28]. For most east Asian languages such as Chinese, dependency parsing relies on word segmentation [29, 30, 31, 32]. It can be roughly put into two categories in terms of searching strategies over parsing trees, graph-based and transition-based [33]. With the development of neural network applied to dependency parsing, there comes continuous progress for better parsing performance [22, 34]. Zhang et al. [35] propose a neural probabilistic parsing model which explores up to third-order graph-based parsing with maximum likelihood training criteria. Li et al. [36] propose a full character-level neural dependency parser together with a released character-level dependency treebank for Chinese. The dependency parsing is shown to be more effective than non-neural parser. Wu et al. [37] propose a system for multilingual universal dependency parsing from raw text. Li et al. [38] propose a tree encoder and integrate pre-trained language model features for a better representation of partially built dependency subtrees and thus enhances the model.

III. MODELS

Fig. 2 shows the architecture of our model. Different from existing methods encoding syntax into the source

Table I
FEATURE EXAMPLES FOR A SENTENCE. FOR ROOT NODE, ITS DEPENDENCY LABEL IS *root*.

words	It	is	a	good	thing	for	people	.
order	1	2	3	4	5	6	7	8
POS tags	PRP	VBD	DT	JJ	NN	IN	NNS	.
Dependency labels	sbj	root	det	amod	obj	case	nmod	punct
parent position	2	0	5	5	2	7	5	2
tree depth	2	1	3	3	2	4	3	2

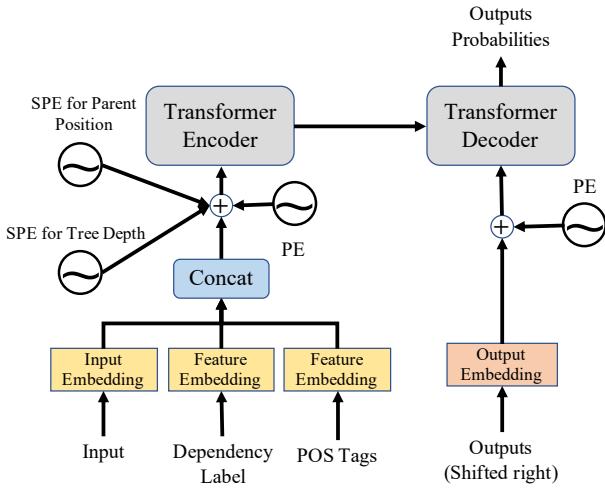


Figure 2. Architecture of our model. Our syntactic features are encoded into the input embedding or position encodings.

[10, 11], we consider two ways to encode syntactic information derived from a syntactic dependency parse tree. One is directly extracting syntactic clues as a part of input embedding, the other is regarding appropriate syntactic clues as a type of position information thus putting them into positional encodings of the Transformer. From syntactic source, we extract four types of syntactic features, including 1) POS tags, 2) dependency labels, 3) parent position and 4) tree depth of a node. The features 1-3 can be integrated into input embedding, and the features 3 and 4 can be encoded into positional encodings of the Transformer.

A. Syntactic Features

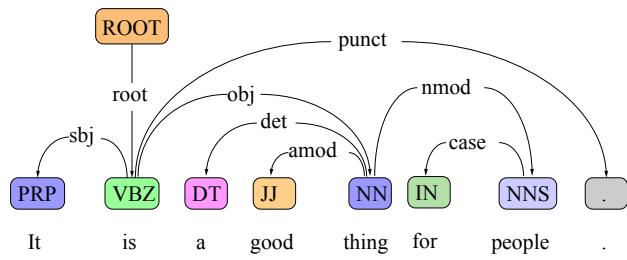


Figure 3. Dependency tree. One tree has only one root. One node except root in the dependency tree has only one parent and one dependency label.

Dependency parsing is to predict the existence and type of linguistic dependency relations between words in

one sentences [39], which results in a dependency tree structure for every word in the sentence as shown in Fig. 3. Each word except for the root in one sentence has one and only one parent. The dependency label between the word and its parent can be viewed as a feature of this word which indicates the relationship type between it and its parent. For the root, it has no parent and thus no defined dependency label, we simply set its dependency label as **root**. POS tags can be processed in the same way as dependency label because each word in sentence is associated with one POS tag.

As a sentence is written in a linear way which only conveniently represents the relationship of neighboring elements, a syntactic parsing tree may disclose nonlinear relationship between words for the same sentence. The relationship between the word and its parent is represented as an edge. In the dependency tree shown in Fig. 3, word **It** is syntactically closer to **is** than **a** because there is a directed edge between **It** and **is**. As dependency relation represented as edges connects any word and its parent, and one word except root in sentence has one and only one parent in the dependency parse tree, we adopt position of parent node in the parse tree for each word as a syntactic feature. Besides, the depth in the parse tree for a node indicates how far it is from the root, which is the most salient word in a sentence, thus the tree depth indicates how important a word is in terms of syntactic impact, we then adopt tree depth of the node as another syntactic feature.

Note the parent position and tree depth of nodes are non-zero integer. The range of the two syntactic feature values are also limited to the length of sentence.

B. Adding Syntactic Features into Input Embedding

Linearization method has been adopted for integrating syntax into RNN-based NMT [12, 14]. In this work, following [14], we take three syntactic features, dependency label, POS tag and parent position respectively as three types of embeddings to feed the model. All these extra syntactic feature embeddings are concatenated to the original input embedding as the model input as shown in Fig. 2.

Viewing the dependency label and POS tag as two features, the model needs to individually put the features into a range of dimensions. To avoid the hidden vector divided by multi-head attention [7], the dimension of one head should be larger than the dimension of one feature.

Note that syntactic features introduced in this input embedding way can be updated during training, which is

essentially important for effectively using parent position feature.

C. Syntactic Positional Encodings

The Transformer contains no recurrence and no convolution [7], in order to make use of word order information in sentence, it uses positional encoding (PE) to encode the position of words. The PE has the same dimension as the input embedding so that they can be summed. PE used in the Transformer is computed by

$$\begin{aligned} PE_{(pos,2i)} &= \sin(pos/10000^{2i/d_{\text{model}}}), \\ PE_{(pos,2i+1)} &= \cos(pos/10000^{2i/d_{\text{model}}}), \end{aligned} \quad (4)$$

where pos is word position in sentence, i is the dimension.

For each word, its parent position and tree depth in dependency parse tree are similar to the word position. Thus we design syntactic positional encoding (SPE) to accommodate two types of syntactic clues, the parent position and tree depth of the node. The functions to compute different dimensions of positional encoding are similar to the original Transformer

$$\begin{aligned} SPE_{(pos,2ni+b,c_f)} &= \sin(pos/c_f^{2ni/d_{\text{model}}}), \\ SPE_{(pos,2ni+b+1,c_f)} &= \cos(pos/c_f^{2ni/d_{\text{model}}}), \end{aligned} \quad (5)$$

where pos is the value of parent position or tree depth, n is the number of features, i is the dimension, b is the offset of dimensions and c_f is a constant for feature f . Equation (5) sets different features the same dimensions. Dimensions of different features are staggered by b . For different features, c_f will be set to different value. In detail, we add two SPE features, parent position and tree depth of node. For these two features, we set c_f to 2000 for the parent position and 400 for the tree depth. As the original PE is independent from SPE, it will be retained in our model.

D. Dependency Tree on Subwords

As we introduce a predicted syntactic parse tree which is built over a sequence of words while the current open-vocabulary NMT relies on a subword segmentation over source or target words which usually adopts BPE (byte pair encoding) algorithm [40], there is a word/subword mismatch when we try to integrate syntax into the source side of NMT models. For words segmented into multiple subwords, we set the first subword part of a word to inherit its dependency label, parent position and POS tag. For the rest subwords of the word, we build new edges between them and the first one, respectively. Note that our processing here is different from [14] that uses a feature to mark the boundary of subwords.

IV. EXPERIMENTS

A. Datasets

We evaluate our model on two translation tasks, IWSLT14 German-English (De-En) and WMT14 English-German (En-De).

IWSLT14 German-English IWSLT14 De-En dataset contains 153K training sentence pairs. We use 7K data

from the training set as validation set and use the combination of dev2010, dev2012, tst2010, tst2011 and tst2012 as test set with 7K sentences which are preprocessed by script². BPE algorithm is used to process words into subwords, and number of subword tokens in the shared vocabulary is 31K.

WMT14 English-German We use the WMT14 En-De dataset from Stanford³ with 4.5M sentence pairs for training. We use the combination of newstest2012 and newstest2013 as validation set and newstest2014 as test set. The sentences longer than 80 are removed from the training dataset. Dataset is segmented by BPE so that number of subwords in the shared vocabulary is 32K.

Generally, we use Stanford Parser⁴ to process German corpus to get dependency label and POS tags. For English corpus, we use Stanford Dependency Parser⁵ to get dependency label and Stanford POS Tagger⁶ to get predicted POS tags. The parent position of one word will be set to 0 if the parent is root.

B. Hyperparameters

The hyperparameters for our experiments are shown in Table IV. For De-En, we follow the setting of Transformer-small. For En-De, we follow the setting of Transformer-base. For both tasks, the dimension for one feature is 32. The input embedding size of our model is from summing up the dimensions of word embeddings and syntactic features.

C. Training

All our models are trained on one CPU (Intel i7-5960X) and one nVidia 1080Ti GPU. The implementation of model is based on OpenNMT-py⁷. We choose Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-9}$ and the learning rate setting strategy, which are all the same as [7]. We use beam search decoder for De-En task with beam width 6. For En-De, following [7], the width for beam search is 5 and the length penalty α is 0.6. The batch size is 1024 for De-En and 2048 for En-De. We evaluate the translation results by using case-insensitive BLEU⁸.

V. RESULTS

Our baselines for De-En and En-De tasks are the Transformer-small and the Transformer-base. Table II is the results of our main experiments. For De-En and En-De tasks, our model which incorporates POS tags, parent position and dependency labels through embeddings respectively outperforms the baselines by 0.4 BLEU and 0.50 BLEU. Our model which incorporates POS tags and dependency labels through embeddings and

²<https://github.com/eske/seq2seq/blob/master/config/IWSLT14/prepare-mixer.sh>

³<https://nlp.stanford.edu/projects/nmt/>

⁴<https://nlp.stanford.edu/software/lex-parser.html>

⁵<https://nlp.stanford.edu/software/nndep.html>

⁶<https://nlp.stanford.edu/software/tagger.html>

⁷<https://github.com/OpenNMT/OpenNMT-py/archive/0.7.0.zip>

⁸<https://github.com/OpenNMT/OpenNMT-py/blob/master/tools/multi-bleu.perl>

Table II
BLEU SCORES ON IWSLT 2014 AND WMT14. THE BASELINES FOR DE-EN TASK AND EN-DE TASK ARE THE TRANSFORMER-SMALL AND THE TRANSFORMER-BASE, RESPECTIVELY.

Model	BLEU	
	DE-EN	EN-DE
Transformer (small)	31.80	-
Transformer (base)	-	26.80
Our method (POS Tag & Parent & Dependency label)	32.20	27.30
Our method (POS Tag & Dependency label & SPE)	31.90	27.15

Table III
BLEU SCORES ON WMT14 EN-DE EXPERIMENT WITH EMBEDDING TO ADD FEATURES. WE REMOVE DIFFERENT FEATURE, SET THE DIMENSIONS OF INPUT EMBEDDINGS TO 576 AND THE HEADS TO 9. IN THESE EXPERIMENTS, WE USE ONLY EMBEDDINGS TO INCORPORATE FEATURES.

Model	EN-DE
Transformer (base)	26.80
Our Model	27.30
- Parent Position	27.00
- POS Tags	27.01
- Dependency Label	26.70

Table IV
HYPERPARAMETERS FOR OUR EXPERIMENTS. FF IS SHORT FOR FEED-FORWARD LAYER. THE NUMBER OF HEADS IS BASED ON THE DIMENSION FOR WORD AND FEATURE.

Parameter	DE-EN		EN-DE	
	Without SPE	With SPE	Without SPE	With SPE
Layers	6	6	6	6
Dimension for Word	256	256	512	512
Dimension for Feature	32	32	32	32
Head	11	5	19	9
FF	1408	1280	2432	2304
Dropout	0.3	0.3	0.1	0.1

incorporates parent position and tree depth of nodes through SPE respectively outperforms the baselines by 0.10 BLEU and 0.35 BLEU. The results show that both of our model settings with syntax-informed features can enhance the respective NMT. The model using only embeddings to introduce syntax performs better than the model with SPE. We attribute this difference to the trainable ways between embedding and SPE. The input embedding can be updated during training but SPE cannot. Such representation updating difference causes the observable performance change of the models.

Table V
BLEU SCORES ON WMT14 EN-DE. WE RESPECTIVELY REMOVE THE FEATURE OF PARENT POSITION AND THE FEATURE OF TREE DEPTH FROM OUR MODEL WITH FULL SPE FEATURES.

Model	EN-DE
Transformer (base)	26.80
Our Model with SPE	27.15
- Parent Position	27.00
- Tree Depth	26.68

To compare importance of different features, we test models by removing one feature one by one from the three syntactic features. Table III shows that the model without the dependency labels obtains 26.70 BLEU score, while the BLEU scores without POS tags and parent position are respectively 27.01 and 27.00, which indicates that dependency label feature contributes most to the performance. Overall, the model with full three syntactic features outperforms the baseline model, which verifies the effectiveness of syntactic clues.

To evaluate the effect of SPE features, we perform experiments on WMT En-De 14 task by removing either of SPE features one by one. Table V shows the results. Comparing these results shows that the tree depth is more informative than the parent position for performance improvement. In the meantime, either of the two SPE features cannot individually outperform full SPE features which indicates either SPE feature is essential to the performance contribution.

VI. CONCLUSION

In this paper we propose to exploit syntactic information to enhance Transformer through input word embeddings and PE. The experimental results verify that the proposed methods using syntax can improve the performance of Transformer without changing the architecture of model. We also compare the proposed two ways of introducing syntax into the Transformer, which shows that embedding updating is a factor making the translation performance difference and thus the direct input embedding integration gives better translation. In addition, we empirically study the contribution of detailed syntactic features including POS tags, dependency labels, parent position and tree depth. We conclude that all the proposed syntactic features are indeed helpful for enhancing the Transformer in NMT.

REFERENCES

- [1] N. Kalchbrenner and P. Blunsom, “Recurrent continuous translation models,” in *EMNLP*, 2013, pp. 1700–1709.
- [2] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *NIPS*, 2014, pp. 3104–3112.
- [3] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.

- [4] Y. Wu and H. Zhao, “Finding better subword segmentation for neural machine translation,” in *CCL*, 2018, pp. 53–64.
- [5] H. Zhang and H. Zhao, “Minimum divergence vs. maximum margin: an empirical comparison on seq2seq models,” in *ICLR*, 2019.
- [6] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, “Convolutional sequence to sequence learning,” in *ICML*, 2017, pp. 1243–1252.
- [7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *NIPS*, 2017, pp. 6000–6010.
- [8] M. Galley, M. Hopkins, K. Knight, and D. Marcu, “What’s in a translation rule?” in *HLT-NAACL*, 2004, pp. 273–280.
- [9] M. Galley, J. Graehl, K. Knight, D. Marcu, S. DeNeeve, W. Wang, and I. Thayer, “Scalable inference and training of context-rich syntactic translation models,” in *ACL*, 2006, pp. 961–968.
- [10] A. Eriguchi, K. Hashimoto, and Y. Tsuruoka, “Tree-to-sequence attentional neural machine translation,” in *ACL*, 2016, pp. 823–833.
- [11] R. Aharoni and Y. Goldberg, “Towards string-to-tree neural machine translation,” in *ACL*, 2017, pp. 132–140.
- [12] J. Li, D. Xiong, Z. Tu, M. Zhu, M. Zhang, and G. Zhou, “Modeling source syntax for neural machine translation,” in *ACL*, 2017, pp. 688–697.
- [13] X. Wang, H. Pham, P. Yin, and G. Neubig, “A tree-based decoder for neural machine translation,” in *EMNLP*, 2018, pp. 4772–4777.
- [14] R. Sennrich and B. Haddow, “Linguistic input features improve neural machine translation,” in *ACL*, 2016, pp. 83–91.
- [15] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016, pp. 770–778.
- [16] J. Lei Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016.
- [17] P. Shaw, J. Uszkoreit, and A. Vaswani, “Self-attention with relative position representations,” in *NAACL*, 2018, pp. 464–468.
- [18] F. Xiao, J. Li, H. Zhao, R. Wang, and K. Chen, “Lattice-based transformer encoder for neural machine translation,” in *ACL*, 2019, pp. 3090–3097.
- [19] X. Ma, X. Zhang, H. Zhao, and B. Lu, “Dependency parser for Chinese constituent parsing,” in *CIPS-SIGHAN*, 2010.
- [20] X. Ma and H. Zhao, “Fourth-order dependency parsing,” in *COLING*, 2012, pp. 785–796.
- [21] H. Zhao, X. Zhang, and C. Kit, “Integrative semantic dependency parsing via efficient large-scale feature selection,” *J. Artif. Intell. Res.*, vol. 46, pp. 203–233, 2013.
- [22] Z. Li, S. He, Z. Zhang, and H. Zhao, “Joint learning of POS and dependencies for multilingual universal dependency parsing,” in *CoNLL*, 2018, pp. 65–73.
- [23] Z. Li, S. He, J. Cai, Z. Zhang, H. Zhao, G. Liu, L. Li, and L. Si, “A unified syntax-aware framework for semantic role labeling,” in *EMNLP*, 2018, pp. 2401–2411.
- [24] S. He, Z. Li, H. Zhao, and H. Bai, “Syntax for semantic role labeling, to be, or not to be,” in *ACL*, 2018, pp. 2061–2071.
- [25] J. Zhou and H. Zhao, “Head-driven phrase structure grammar parsing on penn treebank,” in *ACL*, 2019, pp. 2396–2408.
- [26] J. Cai, S. He, Z. Li, and H. Zhao, “A full end-to-end semantic role labeler, syntactic-agnostic over syntactic-aware?” in *COLING*, 2018, pp. 2753–2765.
- [27] Z. Li, S. He, H. Zhao, Y. Zhang, Z. Zhang, X. Zhou, and X. Zhou, “Dependency or span, end-to-end uniform semantic role labeling,” in *AAAI*, 2019, pp. 6730–6737.
- [28] C. Guan, Y. Cheng, and H. Zhao, “Semantic role labeling with associated memory network,” in *NAACL-HLT*, 2019, pp. 3361–3371.
- [29] X. Wang, D. Cai, L. Li, G. Xu, H. Zhao, and L. Si, “Unsupervised learning helps supervised neural word segmentation,” in *AAAI*, 2019, pp. 7200–7207.
- [30] D. Cai, H. Zhao, Z. Zhang, Y. Xin, Y. Wu, and F. Huang, “Fast and accurate neural word segmentation for Chinese,” in *ACL*, 2017, pp. 608–615.
- [31] D. Cai and H. Zhao, “Neural word segmentation learning for Chinese,” in *ACL*, 2016, pp. 409–420.
- [32] H. Zhao and C. Kit, “Integrating unsupervised and supervised word segmentation: The role of goodness measures,” *Inf. Sci.*, vol. 181, no. 1, pp. 163–183, 2011.
- [33] Z. Li, J. Cai, S. He, and H. Zhao, “Seq2seq dependency parsing,” in *COLING*, 2018, pp. 3203–3214.
- [34] H. Wang, H. Zhao, and Z. Zhang, “A transition-based system for universal dependency parsing,” in *CoNLL*, 2017, pp. 191–197.
- [35] Z. Zhang, H. Zhao, and L. Qin, “Probabilistic graph-based dependency parsing with convolutional neural network,” in *ACL*, 2016, pp. 1382–1392.
- [36] H. Li, Z. Zhang, Y. Ju, and H. Zhao, “Neural character-level dependency parsing for Chinese,” in *AAAI*, 2018, pp. 5205–5212.
- [37] Y. Wu, H. Zhao, and J. Tong, “Multilingual universal dependency parsing from raw text with low-resource language enhancement,” in *CoNLL*, 2018, pp. 74–80.
- [38] Z. Li, J. Cai, and H. Zhao, “Effective representation for easy-first dependency parsing,” in *PRICAI*, 2019, pp. 351–363.
- [39] X. Ma, Z. Hu, J. Liu, N. Peng, G. Neubig, and E. Hovy, “Stack-pointer networks for dependency parsing,” in *ACL*, 2018, pp. 1403–1414.
- [40] R. Sennrich, B. Haddow, and A. Birch, “Neural machine translation of rare words with subword units,” in *ACL*, 2016, pp. 1715–1725.

An End-to-End Model Based on TDNN-BiGRU for Keyword Spotting

Shuzhou Chai, Zhenye Yang, Changsheng Lv, Wei-Qiang Zhang

Beijing National Research Center for Information Science and Technology

Department of Electronic Engineering, Tsinghua University

Beijing 100084, China

chaisz19@mails.tsinghua.edu.cn, yeezy990511@my.swjtu.edu.cn, lvchangsheng@bupt.edu.cn, wqzhang@tsinghua.edu.cn

Abstract—In this paper, we proposed a neural network architecture based on Time-Delay Neural Network (TDNN)-Bidirectional Gated Recurrent Unit (BiGRU) for small-footprint keyword spotting. Our model consists of three parts: TDNN, BiGRU and Attention Mechanism. TDNN models the time information and BiGRU extracts the hidden layer features of the audio. The attention mechanism generates a vector of fixed length with hidden layer features. The system generates the final score through vector linear transformation and softmax function. We explored the step size and unit size of TDNN and two attention mechanisms. Our model has achieved a true positive rate of 99.63% at a 5% false positive rate.

Keywords-TDNN; BiGRU; Keyword Spotting; Attention Mechanism;

I. INTRODUCTION

With the development of artificial intelligence, various intelligent terminal devices have emerged. Voice interaction has become an indispensable part of smart life. Users can use specific words to wake up smart devices, which is widely used today. For example, we can use “Hey Siri” to wake up Apple Devices and “Okay/Hey Google” on Google Home [1] for voice search. Keyword Spotting (KWS) aims to detect predefined and small-sized keywords in an audio stream.

Classical methods used in KWS include the Keyword/Filler Hidden Markov Model (HMM) [2,3,4] and large vocabulary continuous speech recognition systems (LVCSR) [5,6]. LVCSR requires at least tens of hours training corpus including annotated data and a reliable pronunciation dictionary. However, obtaining such data in practical applications requires a high price. In addition, LVCSR need to generate rich lattices, which require a large amount of computing resources. It is not suitable for mobile terminals with low power and limited performance. An HMM model is trained for each keyword while training one or more filler models HMM for the non-keyword speech segments. At the time of detection, keyword discrimination and localization are realized by

Viterbi decoding. According to the HMM topology, a large amount of operations and memory may be caused.

The end-to-end KWS systems have become popular in recent years. Deep KWS [7] introduced a multilayer perceptron as an alternative to the HMM-based approach. Sainath and Parada [8] build on this work and use Convolutional Neural Networks (CNN) to achieve better results. They mentioned that reduced the model footprints is the main motivation for turning to CNNs. Later, feed-forward DNN is replaced by more powerful networks such as recurrent neural networks (RNNs) [9] and residual networks (ResNet) [10]. ResNet does not have a good long-term dependence on voice audio. The RNN directly models on the input features without learning the local structure between successive time series and frequency steps. The Convolutional Recurrent Neural Networks (CRNN) [11] combines RNN and CNN with better performance than RNN or CNN.

GRU is a good variant of the LSTM network. It is simpler than the LSTM network. It is also a very popular network. There are three gate functions in LSTM: input gates, forgetting gates, and output gates to control input values, memory values, and output values. There are only two gates in the GRU model: the update gate and the reset gate. Bidirectional Gated Recurrent Unit (BiGRU) provides bi-directional time series features. Each layer of BiGRU includes a forward pass and a backward pass. The output at each moment is determined by the GRUs of the previous moment in opposite directions. The final output at this moment is generated by weighted the forward pass and backward pass.

In this paper, we propose a method combining TDNN and BiGRU to train parameters and improve the accuracy of the KWS system. We use a TDNN model with the ability to extract context information at the same time. It has multiple layers and each layer has sufficient internal connections to ensure that the network can learn complex nonlinear decision surfaces. Secondly, it can depict the relationship between the frames which do not require precise time alignment of the label. And we use a layer of BiGRU to obtain the sequence characteristics of the data. Besides, we introduce the attention mechanism model to weight the average of features and improve the accuracy of the KWS system. Finally through linear transformation and softmax function, we can output the decision score. Our model

The corresponding author is Wei-Qiang Zhang.

This work was supported by the National Natural Science Foundation of China under Grant No. U1836219 and the National Key R&D Program of China.

Zhenye Yang is with the Southwest Jiaotong University, Chengdu 611756, China, Changsheng Lv is with the Beijing University of Posts and Telecommunications, Beijing 100876, China

directly outputs the results of keyword detection without complex searches. In addition, we explored the impact of different networks and corresponding parameters, including average attention and soft attention, the step of TDNN and the unit of TDNN.

II. MODEL DESCRIPTION

A. End-to-end architecture

We propose to use a combination of TDNN and BiGRU models in small footprint keyword spotting. As depicted in Figure 1, we first extract the Fbank features $\mathbf{x} = (x_1, \dots, x_T)$ and preprocess the audio of different lengths

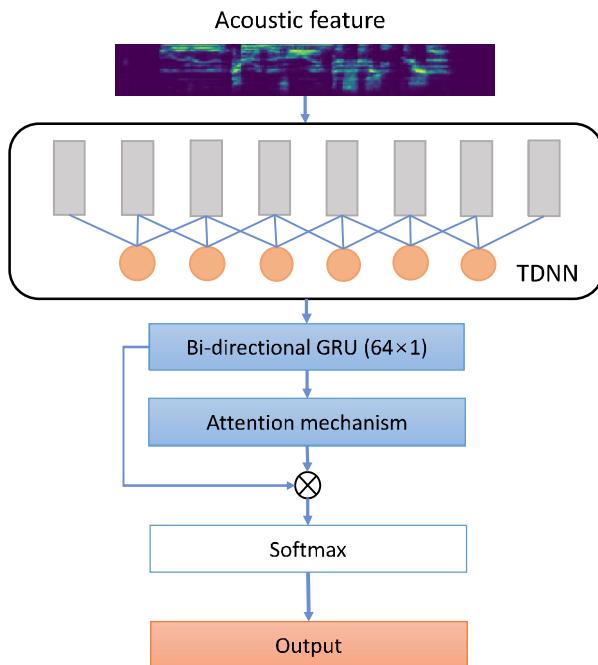


Figure 1. Architecture of our neural networks. TDNN extracts the time information. It generates a vector of fixed length as weighted average of the features of the BiGRU output and the normalized weights of attention mechanism. The Softmax layer outputs the final detection score.

for each audio. Then we use TDNN to get feature $\mathbf{h} = (h_1, \dots, h_T)$ that contain more time relationships:

$$\mathbf{h} = TDNN(\mathbf{x}) \quad (1)$$

A set of bidirectional GRUs are utilized to obtain long term dependencies in audios, which results in a feature of high level $\mathbf{g} = (g_1, \dots, g_t)$:

$$\mathbf{g} = BiGRU(\mathbf{h}) \quad (2)$$

The attention mechanism focuses the model at a position that should be noted, and learns the normalized weight α_t of each frame in the output fatures of the BiGRU:

$$\alpha_t = Attend(\mathbf{g}_t) \quad (3)$$

The output vector \mathbf{d} of the model is calculated by weighted average the output \mathbf{d} features of BiGRU:

$$\mathbf{d} = \sum_{t=1}^T \alpha_t \mathbf{g}_t \quad (4)$$

Finally the probability distribution of the result is generated by the linear transformation and the softmax function. We choose cross-entropy as the loss function and rectified linear units (ReLU) [12] as the activation function.

B. Feature Extraction

The Filter-bank (Fbank) feature is selected to extract in the feature extraction model. we generate acoustic features based on 40-dimensional log-filterbank energies computed every 10 ms over a window of 25 ms. Each additional frame of future context adds 10ms of latency to system because of the asymmetry in the input window. To provide the best trade-off between accuracy, latency, and computation [13], we use 5 future frames and 10 past frames. There is no stack of frames based on the GRU system.

C. TDNN Extraction Features

TDNN is a feedforward neural network architecture for modeling time series information. Unlike DNN, the input of the next hidden layer is obtained by splicing the output of the current time and the current time before and after the previous hidden layer. The speech mode[14] suitable for time step can learn a wider time relationship. In this paper, for a segment of speech with length N, each frame has 40-dimensional Fbank characteristics. The input of these time delay units which are 3 steps and 288 units are taken as baselines. These time delay units enter three frame windows spatially, and each unit receives input from the coefficients in three frame windows (through 3*40 weighted connections). Research shows that the 30ms window is sufficient to represent the keyword combination for detection[15].

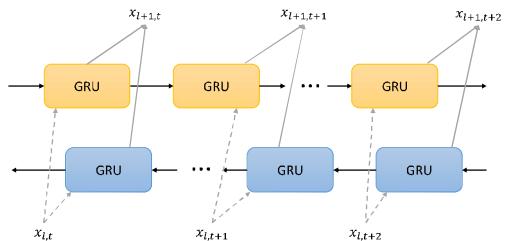


Figure 2. Structure of BiGRU. There are a backward pass a forward pass in every BiGRU layer. Parameter t represents time and l represents the 1 th layer.

D. BiGRU encoder

Long short-term memory (LSTM) is a network of RNNs for modeling long-range dependencies. It is designed to solve the problem of explosion and vanishing gradients. Bidirectional long short-term memory (BiLSTM) combines information of past, current and future time with great success in automatic speech recognition. GRU is a good variant of LSTM. Inspired by the success of BiLSTM, we choose BiGRU as the encoding layer in the model.

There are a forward pass and a backward pass in each BiGRU layer. At each moment, the input provides two

GRUs in opposite directions and the output is determined by the two one-way GRUs. The current hidden layer state of BiGRU is determined by the current input x_t , the output $\overrightarrow{h}_{(t-1)}$ of the forward hidden layer state at $(t-1)$ time, and the output $\overleftarrow{h}_{(t-1)}$ of the reverse hidden layer state:

$$\overrightarrow{h}_t = GRU(x_t, \overrightarrow{h}_{(t-1)}) \quad (5)$$

$$\overleftarrow{h}_t = GRU(x_t, \overleftarrow{h}_{(t-1)}) \quad (6)$$

$$h_t = w_t \overrightarrow{h}_t + v_t \overleftarrow{h}_t + b_t \quad (7)$$

The $GRU()$ function nonlinearly transforms the input audio feature vector and encodes the audio feature vector into the corresponding GRU hidden layer state. W_t represents the forward hidden layer state, \overrightarrow{h}_t corresponding to the BiGRU at time t . V_t represents the weight corresponding to the reverse hidden state \overleftarrow{h}_t , b_t represents the offset corresponding to the hidden layer state at time t .

E. Attention mechanism

Attention mechanism makes our model more focused on the part of the speech that contains the keyword and ignores the unrelated part, which is similar to human listening attention. Our attention mechanism provides a better accuracy for KWS. We try both average attention and soft attention. The soft attention is selected to automatically learn how to describe the speech content because of the higher efficiency[16].

Average attention: The attention score α_t is set as the average of T :

$$\alpha_t = \frac{1}{T} \quad (8)$$

Soft attention: Firstly, it learns a scalar score as

$$e_t = v^T \tanh(W h_t + b) \quad (9)$$

where, h_t is the hidden states. Then soft max is applied to compute the normalized weight as

$$\alpha_t = \frac{\exp(e_t)}{\sum_{j=1}^T \exp(e_j)} \quad (10)$$

where, α_t stands for the attention score.

III. EXPERIMENTAL RESULTS

A. Data

We train and evaluate the model on an open dataset with the wake-up word “Olivia”(<https://drive.google.com/file/d/1m4HIsclvCDbxflU51utMUczcKRZdWv7j/view>). The corpus training set consists of 1544 spoken examples, with the keyword “Olivia”. The test set consists of 550 spoken examples, with the keyword “Olivia” and other command statements, such as: Olivia turn the volume to fifty percent.

We also downloaded 15k audio containing noise and speech from the web. We divide the data set into 8k negative test examples and 7k negative train examples.

B. Experiment Setup

Our model is trained with the Adam optimizer [17] and decayed the learning rate after converged from 1×10^{-3} to 1×10^{-4} . The gradient norm clipping to 1 is added to the model and L2 weight decay 1×10^{-5} . We use cross-entropy loss function to reduce the deletion error. The model parameters are trained for 200 epochs with a minibatchsize of 32. The size of the attention mechanism is 100. The number of hidden layer unit in BiGRU is 64.

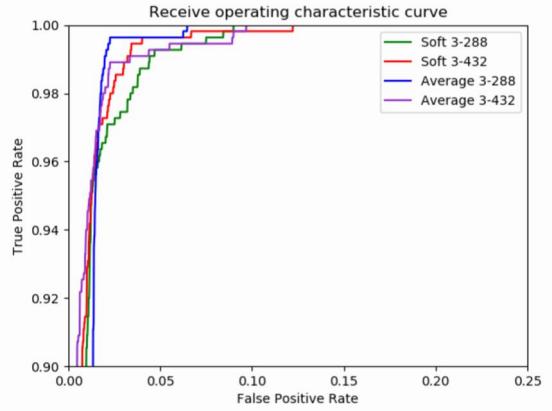


Figure 3. ROCs for soft attention and average attention.

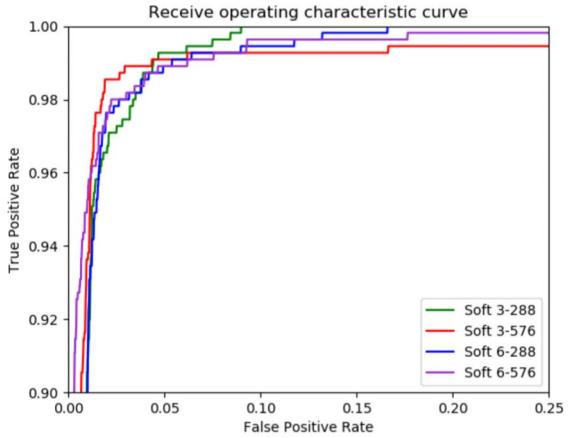


Figure 4. ROCs for different TDNN steps and different TDNN units.

C. Results

We apply the TDNN-BiGRU structure on the “Olivia” data set. Our KWS baseline system is an end-to-end KWS architecture trained with the Connectionist Temporal Classification loss function in [18]. Our model with 3 steps and 288 units is superior to the baseline. As shown in Table 2, at the false positive rate of 5%, the true positive rate of our model reached 99.63%, which was higher than the baseline of 98.1%. We explored the effect of the number of TDNN units on the accuracy of the model. The results are shown in Figure 4 and Table 1. We observed that as the number of units increases, the true positive rate increases first and then decreases. This indicates that

Table I

PERFORMANCE OF THE PROPOSED MODEL, INCLUDING PRECISION (HIGHER IS BETTER IN ALL INDEXES), RECALL, ACCURACY AND F1. WE COMPARED THE FOLLOWING PARAMETERS: 1) THE AVERAGE ATTENTION MECHANISM AND THE SOFT ATTENTION MECHANISM. 2) TDNN STEP SIZE. 3) TDNN UNITS

Attention	TDNN Step	TDNN Unit	Accuracy	Precision	Recall	F1
Average attention	3	288	97.64%	88.05	76.36	81.79
	3	432	98.41%	96.70	79.82	87.45
	3	576	98.52%	93.56	84.55	88.83
Soft attention	3	288	98.16%	91.39	81.09	85.93
	3	432	98.48%	91.99	85.64	88.70
	3	576	98.59%	92.77	86.36	89.45
Soft attention	6	288	98.17%	90.26	82.55	86.23
	6	432	98.47%	92.81	84.55	88.49
	6	576	98.70%	98.48	82.55	89.81

there is an optimal number of units to maximize the true positive rate. However, the recall rate, precision rate and F_1 are almost increasing. Increasing the number of units will convey more hidden information, which is beneficial to the training of network parameters. It can improve the performance of the network.

From Table 1 and Figure 4, we can clearly observe the impact of TDNN step size on model performance. When there are fewer units, the large step size will increase the accuracy, the recall rate and the F_1 value of the model. It is because when the step size increases, the TDNN combines more context information to extract features with less information loss. But the amount of calculations in the system will double. When the number of units increases, the recall rate of the model with a smaller step is higher. However, the precision rate of the model with a larger step is higher. When the step is large, the system combines more context information and the information feature extraction is better. Therefore, there is less misjudgment of the examples with keyword when testing and the false positive rate is low.

Figure 3 shows the roc curve of two attention mechanisms. Table 2 compares the true positive rate of different attention mechanisms. As can be observed, when the number of TDNN units is small, the accuracy and true positive

number of units of TDNN, the true positive rate and the accuracy are similar, and the gap between the precision and the recall is also decreasing. This indicates that the increase of units in TDNN can reduce the impact of the attention mechanism on the results. Therefore, it is possible to consider reducing the amount of calculation by increasing the number of units in the future through using the average attention mechanism.

IV. CONCLUSION

In this paper, we proposed an end-to-end model based on TDNN-BiGRU for keyword spotting. The TDNN-BiGRU-based system has excellent performance on the Olivia dataset compared to the end-to-end model trained with the CTC loss function. Our model consists of three parts: TDNN, BiGRU and attention mechanism. We explored the step size and number of units of TDNN. Experiments show that More units and longer steps can improve the performance of the model. We also explored the impact of the average attention mechanism and the soft attention mechanism on the accuracy of the model. When the number of units is small, the soft attention mechanism works better than the average attention mechanism. When the number of units increases, the two attention mechanisms are not much different. Finally our end-to-end system based on TDNN-BiGRU achieves a true positive rate of 99.63% at a 5% false positive rate.

REFERENCES

- [1] J. Schalkwyk, D. Beeferman, F. O. Beaufays, B. Byrne, C. Chelba, M. Cohen, M. Kamvar and B. Strope “Your word is my command: Google search by voice: A case study,” In *Speech Recognition*, Springer, pp. 61–90, 2010.
- [2] J. R. Rohlíček, W. Russell, S. Roukos and H. Gish, “Continuous hidden Markov modeling for speaker-independent wordspotting,” In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 627–630, 1990.
- [3] R. C. Rose and D. B. Paul, “AhiddenMarkovmodel based keyword recognition system,” In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 129–132, 1990.,

rate of the average attention mechanism model are higher than the soft attention mechanism. However, the precision and recall of the average attention mechanism are lower than the soft attention mechanism. When increasing the

- [4] J. G. Wilpon, L. G. Miller and P. Modi, “Improvements and applications for key word recognition using hidden Markov modeling techniques,” In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 309–312, 1991.
- [5] D. R. Miller, M. Kleber et al, “Rapid and accurate spoken term detection,” In *ISCA INTERSPEECH*., pp. 314–317, 2007.
- [6] S. Parlak and M. Saraclar, “Spoken term detection for Turkish broadcast news,” In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 5244–5247, 2008.
- [7] G. Chen, C. Parada and G. Heigold, “Small-footprint keyword spotting using deep neural networks,” In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4087–4091, 2014.
- [8] T. N. Sainath and C. Parada, “Convolutional neural networks for small-footprint keyword spotting,” In *ISCA INTERSPEECH*., pp. 1478–1482, 2015.
- [9] M. Sun, A. Raju, G. Tucker, S. Panchapagesan et al “Max-pooling loss training of long short-term memory networks for small-footprint keyword spotting,” In *IEEE Spoken Language Technology Workshop (SLT)*., pp. 474–480, 2016.
- [10] R. Tang and J. Lin, “Deep residual learning for small-footprint keyword spotting,” In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5484–5488, 2018.
- [11] S. O. Arik, M. Kliegl, R. Child, J. Hestness, A. Gibiansky, C. Fougner, R. Prenger and A. Coates, “Convolutional recurrent neural networks for small-footprint keyword spotting,” In *arXiv preprint arXiv:1703.05390*., 2017.
- [12] V. Nair and G. E. Hinton, “Rectified linear units improve restricted Boltzmann machines,” In *Proc. 27th Int. Conf. Mach. Learn. (ICML)*., pp. 1–8, 2010.
- [13] X. Lei, A. Senior, A. Gruenstein and J. Sorensen, “Accurate and compact large vocabulary speech recognition on mobile devices,” In *ISCA INTERSPEECH*., 2013.
- [14] D. E. Rumelhart and J. L. McClelland “Parallel Distributed Processing,” In *Parallel distributed processing (Vol. 2)*., Cambridge, MA:: MIT press, 1987.
- [15] S. Makino and K. Kido. “Phoneme recognition using time spectrum pattern,” In *Speech Commun.*, pp. 225–237, 1986.
- [16] F. Chowdhury, Q. Wang, I. L. Moreno and L. Wan. “Attentionbased models for text-dependent speaker verification,” In *arXiv preprint arXiv:1710.10470*, 2017.
- [17] D. P. Kingma and J. Ba. “Adam: A method for stochastic optimization,” In *arXiv preprint arXiv:1412.6980*, 2014.
- [18] C. Lengerich, A. Hannun. “An end-to-end architecture for keyword spotting and voice activity detection,” In *arXiv preprint arXiv:1611.09405*, 2016.

Coarse-to-Fine Document Ranking for Multi-Document Reading Comprehension with Answer-Completion

Hongyu Liu
*School of Computer Science
 and Technology
 Beijing Institute of Technology
 Beijing, China
 liuhongyu@bit.edu.cn*

Shumin Shi ✉
*School of Computer Science
 and Technology
 Beijing Institute of Technology
 Beijing, China
 bjssm@bit.edu.cn*

Heyan Huang
*School of Computer Science
 and Technology
 Beijing Institute of Technology
 Beijing, China
 hhy63@bit.edu.cn*

Abstract—Multi-document machine reading comprehension (MRC) has two characteristics compared with traditional MRC: 1) many documents are irrelevant to the question; 2) the length of the answer is relatively longer. However, in existing models, not only key ranking metrics at different granularity are ignored, but also few current methods can predict the complete answer as they mainly deal with the start and end token of each answer equally. To address these issues, we propose a model that can fuse coarse-to-fine ranking processes based on document chunks to distinguish various documents more effectively. Furthermore, we incorporate an answer-completion strategy to predict complete answers by modifying loss function. The experimental results show that our model for multi-document MRC makes a significant improvement with 7.4% and 13% respectively on Rouge-L and BLEU-4 score, in contrast with the current models on a public Chinese dataset, DuReader.

Keywords-multi-document reading comprehension; document ranking; answer prediction;

I. INTRODUCTION

Machine Reading Comprehension (MRC) is one of the most important natural language processing tasks, which tests the ability of machines to read and understand a document and then answer some questions about the given document. The release of the Stanford Question Answering Dataset (SQuAD) [1] is one of the first large reading comprehension dataset which requires machine to select a text span in the document as an answer to the question. The SQuAD has attracted many researchers in NLP field and has facilitated more research on RNN-based and attention mechanism models and more complicated datasets.

The traditional MRC architecture requires customized question and document modeling layers to capture the interaction between questions and given documents. In recent years, various attention mechanisms have been proposed to obtain better question-awareness document representation [2], [3]. Since last year, Bidirectional Encoder Representations from Transformers (BERT) [4] introduces a powerful text representation architecture that performs very well on multiple NLP tasks, especially in MRC task.

On the other hand, the creation of larger and more complex datasets also contributes to the development of MRC. Compared to SQuAD, the recently proposed datasets [5], [6] are based on more realistic scenarios,

such as multi-document reading comprehension. In the meantime, Chinese reading comprehension has attracted more attention.

In this paper, We demonstrate the validity of the proposed model on a Chinese multi-document reading comprehension dataset, DuReader [5]. The main contributions can be summarized as follow:

1. We introduce a coarse-to-fine document ranking approach to rank documents with different granularity.
2. We propose an answer-completion strategy to empower model to focus on predicting more complete answer.
3. We conduct sufficient experiments on DuReader. The results show that our model achieves great performance on multi-document MRC task.

The remainder of this paper is organized as follows: The second section introduces the latest progress of machine reading comprehension. Detailed of our model will be given in the third section. In the fourth section, we compare the experiments results of our model with other baselines to prove the validity of our proposed model. In the final section, we summarize our work and propose future research directions.

II. RELATED WORK

Since the release of the SQuAD, many researchers have proposed a variety of models. Seo et al. [2] first introduced the idea of bidirectional attention which can compute question-awareness document representation and document-awareness question representation simultaneously for detecting the soft alignment between questions and documents. Wang et al. [3] proposed multi-granularity hierarchical attention and the model surpassed human performance on SQuAD.

In addition to MRC task, attention mechanism plays an important role in many tasks. Google proposed BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [4] in October 2018 which achieved the state-of-the-art for eleven NLP tasks including MRC tasks.

Compared with SQuAD, real-world scenarios have multiple documents for each question so that it requires the machine to have the ability to filter irrelevant documents and aggregate all relevant document information to get the final answer. As a result, there are many multi-document

datasets have been released [5], [6]. Some researchers views document ranking as an effective way to solve multi-document reading comprehension [7], [8]. Previous works rank documents at document or paragraph level and utilize distant supervision to distinguish the positive and negative examples (a document or paragraph containing a text span that exact match with ground truth answer will be considered as a positive sample, otherwise it will be considered as a negative sample). However, the length of the answer is relatively longer so that other documents can not exact match with the ground truth answer except the ground truth document, it is difficult to determine whether a document is a positive or negative example. At the same time, researchers pay more attention to new model architecture which can take multiple documents as input. At present, there are two categories of the current approaches: one is the pipeline-based approaches [9], [10], the other is the multi-task approaches [11], [12]. The pipeline approach firstly gives each document a ranking score and then passes all documents to the MRC model, the final answer is generated by the joint score of the document and the candidate answers. The multi-task approaches can select documents and answers simultaneously which the model can capture richer semantic information from the multi-task shared encoding layers.

There have been some works on Chinese multi-document machine reading comprehension. Wang et al. [12] focused on answer prediction and proposed to model answer content and conduct cross-passage answer verification. Yan et al. [13] introduced a deep cascade model to balance effectiveness and efficiency in real-world scenarios. Liu et al. [14] integrated minimum risk training into their model to mitigate the loss deviation caused by the answer appearing in the document multiple times during training phase.

III. OUR MODEL

In this section, we first introduce BERT architecture for reading comprehension that we used as our backbone network. Then we present our model which consider both document-aspect and answer-aspect for multi-document MRS task. Figure 1 shows our model architecture.

A. BERT Architecture for Reading Comprehension

BERT is a multi-layer bidirectional encoder representation from transformers that can be trained on a large corpus with two novel unsupervised pre-training tasks. In this paper, we use BERT as our backbone network.

1) *Input Representation*: In multi-document reading comprehension scenario, one question q corresponds to a set of documents $D = \{d_1, \dots, d_{N_D}\}$. Because BERT limits the maximum input length, we use each document as the input to our model respectively instead of concatenating all documents into a new document. The input is each question q and one corresponding document d_i and finally we get N_D candidate answers for each question.

Although we don't concatenate all documents into one document, some documents are still exceed the input limit.

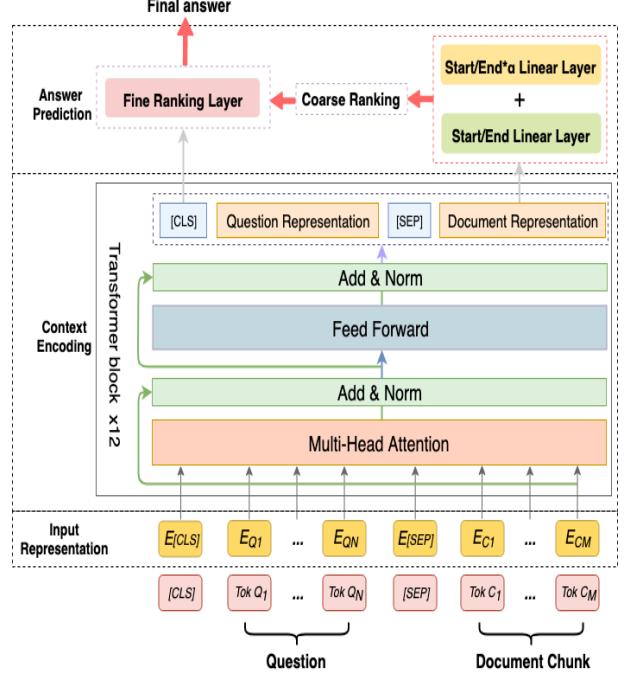


Figure 1. Model architecture with coarse-to-fine ranking and answer-completion

Following [4], we slide a window of length l with a stride s over the document d_i and produce several document chunks $C_i = \{c_{i1}, \dots, c_{iN_C}\}$, where $N_C = \lceil \frac{L_{d_i}-l+s}{s} \rceil$ is the number of document chunks for document d_i and L_{d_i} is the length of the document.

The input sequence consists of a [CLS] token, the tokenized question q , a [SEP] token, the tokenized document chunk c and a [SEP] token, where [CLS] is a classification token and [SEP] is another token for differentiating sentences. For each token in questions and document chunks, its input representation is constructed by summing the token, segment and position embeddings. We recommend readers to read [4] for more details.

2) *Context Encoding*: We use 12 layers of transformer blocks to encode question, document chunk and the interaction between the question and document chunk simultaneously. Due to space limitations, we will omit the description of the transformer architecture. The details of the transformer block can be found in the original paper [15]. We denote the document chunk representation in the transformer output layer as C .

3) *Answer Prediction*: There are two simple linear layers followed by the transformer blocks to predict the probability of each token in the context being the start or the end token of an answer span. More specifically, the unnormalized score and probabilities distribution of start and end position are modeled as:

$$g^s = W_s C, p^s = \text{softmax}(g^s) \quad (1)$$

$$g^e = W_e C, p^e = \text{softmax}(g^e) \quad (2)$$

where p^s and p^e are the probabilities for start and end position of answer spans, and W_s and W_e are trainable parameter matrices.

The objective function is defined as the cross-entropy of the predicted probability indexed by true start and end indices, averaged over all the training samples:

$$L_{ANS} = -\frac{1}{N} \sum_{i=1}^n [\log p^s(y_{start}^i) + \log p^e(y_{end}^i)] \quad (3)$$

where N is the number of samples in the dataset and y_{start}^i, y_{end}^i are the gold start and end positions.

During inference, the probability for a text span from token i to token j being the answer is given by:

$$p(i, j) = \text{softmax}(g^s(i) + g^e(j)) \quad (4)$$

For each document corresponding to one question, every chunk from that document will produce a candidate answer with an answer probability. We take the highest probability answer of all candidate answers from all chunks as the final prediction for current document. In the end, we obtain N_D candidate answers for each question and denote $P(A_i)$ as the probability of the answer from the i^{th} document.

B. Document Ranking

Document ranking is a sub-task in multi-document reading comprehension which aims to rank the documents to get better answer from the candidate answers. Previous approaches first use statistical information or shallow semantic information to prune or filter irrelevant documents, and then utilize deep semantic information to rank the remaining documents to distinguish the correlation between documents and questions. In this work, we propose an coarse-to-fine method to directly rank documents using statistical or shallow semantic information and deep semantic information.

1) *Coarse Document Ranking*: Each question and corresponding documents in DuReader come from the Baidu search engine. Each question corresponds to five documents. The order of the documents in DuReader is the actual order in the search engine. When people use search engines in their daily lives, they will pay more attention to the top search results, because the more advanced search results are more relevant to the question that can provide more valuable information. We believe that search engines can reflect the statistical or shallow semantic information between questions and documents to some extent.

Therefore, we count the frequency at which the answer appears in each document in the training set. The statistical result is shown in Table I. As a result, we set different weights as documents prior knowledge for the answers from different documents to represent the importance of the documents, rather than treating each document equally. We denote the i^{th} document prior probability as dp_i which is derived from the statistics of the training set. We believe it can reflect statistical or shallow semantic

Table I
THE FREQUENCY AND PROPORTION OF ANSWERS APPEAR IN DIFFERENT DOCUMENTS

	Frequency	Proportion
Doc1	114941	44.5%
Doc2	60240	23.3%
Doc3	40230	15.6%
Doc4	25060	9.7%
Doc5	18004	6.9%

information between the question and the document. The probability for a candidate answer from the i^{th} documents is updated by:

$$P(A_i) = \frac{e^{dp_i \cdot P(A_i)}}{\sum_{n=1}^{N_D} e^{dp_n \cdot P(A_n)}} \quad (5)$$

At the same time, we should note that the above search engine order is not always available in other datasets. So we use the normalized token-level *Recall* value between the question and document title as the prior probability of document to verify the validity of our hypothesis, that is, statistical or shallow semantic information can also be used for ranking document.

2) *Fine Document Ranking*: Previous works rank documents at document or paragraph level and utilize distant supervision to distinguish the positive and negative examples. Our work is different from them. First, only one document chunk is visible to the model at a time. In order to distinguish semantic information at a finer granularity, we decide to rank documents at chunk-level. Furthermore, the length of the answer is relatively longer (i.e. the average answer length is greater than 100 words) in Chinese multi-document reading comprehension, we can't utilize distant supervision to label the positive samples because there is no document have a text span exact match with the ground truth answer except the ground truth document. Therefore, we propose a novel document ranking method suitable for Chinese multi-document reading comprehension. There are N_D documents for each question. For the document where the ground truth answer is located, the chunk from that document containing the ground truth answer is labeled as a positive sample. And for other documents, the document chunk with the smallest $F1$ value for ground truth answer is labeled as a negative sample. Then the document ranking can be viewed as two-classification task. Suppose that C_{ls} is the hidden representation of the token [CLS] from the transformer output, which can be viewed as the deep semantic representation of the question and input document chunk. The probability cp that the input document chunk is labeled as a positive sample is modeled as:

$$cp = \text{softmax}(W_c C_{ls}) \quad (6)$$

where W_c is trainable parameter matrix.

we use the cross-entropy loss as the document ranking objective:

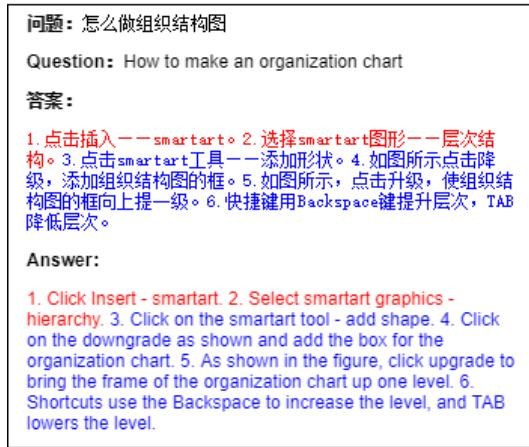


Figure 2. An incomplete answer example in DuReader

$$L_{DOC} = -\frac{1}{K} \sum_{i=1}^k \log(cp) \quad (7)$$

where K is the number of chunks we selected from the dataset.

we denote cp_i as the chunk probability of the document chunk from the i^{th} document. The probability for a candidate answer from the i^{th} documents is further updated by:

$$P(A_i) = \frac{e^{cp_i \cdot P(A_i)}}{\sum_{n=1}^{N_D} e^{cp_n \cdot P(A_n)}} \quad (8)$$

C. Answer-completion Strategy

1) *End token more important:* We discover that in cases where the truth answer contains multiple sentences, the model sometimes only predict the first few sentences. Figure 2 shows a qualitative example of this phenomenon. The model only predicts the red answer and discards blue answer. In order to address this problem, we modify the loss function (Eq.3) of the model so that the model can pay more attention to the end token of answers in the training phase. We set the hyperparameter α to indicate the importance of the end token. The modified unnormalized score, probabilities distribution of start and end position as well as the loss function is given by:

$$g^{s'} = W'_s C, p^{s'} = softmax(g^{s'}) \quad (9)$$

$$g^{e'} = W'_e C, p^{e'} = softmax(g^{e'}) \quad (10)$$

$$L'_{ANS} = -\frac{1}{N} \sum_{i=1}^n [\log p^{s'}(y_{start}^i) + \log p^{e'}(y_{end}^i) \cdot \alpha] \quad (11)$$

It has been experimentally verified that the modified loss function may hurt the performance of our model correctly predicts the start token. So we jointly consider

the impact of two loss functions on our model (i.e. we use two different prediction layer), the probability for a text span from token i to token j being the answer (Eq.4) is modified by:

$$p(i, j) = softmax(g^s(i) + g^e(j) + g^{s'}(i) + g^{e'}(j)) \quad (12)$$

D. Training and Inference

During training, we take a multi-task learning approach [16], sharing the parameters of transformer blocks with a joint objective function defined as:

$$L = L_{DOC} + L_{ANS} + L'_{ANS} \quad (13)$$

All samples are used to train document ranking module, but only the positive samples are passed to subsequent modules for training the prediction layer, the negative samples are discarded.

During inference, we take the coarse ranking probability dpi , fine ranking probability cpi , and answer probability $P(A_i)$ into account. The final answer probability is calculated by Eq.12, Eq.5 and Eq.8. We compare the probability across all chunks from the same instance, and choose the final answer according to the final probability.

IV. EXPERIMENT AND ANALYSIS

A. Dataset

We evaluate our model on a Chinese multi-document comprehension reading dataset, DuReader¹. DuReader is the largest Chinese MRC dataset, which contains 200k question, 1M documents and more than 420k human-annotated answers. Each question has 5 evidence documents. All the questions and documents comes from Chinese search engine Baidu.

B. Data Preprocessing

Each document contains several paragraphs. We calculate the maximum $F1$ value of the corresponding question at paragraph level for each document. Then we select the top-N paragraphs that have largest $F1$ value with each question. These paragraphs are reassembled into a new pruned document in the order of the original document. Through this preprocessing, a large number of irrelevant paragraphs can be filtered out.

C. Evaluation Metrics

According to the evaluation metrics of the DuReader, we evaluate our model via character-level BLEU-4 and Rouge-L.

D. Baselines

Because there haven't any published work research on DuReader based on BERT. We report the experimental results on DuReader in the published work and construct BERT baseline² ourselves without document ranking and answer-completion strategy.

¹<https://github.com/baidu/DuReader>

²<https://github.com/huggingface/pytorch-transformers>

Table II
PERFORMANCE OF OUR MODEL AND COMPETING MODELS ON THE DUREADER

Model	Rouge-L	BLEU-4
BiDAF[2]	39.0	31.8
PR+BiDAF[12]	41.81	37.55
V-NET[12]	44.18	40.97
Deep Cascade[13]	50.71	49.39
MRT[14]	51.09	43.76
BERT	49.61	44.62
Our model	53.27	50.42
Our model*	53.13	54.63
Human performance	57.4	56.1

E. Implementation Details

We use BERT as our baseline. We set up several experiments with the default BERT models configurations. And finally we set hyper-parameters for our proposed model which have the best Rouge-L performance. We use Adam optimizer with a learning rate of 3e-5 and warmup over the first 10% steps to train for 2 epochs. The batch size is 6, the document chunk length is 512 and the document stride length is 128. The document prior probability is 0.45, 0.23, 0.16, 0.09, 0.07 for each document, the values are from the statistical information of the training set. The end token importance hyperparameter α is 2.

F. Main Results

The results of our model on DuReader are summarized in Table II.

Because the DuReader *test set 1.0* is temporarily closed, we show the performance of our model on DuReader development set and the performance of other published models are tested on DuReader *test set 1.0*. According to the previously published data, the test set and development set have similar performance. As we can see in Table II, BERT baseline achieves comparable performance to the currently published models. The coarse-to-fine document ranking and answering strategy we proposed on our model further improve the performance, which improve 7.4% in RougeL-L and 13% on BLEU-4. The last row in Table II is the performance of our model on DuReader *test set 2.0*. DuReader *test set 2.0* consists of the difficult samples for current models in *test set 1.0*, which is more challenging than *1.0*. Our model also achieves great performance on *test set 2.0* which can demonstrate the validity of our proposed model. We will analyze our model in detail in the next section.

G. Ablation Study

To further study the effectiveness of our model, we conduct an in-depth ablation study on the development set of DuReader, which is shown in Table III.

We first evaluate the answer-completion strategy by ablating the modified loss function so that the modified loss will not be used during training and testing. Then we remove the fine and coarse ranking in order to test the necessity of the coarse-to-fine ranking.

Table III
ABLATION STUDY ON DUREADER DEVELOPMENT SET

Model	Rouge-L	Δ	BLEU-4	Δ
Our model	53.27	-	50.42	-
- answering strategy	52.56	-0.71	49.67	-0.75
- fine ranking	52.11	-0.45	49.02	-0.65
- coarse ranking	49.61	-2.5	44.62	-4.4

Table IV
COMPARISON AMONG DIFFERENT DOCUMENTS AND DIFFERENT RANKING APPROACH ON DUREADER DEVELOPMENT SET

	Rouge-L	BLEU-4
Doc1	50.18	48.27
Doc2	44.4	41.31
Doc3	40.42	35.87
Doc4	36.11	30.2
Doc5	34.02	27.97
Multi-doc	49.61	44.62
Coarse ranking (Re)	50.83	48.33
Coarse ranking (PK)	52.11	49.02
PK + fine ranking	52.56	49.67

From Table III, we can see that the coarse-to-fine ranking can help the model to distinguish between different documents and focus on the documents related to the question. The coarse ranking can effectively help model filter irrelevant documents and the fine ranking can further improve the performance because it can capture deep semantic relation between questions and documents which maybe ignored in the coarse ranking. We will further analyze the document ranking and answer-completion strategy respectively in the following section.

H. Document Ranking

As is shown in Table IV, we evaluate the performance of the model on different ranking method. In order to verify our hypothesis (i.e. the order of search engines can cover some statistical or shallow semantic information), we evaluate our model on different documents. We can see that the performance of the model using the last document is much lower than using the first document which can indicate the top ranked documents can provide more reliable answers. And equally considering all the documents without any document ranking does not work well because the model can't filter the irrelevant documents very well. After adding the document prior probability as prior knowledge (PK), the performance of the model can be effectively improved. But the prior knowledge is not always available in other datasets. We also experiment with a simple coarse ranking method (Re) that rank documents by using the *Recall* value between question and the document title to prove the validity of the coarse ranking. We can see that although the performance is a little worse than prior knowledge, because the order of the search engines involves complex ranking mechanisms which can provide us with a good coarse ranking effect. However, it is still better than the performance without ranking. After adding the fine ranking, the model achieves the best results which shows that the coarse ranking and

Table V
THE IMPACT OF THE ANSWER-COMPLETION STRATEGY ON ROUGE-L SCORE ON DUREADER DEVELOPMENT SET

	Zhida			Search		
	D	E	Y_N	D	E	Y_N
W/O	53.1	58.8	42.6	51.1	51.9	36.6
W/	53.8	59.5	44.4	51.5	53.1	38.1

the fine ranking can capture the correlation between question and documents at different granularity. The former is statistical information or shallow semantic information and even some additional information that search engines can provide, while the latter is mainly deep semantic information.

I. Answer-completion Strategy

In this section, we discuss in depth the effectiveness of our proposed answering strategy on Rouge-L score. We believe that the modified loss function can make the model pay more attention to the end token of the answer during the training phase. DuReader can be divided into two types according to the source of the data, i.e. *Zhida* and *Search*. The question types for each subset include *Description*, *Entity* and *Yes-No*. We evaluate the model with and without answering strategy on these six subsets. The results are shown in Table V.

In Table V, W/O and W/ indicates whether the model is combined with the answering strategy or not. D, E, Y_N represent different question types. We can see that the performance of the model with answer-completion strategy is better than the model without it on all six subsets. It demonstrate the answering strategy we proposed can further improve the performance of our model. And we can noticed that for both *Zhida* and *Search*, where the *Yes-No* question type has the largest improvement. Because for the *Yes-No* question, in addition to predict the answer, we need to predict the category of the views based on the predicted answer (i.e. *Yes*, *No* and *Depend*). We believe that a more complete answer can further improve the performance of the classifier, so the *Yes-No* question type has the largest improvement.

V. CONCLUSION

In this paper, we consider both document-aspect and answer-aspect for multi-document reading comprehension. First of all, we propose a coarse-to-fine document ranking based on document chunks to measure the relevance of questions and documents at different granularity. And then the answer-completion strategy can enable our model to pay more attention to the end token for detecting one entire answer. The experimental results demonstrate that our model achieves great performance in the multi-document MRC task. We will focus on integrating external knowledge for multi-document MRC task in future work.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (No.61671064) and the

National Key RD Program of China under Grant (No.2018YFC0831704).

REFERENCES

- [1] Rajpurkar, Pranav, et al. Squad: 100,000+ questions for machine comprehension of text. arXiv preprint arXiv:1606.05250, 2016.
- [2] Seo, Minjoon, et al. Bidirectional attention flow for machine comprehension. arXiv preprint arXiv:1611.01603, 2016.
- [3] Wang, Wei, Ming Yan, and Chen Wu. Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. arXiv preprint arXiv:1811.11934, 2018.
- [4] Devlin, Jacob, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.
- [5] He, Wei, et al. Dureader: a chinese machine reading comprehension dataset from real-world applications. arXiv preprint arXiv:1711.05073, 2017.
- [6] Nguyen, Tri, et al. MS MARCO: A human generated machine reading comprehension dataset. arXiv preprint arXiv:1611.09268, 2016.
- [7] Htut, Phu Mon, Samuel R. Bowman, and Kyunghyun Cho. Training a ranking function for open-domain question answering. arXiv preprint arXiv:1804.04264, 2018.
- [8] Lee, Jinyuk, et al. Ranking Paragraphs for Improving Answer Recall in Open-Domain Question Answering. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018: 565-569.
- [9] Chen, Danqi, et al. Reading Wikipedia to Answer Open-Domain Questions. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017: 1870-1879.
- [10] Wang, Shuhang, et al. R3: Reinforced Ranker-Reader for Open-Domain Question Answering. Thirty-Second AAAI Conference on Artificial Intelligence. 2018: 5981-5988.
- [11] Clark, Christopher, and Matt Gardner. Simple and Effective Multi-Paragraph Reading Comprehension. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2018: 845-855.
- [12] Wang, Yizhong, et al. Multi-Passage Machine Reading Comprehension with Cross-Passage Answer Verification. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2018: 1918-1927.
- [13] Yan, Ming, et al. A Deep Cascade Model for Multi-Document Reading Comprehension. Thirty-Third AAAI Conference on Artificial Intelligence. 2019: 7354-7361.
- [14] Liu, Jiahua, et al. A Multi-answer Multi-task Framework for Real-world Machine Reading Comprehension. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018: 2109-2118.
- [15] Vaswani, Ashish, et al. Attention is all you need. Advances in neural information processing systems. 2017: 5598-6008.
- [16] Caruana, R. Multitask learning: A knowledge-based source of inductive bias. Machine Learning. 1997: 41-75.

Developing a machine learning-based grade level classifier for Filipino children's literature

Joseph Marvin Imperial¹, Rachel Edita Roxas², Erica Mae Campos,
 Jemelee Oandasan, Reyniel Caraballo, Ferry Winsley Sabdani
 College of Computing and Information Technologies
 National University
 Manila, Philippines
 {jrimperial¹, reoroxas²}@national-u.edu.ph

Ani Rosa Almario
 Adarna House, Inc.
 Quezon City, Philippines
 ani@adarna.com.ph

Abstract—Reading is an essential part of children’s learning. Identifying the proper readability level of reading materials will ensure effective comprehension. We present our efforts to develop a baseline model for automatically identifying the readability of children’s and young adult’s books written in Filipino using machine learning algorithms. For this study, we processed 258 picture books published by Adarna House Inc. In contrast to old readability formulas relying on static attributes like number of words, sentences, syllables, etc., other textual features were explored. Count vectors, Term Frequency–Inverse Document Frequency (TF-IDF), n-grams, and character-level n-grams were extracted to train models using three major machine learning algorithms—Multinomial Naïve-Bayes, Random Forest, and K-Nearest Neighbors. A combination of K-Nearest Neighbors and Random Forest via voting-based classification mechanism resulted with the best performing model with a high average training accuracy and validation accuracy of 0.822 and 0.74 respectively. Analysis of the top 10 most useful features for each algorithm show that they share common similarity in identifying readability levels—the use of Filipino stop words. Performance of other classifiers and features were also explored.

Keywords—Filipino, storybook, readability, machine learning, classification

I. INTRODUCTION

“*Pagbasa...pag-asa.*” (*In reading, there is hope.*)

—Reading Association of the Philippines

Readability is the difficulty or complexity of a published work [1, 2, 3, 4]. Over the years, researchers have studied and used different techniques to measure readability. Determining the readability of texts ensures that a certain book matches the reading ability of the target reader.

At present, over a hundred readability formulas have been developed for the English language. Traditional readability formulas use vocabulary and sentence difficulty as their basis in calculating text difficulty yet those proved to be misleading since other factors should be considered too [5, 6, 7]. Researchers are now trying out new approaches by applying different advancements in the field of Natural Language Processing and Computational Linguistics. Most of the existing readability measurements are used solely for the English language and when used in other languages, results yielded were invalid as shown by Kumiko, Satoshi and Hiroshi [8].

There is still no known and approved scale for measuring the readability of Filipino texts. Since the Filipino language has a structure different from English, different characteristics

should be considered and explored in determining how to measure its readability.

In this paper, the researchers developed a baseline prototype model for leveraging the readability of Filipino texts from an extensive experimentation of extracting various text-level features of picture book data such as word frequencies, character and word grams, and TF-IDF scores using machine learning algorithms. This study would benefit authors, publishers, educators, and readers as it aims to ensure that a certain reading material is appropriate for the intended users.

II. REVIEW OF RELATED LITERATURE

Text Readability Indexing (TRI) is the process where texts are evaluated and categorized into grade levels or scores based on their difficulty [2]. Readability assessment has been widely used for many purposes, primarily to determine how difficult a passage is to be read [9]. To be able to compute the readability of a certain text makes it easier for authors and publishers to ensure that they would be able to deliver ideas effectively to their target audience [5, 3, 10, 11]. Along with this, the reading ability of the reader should also be kept in mind to ensure effective measurement of text complexity [12].

It is important for the complexity of the learning material to match the reading ability of the student [1, 7]. When a student is given text with a difficulty level that does not match his reading level, the student may struggle in understanding and processing the content of the given text. But if his readability level matches that of the text, he can focus more and would probably perform better in school [10].

Without an accurate and approved system of measuring readability, some texts may not be applicable to students from a certain grade level. Because of this, teachers and educators learned to improvise for the benefit of their students [1]. Readability indexing systems should consider the learners’ reading abilities to ensure accuracy when used in academic reading and instructional materials [2]. Effective readability assessments would help educators prepare useful and proper reading materials for learners [7].

A. Factors Affecting Readability

According to Kumiko, Satoshi and Hiroshi [8], different features and indicators affect readability. Moreover, it is dependent on two factors: the reader (language capability, background knowledge of subject matter, and motivation) and the text (syntactic and semantic rules) as shown by Al-Tamimi et. al [3]. In semantic complexity, word familiarity, word difficulty, number of syllables per word, and number of words per sentence is a great contribution to existing

formulas. Certain text variables such as format, typography, content, literary style, vocabulary difficulty, sentence complexity, concept density, and cohesiveness all play a role as well [10, 12].

Text-related factors (e.g. word length, word frequency, vocabulary load, number of difficult words, average sentence length, sentence complexity, clarity of idea, use of topology or metaphors, and grammatical structure complexity) are considered in the calculation of readability [3]. According to Beinborn et. al [7], the complexity of a sentence's syntactic structure affects its difficulty. In conjunction with this, a sentence is said to be more readable when common words and terms are used in its construction.

While certain studies state that the longer the sentence and word, the more complex the text is, it can also be the other way around [4]. Graesser et al. [13] supported the above statement and said that texts with longer words and sentences are more difficult to read than shorter ones.

B. Traditional vs. Modern Approaches in Readability Identification

Despite the popularity of traditional readability formulas, these have received criticism on how they were unable to consider other factors such as text-based processing, situation levels, and sentence cohesion [5]. Their reliability and validity have been questioned since they only focused on structural characteristics and / or surface-based factors which can be misleading [4,5,6,7]. Features used by traditional readability formulas do not really analyze text at a deeper level, but have been popularly used.

Recent studies regarding readability greatly progressed from using traditional or superficial language properties to taking advantage of natural language processing tools to further analyze texts. The modern age gave way to improvements in the calculation of readability of texts. Newly developed readability formulas were proven to perform better [14]. Algorithms, machine learning techniques, frameworks, language modeling, and more recently developed technologies are creeping its way in computational linguistics and NLP.

Cognitive models are starting to be used in many researches as an approach to assessing texts [4]. Machine learning approaches in determining text readability allow for broader features to be considered [15]. Data training may lead machines to predict the complexity of unseen texts [16].

A study by Si and Callan [17] made use of the Expectation Maximization (EM) algorithm to compute for the weight value of their proposed models. The research produced two models: the unigram language model, which made use of words in text, and the sentence length distribution model. Based on the experiments performed, sentence length proved to be a useful predictor for readability since its mean values increases alongside the text's readability level while syllable count doesn't exhibit the same performance.

Advances in Natural Language Processing, Machine Learning, and language psychology delivered the field of

research a huge range of linguistic features; the ability to process large amounts of information; and awareness regarding deeper-lying text features that may have a great influence on the readability of a certain text [4].

C. On Readability of Filipino Texts

Readability formulas used for assessing texts in English proved to be unsuccessful in determining the readability of texts written in other languages. This is due to the different linguistic characteristics across the languages [1]. There is a high possibility that using a readability formula for a language would yield invalid and disappointing results when used for another language. There are other factors and elements to be considered that aren't specified in existing formulae [10].

The Philippines is a multilingual country that uses both English and Filipino as official languages. Both languages are formally taught in schools and used in different instructional materials and in literature. There are countries that have developed their own language's formula in assessing the readability of texts written in it. In the present time, there is no known scale that is extensively used for evaluating the difficulty of Filipino texts [1]. Since there is no formula that is recognized to determine the readability of Filipino texts yet, books and reading materials written in the language are categorized via subjective judgment [16]. There are Filipino researchers who have attempted to develop their own readability formulas [1].

In a research by Gutierrez [10], she used Fry and SMOG readability formulae in processing English and Filipino passages. The dataset was inputted into an online readability calculator with the application of the said formulas. The testing yielded invalid results during the Fry formula testing. Invalid results weren't shown after the SMOG formula testing. There may be a possibility that SMOG could be a suitable formula for Philippine English if certain adjustments are considered by Filipino authors. In the testing of texts written in Filipino, no valid result was produced by the two formulas which implies the need for a readability formula intended for Filipino texts.

III. METHODOLOGY

Described in this section is the step-by-step process of the study.

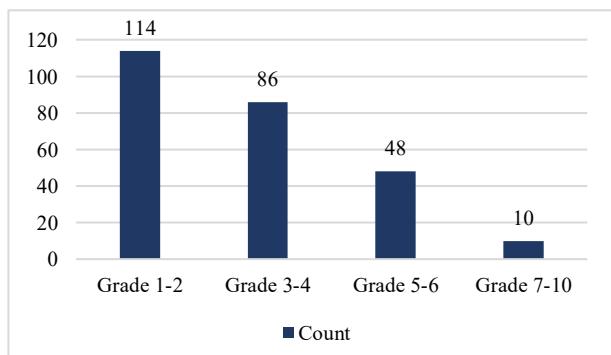


Fig. 1. Grouping categories per grade level

A. Data Collection and Annotation

The researchers collected a total of 258 pre-labelled picture book data from Adarna House Inc. from grade levels 1 to 10. The categorization of picture books according to reading age (category) was done by reading experts. The data was then partitioned to 80:20 ratio for training and testing respectively. All picture book data were written in Filipino. Due to the imbalanced dataset gathered, the researchers decided to group the classes to reduce the complexity of the data and to improve generalization. The clustered picture book grade levels are shown in Figure 2. Therefore, the final label for each test and validation data will be one of the four grouped classes.

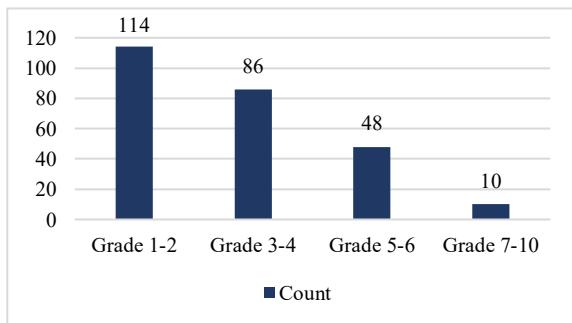


Fig. 2. Grouped picture book grade levels

In observance with the machine learning-based research methodology, the training set for each classifier was processed using select machine learning algorithms discussed below. On the other hand, the testing set was used to leverage the learning of the algorithms after training. Accuracy of the models was computed for training and testing. Lastly, the validation set will identify the reliability of the models in predicting unseen data.

B. Preprocessing

Before directly processing the raw data for the machine learning algorithms, several cleaning processes, including removal of numeric characters, symbols, extra spaces were performed to structure the data into uniform format and remove noise which may negatively affect the overall performance of the model.

C. Feature Extraction

Feature extraction is an essential step in machine learning-based research. It refers to the extraction of linguistic and textual elements from a given corpus to provide a representative, reduced sample of the content. Distinctive vocabulary “features” found in a document are assigned to the different categories by measuring the importance of those elements to the document content [26].

Count Vectors - a type of feature which converts text data into a matrix of token counts. It is a vocabulary construction feature where for each document in the dataset represented per row and each word in the document represented per column, the number of occurrences or frequency count of words are listed. The total size of the matrix or dictionary is the product of the number of documents and the number of unique words present in all documents.

Term Frequency – Inverse Document Frequency - TF-IDF is a common feature used in text analysis. It shows how important a word is to a document in a collection or corpus

[21]. The TF-IDF value increases to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, this caters to frequently occurring words by giving them lower weights than rare words.

Word Trigrams - Trigram are three slices of words or characters. For example, the trigram or three-word slice of the Filipino sentence “*Maganda ang bulalak sa hardin*” (The flowers in the garden are beautiful) would be “*Maganda ang bulalak*,” “*ang bulalak sa*,” “*bulalak sa hardin*.” For this study, the trigram TF-IDF feature represents the frequency of trigrams occurring in the document multiplied by the logarithm of the total number of documents divided by the number of trigrams appearing in it.

Character Trigrams - On the other hand, character trigrams are n-slices of characters of words. For example, the character trigram slice of the Filipino word “*mabuhay*” (greetings) would be “*_ma*,” “*mab*,” “*abu*,” “*buh*,” “*aha*,” “*hay*,” “*ay_*.” This feature represents the frequency of character trigrams occurring in the document multiplied by the logarithm of the total number of documents divided by the number of character trigrams appearing in it.

The proponents noted the use of trigrams instead of bigrams or unigrams because trigrams were often used in building language models of Filipino as shown by Oco, Syliongka, and Roxas [25].

D. Training

For this study, common machine learning algorithms commonly used for document classification and sparse datasets were experimented to see how they perform in identifying readability of Filipino texts. The algorithms are described below.

Multinomial Naïve-Bayes

The Naive Bayes classifier is based from the Bayes theorem. It has a ‘naive’ assumption of that a feature in a class is unrelated to any other feature present [22]. Given a sample story text file to be classified, represented by a vector $x = (x_1, \dots, x_n)$ representing some n features (independent variables), it assigns to this instance probabilities $p(C_k | x_1, \dots, x_n)$ for each of K possible outcomes in the set of classes C_K determined. In the present study, the researchers used Multinomial Naïve Bayes in training the converted data vectors (both count vectors and TF-IDF vectors). Multinomial Naïve Bayes implements the Naive Bayes algorithm for multinomially distributed data [24]. The formula for the multinomial Naïve Bayes is shown below:

$$C_{NB} = \operatorname{argmax} (\log P(C_k) + \sum_{i=1}^n x_i \log p_{ki})$$

where p_{ki} is the probability of a story belonging to class K and x_i is the features of the story.

K-Nearest Neighbors

For K-Nearest Neighbors or KNN, the textual features are plotted on a plane and its K nearest neighbors are searched using a given distance formula, in this study the Euclidean distance was used, where K is the number of features nearest to the feature being identified. The formula for getting the

distance between two feature points is shown below. The researchers set the value of k to 5 for each feature word.

$$D(u, v)^2 = \|u - v\|^2 = (u - v)^T(u - v) = \sum_{i=1}^d (u_i - v_i)^2$$

where u and v are sample real-valued features of a story being compared in terms of distance.

Random Forest

A Random Forest classifier creates multiple decision trees from a random subset of the training data. The classifier then gets the average of all decision trees generated to improve accuracy and control overfitting [24], this term is also called ‘bagging’. The researchers set the n-estimators (number of trees in the forest) to 100 and the random state (seed used by the random number generator) to 0. After training, predictions for an unseen story \hat{x} can be made by averaging the predictions from all the individual decision trees on the said story or by taking the ‘majority’ vote in the case of classification trees. The mathematical equation for this process is shown below.

$$\hat{f} = \frac{1}{B} \sum_{b=1}^B f_b(\hat{x})$$

where B is the number of bagging iterations during training and f_b is the current decision tree being trained.

IV. RESULTS AND DISCUSSION

The proponents conducted an extensive experimentation by pairing each feature set extracted (count vectors, TF-IDF, trigram, and character trigram) to the select machine learning algorithms (Multinomial Naïve Bayes, KNN, Random Forest) for document classification. The following tables, Table I, II, and III, describes the performance of each algorithm with varying features on identifying readability of story book test data. Cross fold validation during was performed with five instances ($k = 5$) to avoid bias and overfitting. Insights from the analysis of the experiments are described below.

For the experiment with Multinomial Naïve-Bayes, the best model came from using trigrams as features with an accuracy of 0.61. From this result, the researchers can infer that there are common trigram combinations in the Filipino language used for constructing storybooks that are prominent enough for the Naïve-Bayes algorithm to recognize what readability class they belong.

For K-Nearest Neighbors, the best model is using the count vectors as features. The researchers noted the stable value of accuracy in the training folds, with one instance reaching 0.857. The researchers note that the occurrence of words for each story is enough for the KNN algorithm to categorize which readability label a test storybook belongs to just by getting the first five (set value) nearest features (neighbors) for each word of the test story book.

On the other hand, for Random Forest, the best model achieved a high accuracy of 0.942 using count vectors as features, same with KNN described above. The Random Forest algorithm outperformed both Multinomial Naïve-Bayes and KNN in terms of performance during training. The researchers expected this result, as Random Forest is an

ensemble and decision-tree based algorithm which effectively reduces biases, control overfitting, and improves classification performance by aggregating the results from a collection of decision trees. This shows that the frequency of words for each storybook is enough for Random Forest to differentiate which grade level a storybook belongs to.

TABLE I. RESULT OF CLASSIFICATION WITH MULTINOMIAL NAÏVE BAYES (MNB) AND VARIOUS FEATURES

Algorithm & Feature	Training Folds					Average
	K=1	K=2	K=3	K=4	K=5	
MNB + Count Vectors	0.342	0.421	0.297	0.314	0.2	0.31
MNB + TF-IDF	0.5	0.368	0.514	0.514	0.486	0.48
MNB + Trigrams	0.605	0.474	0.622	0.629	0.657	0.6
MNB + Char Trigrams	0.421	0.421	0.432	0.429	0.429	0.43

TABLE II. RESULT OF CLASSIFICATION WITH KNN AND VARIOUS FEATURES

Algorithm & Feature	Training Folds					Average
	K=1	K=2	K=3	K=4	K=5	
KNN + Count Vectors	0.857	0.828	0.742	0.742	0.7428	0.7828
KNN + TF-IDF	0.371	0.6	0.4	0.314	0.3714	0.4114
KNN + Trigrams	0.457	0.628	0.45	0.371	0.514	0.485
KNN + Char Trigrams	0.342	0.485	0.314	0.285	0.314	0.348

TABLE III. RESULT OF CLASSIFICATION WITH RANDOM FOREST AND VARIOUS FEATURES

Algorithm & Feature	Training Folds					Average
	K=1	K=2	K=3	K=4	K=5	
RF + Count Vectors	0.942	0.857	0.685	0.885	0.685	0.811
RF + TF-IDF	0.742	0.742	0.771	0.8	0.54	0.72
RF + Trigrams	0.742	0.742	0.657	0.828	0.6	0.714
RF + Char Trigrams	0.857	0.74	0.685	0.685	0.714	0.737

With count vectors as a prominent feature for KNN and Random Forest, the researchers can infer from this result that the occurrence of words and its frequency is a great factor in determining readability of books. For instance, inflected and lengthy Filipino words like “nangagalaiti” (furious), “sumusubabaybay” (tracking or following), “nagtutulog-tulugan” (pretending to be asleep) were often found on higher grade levels (6-10) than lower grade levels. In addition, from using count vectors as feature, the researcher observe that higher grade levels have higher word count than storybooks from lower grade levels. All these patterns are taken in consideration by KNN and Random Forest, thus achieving a high accuracy score during training.

A. Applying a Voting Mechanism for Classification Enhancement

In the hopes of achieving a higher accuracy, the researchers implemented an additional ‘voting’ feature to combine the classification effectiveness of the two highest accuracy models utilizing the KNN and RF algorithm. A voting classifier model combines multiple different models into a single model, which is (ideally) stronger than any of the individual models alone as described in [24]. For soft voting, we predict the class labels based on the predicted probabilities p for a classifier. The mathematical formula for soft voting mechanism is shown below:

$$\hat{y} = \arg \max \sum_{j=1}^m w_j p_{ij}$$

where w_j is the weight that can be assigned to the j th classifier. Meanwhile in hard voting, we predict the class label \hat{y} via majority (plurality) voting of each classifier C_j [24]. The mathematical formula for hard voting mechanism is shown below:

$$\hat{y} = \arg \max \sum_{j=1}^m w_j X_A(C_j(x) = i)$$

where X_A is the characteristic function $[C_j(x) = i \in A]$ and A is the set of unique class or grade levels.

TABLE IV. RESULT OF CLASSIFICATION OF KNN AND RF WITH VOTING MECHANISM

Algorithm & Feature	Training Folds					Average
	K=1	K=2	K=3	K=4	K=5	
Soft Voting (KNN + RF with Count Vectors)	0.91	0.86	0.77	0.86	0.71	0.822
Hard Voting (KNN + RF with Count Vectors)	0.86	0.77	0.742	0.86	0.69	0.628

By using a soft-voting classifier combining the power of KNN, RF and count vectors, a high average accuracy of 0.822 was achieved. The proponents note a high score of 0.914 at the first random fold. It is also noted that incorporating a soft-voting classifier increased the current single-algorithm model accuracy by small 0.11 fraction. Using a hard-voting classifier produced a mediocre result of 0.628 in accuracy. With this, the proponent selected the model using a voting classifier combining KNN, RF and count vectors as the model to be used for predicting unseen story book data, or the validation set.

B. Validating on Unseen Data

The table below shows how the selected model performed on unseen data not included in the training. The voting-enhanced classifier identified the grade level of 23 books out of 31 from the validation set of 10% from the total storybook data count, equal to a percentage of 0.7419 against its 0.822 average accuracy on the testing data. The proponents deem this result as the highest validation score the model can achieve without overfitting. Overfitting is the result of model memorizing the training data too much instead of generalizing the shared features across categories.

TABLE V. PERFORMANCE ON UNSEEN, VALIDATION DATA

Algorithm & Feature	Books Classified Correctly	Total Number of Books	Percentage
Soft Voting (KNN + RF with Count Vectors)	23	31	0.7419

C. Analysis of Top Storybook Feature Words

While achieving a high accuracy score for training and validation is good, it is also essential to analyze the top features used by the algorithms to know how much they contributed in identifying the readability of the data and how it can be improved for future work.

Table VI shows the top performing and most useful in terms of identifying and discerning patterns from different grade levels. For each level, more than 80% of the words are Filipino stop words. Stop words refer to the common, connecting words used in the language. Usually these words are removed before preprocessing since they do not contribute in context according to Rajaraman and Ullman [21]. But for this research, it is discovered that Filipino stop words serve as a very important feature and can be used to identify the readability of Filipino story books.

V. CONCLUSION

The main purpose of research is to create a new method of automatically identifying the readability of Filipino literature, in the hopes of helping children find books suited for their reading ability. With data provided by Adarna House Inc., the researchers were able to explore various classification models using machine learning algorithms (Naïve-Bayes, KNN, and Random Forest) and features (count vectors, TF-IDF, trigrams, and character trigrams) present in the data. From the training, the best performing models were the combination K-Nearest Neighbors and Random Forest enhanced by a voting mechanism and using count vectors feature which obtained an average validation accuracy score of 74%. Analysis of features used show that Filipino stop words contribute greatly in terms of identifying readability. For future work, the researchers recommend the use of a larger data set with each category having the balanced data count to produce better predictions by thoroughly representing every grade level in the semantic space. With a larger set of data, more powerful methods like Deep Learning can be used for automatic feature extraction. Application of the same algorithms and features may be used for other Philippine languages, however, retraining may be needed.

ACKNOWLEDGMENT

We would like to thank Adarna House Inc. for providing the much-needed storybook data which became the backbone of this study. We acknowledge the use of National University - Human Language Technology lab during the whole conduct of the study. And lastly, we are highly indebted to all the people who gave their assistance for this research.

TABLE VI. TOP FEATURE WORDS FOR EACH GRADE LEVEL

Level 1	Level 2	Level 3	Level 4
<i>isa</i>	0.003266	<i>nang</i>	0.003989
<i>lamang</i>	0.003335	<i>higit</i>	0.004027
<i>dati</i>	0.003391	<i>noon</i>	0.004041
<i>dahil</i>	0.003561	<i>ni</i>	0.004157
<i>noon</i>	0.003608	<i>wala</i>	0.004157
<i>pang</i>	0.003617	<i>nito</i>	0.004169
<i>lahat</i>	0.003743	<i>tatay</i>	0.00461
<i>mga</i>	0.003811	<i>mula</i>	0.004731
<i>ito</i>	0.003824	<i>siya</i>	0.004871
<i>kung</i>	0.004086	<i>maynila</i>	0.005024
		<i>itinatag</i>	0.003764
		<i>lahat</i>	0.004349
		<i>kay</i>	0.004356
		<i>lamang</i>	0.004376
		<i>ni</i>	0.004414
		<i>din</i>	0.004787
		<i>ilanlang</i>	0.004834
		<i>panahon</i>	0.004888
		<i>hanggang</i>	0.004931
		<i>na</i>	0.005016
		<i>rin</i>	0.004466
		<i>bayan</i>	0.004207
		<i>bansa</i>	0.004385
		<i>may</i>	0.004485
		<i>ay</i>	0.004715

REFERENCES

- [1] H.B. Macahilig. 2014. A Content-Based Readability Formula for Filipino Texts. The Normal Lights: Journal on Teacher Education, vol.8, no.1, ICEAP-17 (2014).
- [2] A.R. Razon, P.C. Naval,Jr., M.S. Bosque, R.R. Ledesma, and H.M. Loberas. 2015. A New Approach to Automated Text Readability Classification based on Concept Indexing with Integrated Part-of-Speech n-gram Features. Proceedings of Recent Advances in Natural Language Processing (September 2015), 521–528.
- [3] A. Al-Tamimi, M. Jaradat, N.Aljarrah, and S. Ghanim. 2014. AARI: Automated Arabic Readability Index. The International Arab Journal of Information Technology, vol. 11, no.4 (July 2014), 370-378.
- [4] P.van Oosten, D. Tanghe, and V. Hoste. 2010. Towards an Improved Methodology for Automated Readability Prediction. LREC 2010: Seventh Conference on International Language Resources and Evaluation (2010), 775-782.
- [5] D.S. McNamara, M.M. Louwerse, and A.C. Graesser. 2005. Coh-Metrix: Automated Cohesion and Coherence Scores to Predict Text Readability and Facilitate Comprehension. (2005).
- [6] D. Gemoets, G. Rosemblat, T. Tse, and R. Logan. 2004. Assessing Readability of Consumer Health Information: An Exploratory Study. MEDINFO 2004 (2004), 869 – 873.
- [7] L. Beinborn, T. Zesch, and I. Gurevych. 2012. Towards fine-grained Readability Measures for Self-Directed Language Learning. Proceedings of the SLTC 2012 Workshop on NLP for CALL 80 (2012), 11 – 19.
- [8] K. Tanaka-Ishii, S. Tezuka, and H. Terada. 2010. Sorting texts by Readability. Association for Computational Linguistics, vol. 36, no. 2 (2010), 203 – 228.
- [9] S. Tonelli, K.T. Manh, and E. Pianta. 2012. Making Readability Indices Readable. NAACL-HLT 2012, Workshop on Predicting and Improving Text Readability for Target Reader Populations (2012), 40 – 48.
- [10] M. M. Gutierrez. 2014. The Suitability of the Fry and SMOG Readability Formulae in Determining the Readability of Filipino Texts. The Normal Lights, vol. 8, no. 1 (2014).
- [11] J. Hancke, S. Vajjala, and D. Meurers. 2012. Readability Classification for German using Lexical, Syntactic, and Morphological Features. Proceedings of COLING 2012: Technical Papers (2012), 1063 – 1080.
- [12] E. Pitler and A. Nenkova. 2008. Revisiting Readability: A Unified Framework for Predicting Text Quality. Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (2008), 186 – 195.
- [13] A.C. Graesser, D.S. McNamara, M.M. Louwerse, and Z. Cai. 2004. Coh Metrix: Analysis of Text Cohesion and Language. Behavior Research Methods, Instruments, and Computers, vol. 36, no. 2 (2004), 193 – 202.
- [14] J.O. dela Rosa and C.F. Genuino. 2018. Correlating Writer's Block and ESL Learners' Writing Quality. Indonesian Journal of Applied Linguistics, vol. 7, no. 3 (2018), 604 – 612.
- [15] M. Xia, E. Kochmar, and T. Briscoe. 2016. Text Readability Assessment for Second Language Learners. Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications (2016), 12-22.
- [16] A.R. Razon, P.C. Naval,Jr., M.S. Bosque, R.R. Ledesma, and H.M. Loberas. 2011. A New Approach to Automated Text Readability Indexing using Concept Indexing with K-Means Algorithm. The 11th Philippine Computing Science Congress (March 2011).
- [17] L. Si and J. Callan. A Statistical Model for Scientific Readability. Proceedings of the 2001 ACM CIKM, 10th International Conference on Information and Knowledge Management. 2001.
- [18] M.J. Heilman, K. Collins, and J. Callan. Combining Lexical and Grammatical Features to Improve Readability Measures for First and Second Language Texts. Proceedings of NAACL HLT. 2007.
- [19] K.M. Sheehan, I. Kostin, Y. Futagi, and M. Flor. 2010. Generating Automated Text Complexity Classifications that are Aligned with Targeted Complexity Standards. ETS RR-10-28 (2010).
- [20] A.M. Villamin and E.S. de Guzman. Pilipino Readability Formula: The Derivation of a Readability Formula and a Pilipino Word List. Language Study Center: Philippine Normal University. 1979.
- [21] A. Rajaraman and J.D. Ullman. (2011). "Data Mining" (PDF). Mining of Massive Datasets. pp. 1–17. doi:10.1017/CBO9781139058452.002. ISBN 978-1-139-05845-2.
- [22] Sci-Kit Learn Org. 2007-2018. 1.9. Naive Bayes. Retrieved from https://scikit-learn.org/stable/modules/naive_bayes.html#naive-bayes.
- [23] Sunil Ray. 2017. 6 Easy Steps to Learn Naive Bayes Algorithm (with codes in Python and R). Retrieved from <https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/>.
- [24] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.
- [25] Oco, N., Ilao, J., Syliongka, L., & Roxas, R. (2013). Measuring language similarity using trigrams: Limitations of language identification. 2013 International Conference on Recent Trends in Information Technology (ICRTIT). doi:10.1109/ICRTIT.2013.6844
- [26] Antonina Durfee. (2008). Text Mining. In G. Garson, & M. Khosrow-Pour, D.B.A. (Eds.), Handbook of Research on Public Information Technology (pp. 592-603). Hershey, PA: IGI Global. doi:10.4018/978-1-59904-857-4.ch054

Exploring Context's Diversity to Improve Neural Language Model

Yanchun Zhang^{†‡}, Xingyuan Chen[‡]

[†]School of Computer and Software Engineering
Xihua University
Chengdu, China

Email(Yanchun Zhang): 834612949@qq.com

Peng Jin[‡], Yajun Du[†]

[‡]School of Computer Science
Leshan Normal University
Leshan, China

Email(Peng Jin): jandp@pku.edu.cn

Abstract—The neural language models (NLMs), such as long short term memory networks (LSTMs), have achieved great success over the years. However the NLMs usually only minimize a loss between the prediction results and the target words. In fact, the context has natural diversity, i.e. there are few words that could occur more than once in a certain length of word sequence. We report the natural diversity as context's diversity in this paper. The context's diversity, in our model, means there is a high probability that the target words predicted by any two contexts are different given a fixed input sequence. Namely the softmax results of any two contexts should be diverse. Based on this observation, we propose a new cross-entropy loss function which is used to calculate the cross-entropy loss of the softmax outputs for any two different given contexts. Adding the new cross-entropy loss, our approach could explicitly consider the context's diversity, therefore improving the model's sensitivity of prediction for every context. Based on two typical LSTM models, one is regularized by dropout while the other is not, the results of our experiment show its effectiveness on the benchmark dataset.

Keywords-Neural language model; Context's diversity; Loss function.

I. INTRODUCTION

As a fundamental task, language model (LM) which uses the previous words to predict the next word has been playing an important role in natural language processing (NLP). It is widely used in speech recognition[22], machine translation[21], text generation[5] and so on. The early methods estimate n-gram probabilities of a sequence of strings (words) resting on the count-based models with counting and smoothing[12]. But they are hard to get good performance because of data sparsity. In order to alleviate the data sparsity, the neural language models (NLMs)[24][25] are proposed. The neural language models do not use the one-hot-vectors but the low-dimension vectors to represent the input sequence of the neural network. These low-dimension vectors are initialized with random parameters.

Recently, recurrent neural network (RNN)[18], a neural sequence model, has achieved the state-of-the-art in language model[19][20]. RNN like LSTM[14] is good at alleviating the problem of gradient explosion and disappearance. It becomes one of the most popular models in NLP later. To the best of our knowledge, the NLMs mentioned above usually have the only loss function between the prediction results and the target words. However, we randomly select two words from a

context and will find that these two selected words are different from each other in most occasions, especially when they are close in a sequence. We show an example, a randomly selected sentence from the PTB, as follows:

sequence: *big investment banks refused to step up to the plate to support the beleaguered floor traders by buying big blocks of stock traders say heavy selling of standard & poor's 500-stock index futures in chicago unk beat stocks downward*

In this sequence, the number of the words that appear more than once is 5, and the length of the sequence is 39. The probability that the two selected words are the same is about 0.0094 which is calculated by equation (1). Equation (1) is a combinatory formula, where $E_i, i \in [1, 5]$, is the number of the i -th word that appears more than once in the sequence. L is the length of the sequence and P is the probability that the two selected words are the same.

$$P = \frac{\sum_{i=1}^5 \binom{2}{E_i}}{\binom{2}{L}} \quad (1)$$

It shows that if we randomly select two contexts in the text and feed them to train a NLM, the model's prediction results of these two contexts should be distinct. Because the target words corresponding to the two contexts are much likely to be diverse. In this paper, we introduce a new loss function that considers the diversity of the context as another constraint of the model.

Inspired by the discovery introduced above, we hope our model could improve its sensitivity of prediction for every context. In order to explicitly consider the effect brought by the diversity of different contexts, we create a new cross-entropy loss between the prediction results for any two different given contexts. The new cross-entropy loss could be added to the NLMs and is expected to be maximized. We implement our method on the baseline then called RNNR proposed by the paper[23] - recurrent neural network regularization. The RNNR will be introduced in section III(A). The code of our model could be accessed at <https://github.com/teanon/contextDiversity-LM>.

In summary, our contributions are as follows:

- We observe the diversity of the text and explorit it to

improve a neural network LM without increasing model's parameters.

- Experimental results on the benchmark dataset[9] show that our method exceeds the baseline model. Our model has a decrease in the number of iteration, which will reduce the training time of a model especially that has a large size.

II. RELATED WORK

As a fundamental task, language model is widely used in other tasks of NLP such as words prediction[13], text generation[1] and so on. So how to improve the performance of language model is particularly important. To our knowledge, many works are dedicated to modifying the network structure of the LM, such as RNN[18], the CRNNs[3][6] that combine CNN with RNN and self-attention mechanism[17]. But the models mentioned above just use the traditional softmax loss as the model's final loss and don't try to improve their performance by modifying the loss of model.

However improving language model's performance by modifying the loss function also is an effective way[15]. [7] replaces the softmax loss with margin loss to address the issue that softmax loss only focuses on whether the sample is correctly classified, and does not require intra-class compactness and inter-class separation. What's more, [11] adds a new loss term which attempts to equalize the probabilities of male and female words in the output to the traditional loss function to alleviate the gender bias in text generation.

In this paper, we take language model as our research task which is one kind of tasks of the RNNR. We attempt to enhance the model's sensitivity of prediction for every context via the way of adding a new cross-entropy loss function to the RNNR. Our experiments will demonstrate that our model is simple but effective.

III. THE MODEL WITH THE DIVERSITY OF CONTEXT

A. The Recurrent Neural Network Regularization

The RNNR is dedicated to modifying dropout operator which has been working very well in feed-forward neural networks [8][16]. Conducting dropout only on the non-recurrent connections could easily learn to conserve effective information for a long period of time. It also uses the traditional cross-entropy loss between Z and Y as the final loss for the model. As shown in equation (2), $Y_i, i \in [1, len]$ is the i -th element of Y , and Y is an one-hot-vector sequence of the target words. $Z_i, i \in [1, len]$, is the softmax result corresponding to the context of the i -th step in LSTM, where len is the length of the input sequence and also is the size of Z . Z is a vector sequence consisting of the softmax outputs of an entire input sequence, and it could be calculated by equation(3). The structure of the RNNR is displayed in Figure.1, and its model is explained by the equations as follows:

$$Loss_{ZtoY} = \frac{1}{len} \sum_{i=1}^{len} crossEntropy(Y_i, Z_i) \quad (2)$$

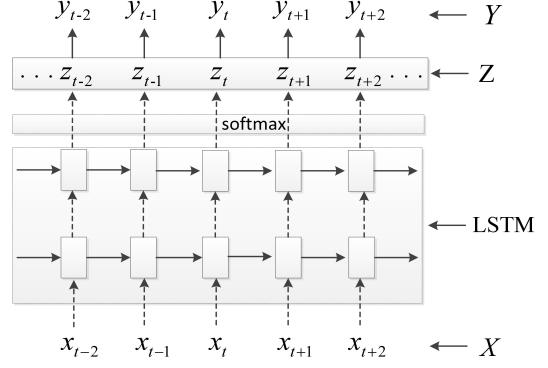


Figure 1. The dashed arrow indicates the connection to which the dropout is applied, while the solid indicates the connection without dropout in the regularized multilayer LSTM where Z is the output sequence of softmax results.

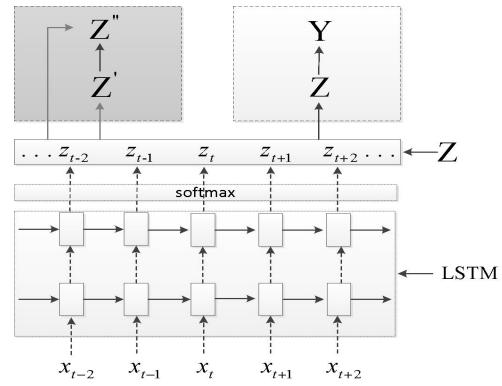


Figure 2. The architecture of our model. The dashed arrow indicates the connection to which the dropout is applied, while the solid indicates the connection without dropout. Z is the sequence of softmax results. Both Z' and Z'' are sampled from Z and our model calculates the new cross-entropy loss between these two subsets.

$$Z = softmax(LSTM(X)) \quad (3)$$

B. The Model with Context's Diversity

In our model, we add a new cross-entropy loss between Z' and Z'' to the RNNR. Both Z' and Z'' are sampled from Z according to some certain sampling rules and the rules will be introduced in detail in section III(C). Equation (4) is the new loss function based on the context's diversity. n is the size of Z' and Z'' . In our model, Z'_j and Z''_j , $j \in [1, n]$, need corresponding to the softmax results of different step in LSTM. Our model is illustrated in Fig.2.

$$Loss_{Z' to Z''} = \frac{1}{n} \sum_{j=1}^n crossEntropy(Z'_j, Z''_j) \quad (4)$$

The new cross-entropy loss between Z' and Z'' is added to the RNNR to form the final loss function shown in equation (5). The final loss function of our model is shown as follows:

$$FinalLoss = Loss_{ZtoY} + \alpha * \frac{1}{\exp Loss_{Z' to Z''} + \gamma} \quad (5)$$

In equation (5), α and γ are hyper-parameters. α is the weight coefficient of $Loss_{Z' to Z''}$ which represents the

diversity of the softmax results corresponding to n pairs of different contexts. γ is used to avoid the exception divided by zero. In our model, we want the softmax outputs of the two different contexts could vary greatly. Therefore $Loss_{Z' \text{to} Z''}$ needs to be maximized. Considering that the $FinalLoss$ is expected to be as small as possible, we apply the reciprocal of $Loss_{Z' \text{to} Z''}$ dealt with exponential function to be a part of the $FinalLoss$.

C. The Sampling Rules

Interval Sampling(ISP): In order to explore how the distance between Z'_j and $Z''_j, j \in [1, n]$, could affect the performance of the model. We let $Z' = [Z_1, Z_2, \dots, Z_{len-k}]$ and then $Z'' = [Z_{1+k}, Z_{2+k}, \dots, Z_{len}]$. Namely the step interval between Z'_j and $Z''_j, j \in [1, n]$, is k in the LSTM, where k is a hyper-parameter. len is the length of the input sequence. $n = len - k$, where n is the number of elements in Z' and Z'' . Because the range of k is small, the change of n can be ignored.

Random Sampling(RSP): There is a conclusion, which will be explained in section V(A), that the performance of the model is not sensitive to k . For this part, we want to explore how the number of samples in Z' and Z'' could affect the performance of the model. We sample Z' and Z'' from Z randomly, and n is the number of samples in Z' and Z'' .

IV. EXPERIMENT

In order to verify the effectiveness of our model, we apply word-level prediction experiments on the PTB (Penn Tree Bank) dataset. We design two rules to sample Z' and Z'' from Z , where Z', Z'' and Z are introduced in section III.

We experiment on the large LSTM with dropout regularization and the small LSTM without regularization respectively. For the two sampling rules, the length of the input sequence and the batchsize are 35, 20 respectively for the unrolled large LSTM, and are 20, 20 respectively for the unrolled small LSTM. All hyper-parameters of our model are used to minimize the perplexity of the validation set. The results of the RNNR are directly cited from their paper.

A. Dataset

As the RNNR did, we also use the PTB (Penn Tree Bank) dataset to measure our model. It consists of three parts, the training set, the validation set and the test set. They are composed of 929k, 73k and 82k tokens respectively. The size of vocabulary is 10k. The dataset has been pre-processed and it contains all 10k different words. All words are lowercase and all sparse words are replaced by unk. The PTB dataset is small and easy to train.

Table I
WORD-LEVEL PERPLEXITY ON THE PTB DATASET, THE LOWER IS THE BETTER.

Model	Validation Set	Test Set
large LSTM (RNNR)	82.2	78.4
large LSTM (ISP)	81.2 \pm 0.056	77.5 \pm 0.019
large LSTM (RSP)	81.3 \pm 0.001	77.4 \pm 0.024
small LSTM (RNNR)	120.7	114.5
small LSTM (ISP)	113.8 \pm 0.134	109.6 \pm 0.049
small LSTM (RSP)	113.5 \pm 0.210	109.4 \pm 0.477

B. The Experiment According to ISP

1) *The Large LSTM :* We run our model with setting the learning rate to 1 during the first 18 epochs and then we apply a factor as 0.78 to decrease the learning rate for the rest of epochs. We totally train our model with 47 epochs. For each group of parameters, we train our model five times and take the average of results as the final result. When k is set to 1 and the value of α is 0.1, the large LSTM according to ISP gets the lowest perplexity on the validation set and its perplexity on the test set drops by 0.9 compared with the RNNR, as it is shown in Table I.

2) *The Small LSTM :* For the first four epochs, we train our model with setting the learning rate to 1, then we use a factor that is 0.35 to decrease the leaning rate for the following epochs. The number of the total epochs is 11. We train our model five times to achieve the average, serving as the final result, of the model's perplexity for each group of parameters. When the k is assigned with 5, and the α is 1, the perplexity values, displayed in Table I, that are 13.9 for the validation set and 109.7 for the test set respectively drop obviously in contrast with the RNNR.

C. The Experiment According to RSP

1) *The Large LSTM :* For the large LSTM, we also train our model five times to take the mean value of the model's perplexity as the final result for each group of parameters. When the hyper-parameters, except the α and n which are 5 and 20 respectively, are the same as the large LSTM introduced in section IV(B.1), the large LSTM according to RSP gains the lowest test perplexity as shown in Table I.

2) *The Small LSTM :* For the small LSTM, we also train our model five times to take the mean value of the model's perplexity as the final result for each group of parameters. When the small LSTM according to RSP gains the lowest perplexity values on both the validation set and the test set, as it is presented in Table I, the α is assigned with 1 and n also is set to 20 and other hyper-parameters are set to being in accordance with the small LSTM of section IV(B.2).

V. DISCUSSION

In this part, we will mainly discuss the hyper-parameters' effect on model's performance. The hyper-parameters we focus on are k , α and n , where k introduced in ISP is the interval between Z'_j and $Z''_j, j \in [1, n]$. n is the number of samples in Z' and Z'' according to RSP. α is the weight coefficient of $Loss_{Z' \text{to} Z''}$.

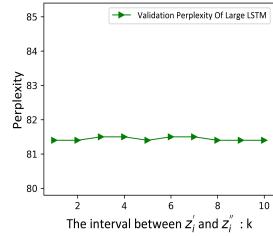
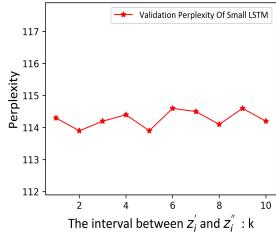


Figure 3. The k is the interval between Z'_j and Z''_j according to ISP. We vary the k from 1 to 10. With the k increasing gradually, the trends of validation perplexity for the small LSTM and the large LSTM are shown in the left picture and the right picture respectively.

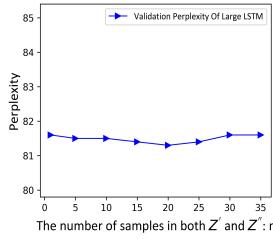
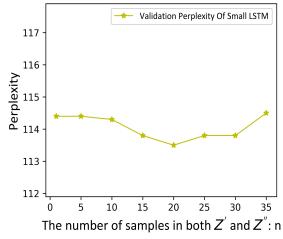


Figure 4. The x-axis represents the number of samples in both Z' and Z'' . The trend of validation perplexity for the small LSTM is shown in the left picture and for the large LSTM is taken on the right picture, with the increase of n .

A. The Influence of k According to ISP

At the begining, k is set to 1, and increase 1 each time until it is 10. The α of the small LSTM and the large LSTM is empirically fixed to 1 and 0.01 repectively. From the results in Fig.3, we can observe that both the large LSTM and the small LSTM have a inconspicuous fluctuating validation perplexity. The values of the variance for the small LSTM and the large LSTM are 0.06 and 0.02 respectively. So we could draw a conclusion that the performance of the model is not sensitive to k . Because the diversity of the two predicted words is not determined by their distance in the input sequence, but by the meaning of themselves and the contexts they respectively correspond to. In the same way, the diversity of two contexts also does not depend on the distance of them in the input sequence. The k used to get the lowest validation perplexity for the small LSTM is 5, and for the Large LSTM is 1.

B. The Influence of n According to RSP

For this section, we observe the variety of performance with respect to the number of samples, i.e. n according to RSP. The α of the small LSTM and the large LSTM are empirically fixed to 1.0 and 0.1 repectively. We respectively set n to 1, 5, 10, 15, 20, 25, 30 and 35 for both the small LSTM and the large LSTM. As it is shown in Fig.4, the results of validation perplexity for these two LSTM models show the same tendency. It first decreases until n is 20 and then increases with n increasing gradually. Both the small LSTM and the large LSTM get the lowest validation perplexity when n is 20.

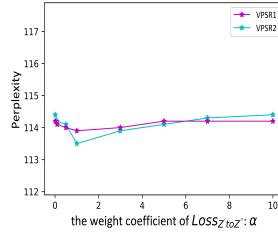
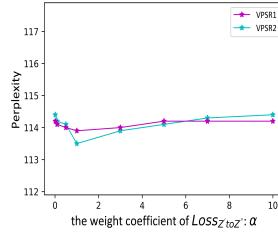


Figure 5. As it is shown in the left picture, the VPSR1 is the validation perplexity of small LSTM according to ISP, and the VPSR2 is the validation perplexity of small LSTM according to RSP. In the right picture, the VPLR1 is the validation perplexity of large LSTM according to ISP, and the VPLR2 is the validation perplexity of large LSTM according to RSP. The x-axis is the value of α which is the weight coefficient of $Loss_{Z' \text{ to } Z''}$

C. The Influence of α in both ISP and RSP

The α is set to 0.01, 0.1, 0.5, 1, 3, 5, 7 and 10 repectively and the results are shown in Fig.5.

For the small LSTM whose results are shown in the left picture of Fig.5, the results of validation perplexity according to both ISP and RSP show the same tendency to decrease first and then increase with α increasing gradually in general. As it is shown in VPSR1 and VPSR2 respectively. The small LSTMs for both ISP and RSP get the lowest validation perplexity with setting the α to 1. In line with the results in section V(A) and V(B), we use the best hyper-parameters setting, i.e. k is 5 and n is 20.

For the large LSTM whose results are shown in the right picture of Fig.5, the results of validation perplexity according to both ISP and RSP change obviously when α is less than 3. But the trends for both ISP and RSP become stable gradually when α is greater than 3. As the VPLR1 shows, the large LSTM resting on ISP gets the best result with setting α to 0.1. The large LSTM based on RSP gets its best result when α is 5 as the VPLR2 shows. According to the results in section V(A) and V(B), we use the best hyper-parameters setting, i.e. k is 1 and n is 20.

VI. CONCLUSION

Different from previous LSTM models which only pay attention to making the prediction results to be the same as the target words as possible, our model additionally expects the prediction results of two different contexts could be different as much as possible. It is combining the $Loss_{Z' \text{ to } Z''}$ with $Loss_{Z \text{ to } Y}$ that our model could take the diversity of the context into consideration when the model is pursuing the consistency between the model's outputs and the labels. Considering the diversity of the context as constraint could enhance model's sensitivity of prediction. Compared with the RNNR, our model drops the test perplexity for the large LSTM by 1 and 5.1 for the small LSTM. The small LSTM and the large LSTM could get the best results after being respectively trained for 11 and 47 epochs in our model, but 13 and 55 epochs in the RNNR. So our model could reduce the training time of the model to some extent.

For the future work, we are going to explore whether the context's diversity could improve the performance

of attention-based NLM[4]. Further more, we will embed it into some applications such as neural machine translation[21] to further manifest the effectiveness of our method.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (61373056, 61472329, 61532006, 61602389, 6187229861874401, 61902324). Xingyuan Chen and Peng Jin are the co-crossponding authors for this paper.

REFERENCES

- [1] B. Dzmitry, C. Kyunghyun and B. Yoshua, *Neural machine translation by jointly learning to align and translate.* : arXiv:1409.0473, 2014.
- [2] H. B.Mcmahan, D. Ramage, K. Talwar and L. Zhang, *Learning Differentially Private Recurrent Language Models.* : international conference on learning representations, 2018.
- [3] J. Chiu and E. Nichols, *Named Entity Recognition with Bidirectional LSTM-CNNs.* Transactions of the Association for Computational Linguistics, **4**, 357—370, 2016.
- [4] J. Devlin, M. Chang, K. Lee and K. Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* CoRR, abs/1810.04805, 2018.
- [5] J. Guo, S. Lu, H. Cai, W. Zhang, J. Wang and Y. Yu, *Long Text Generation via Adversarial Training with Leaked Information.* : national conference on artificial intelligence, 5141—5148, 2018.
- [6] J. Wang, L. Yu, K. Lai and X. Zhang, *Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model.* Berlin, Germany: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics 2016, pp. 225—230, 2016.
- [7] L. Ting-En, X. Hua, *Deep Unknown Intent Detection with Margin Lossn.* : Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 5491—5496, 2019.
- [8] L. Wan, M. Zeiler, S. Zhang, Y. LeCun and R. Fergus, *Regularization of Neural Networks using DropConnect.* : Proceedings of the 30th International Conference on Machine Learning (ICML-13), pp. 118—126, 2013.
- [9] M. Marcinkiewicz, M. Marcus and B. Santorini, *Building a large annotated corpus of english: The penn tree bank.* Computational linguistics, **19**(2), 313—330, 1993.
- [10] Q. Ai, L. Yang, J. Guo and W. Croft, *Improving Language Estimation with the Paragraph Vector Model for Ad-hoc Retrieval.* : international acm sigir conference on research and development in information retrieval, 2016.
- [11] Q. Yusu, M. Urwa, Z. Ben and H.W. Jae, *Reducing Gender Bias in Word-Level Language Models with a Gender-Equalizing Loss Function.* : Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, 223—228, 2019.
- [12] S. Chen, and J. Goodman, *An Empirical Study of Smoothing Techniques for Language Modeling.* : Technical Report, Harvard University, 1998.
- [13] S. Ghosh, O. Vinyals, B. Strope, S. Roy, T. Dean and L. Heck, *Contextual LSTM (CLSTM) models for Large scale NLP tasks.* CoRR, abs/1602.06291, 2016.
- [14] S. Hochreiter and J. Schmidhuber, *Longshort-term memory, Neural computation.* **9**(8), 1735—1780, 1997.
- [15] S. Lei, X. Hu and L. Bing, *Deep open classification of text documents.* : In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2911—2916, 2017.
- [16] S. Wang, and C. Manning, *Fast dropout training.* : Proceedings of the 30th International Conference on Machine Learning (ICML-13), pp. 118—126, 2013.
- [17] T. Ke, B. Arianna and M. Christof, *Recurrent memory networks for language modeling.* : arXiv preprint arXiv:1601.01272, 2016.
- [18] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky and S. Khudanpur, *Recurrent Neural Network Based Language Model.* : Proceedings of INTERSPEECH, 2010.
- [19] T. Mikolov, S. Kombrink, A. Deoras, L. Burget and J. Cernocky, *RNNLM-recurrent neural network language modeling toolkit.* : Proceedings of the 2011 ASRU Workshop, pp. 196—201, 2011.
- [20] T. Mikolov and G. Zweig, *Context dependent recurrent neural network language model.* : SLT Workshop, 2012.
- [21] U. Hermjakob, Q. Li, D. Marcu, J. May, S. Mielke, N. Pourdamghani, and H. Ji, *Incident-Driven Machine Translation and Name Tagging for Low-resource Languages.* Machine Translation, 59—89, 2018.
- [22] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke and G. Zweig, *The microsoft 2016 conversational speech recognition system.* : international conference on acoustics, speech, and signal processing, pp. 5255—5259, 2017.
- [23] W. Zaremba, I. Sutskever and O. Vinyals, *Recurrent neural network regularization.* : arXiv preprint arXiv:1409.2329, 2014.
- [24] Y. Bengio, R. Ducharme, P. Vincent and C. Janvin, *A neural probabilistic language model.* Journal of Machine Learning Research, **3**(6), 1137—1155, 2003.
- [25] Y. Bengio, R. Ducharme and P. Vincent, *A Neural Probabilistic Language Model.* Journal of Machine Learning Research**3**, 1137—1155, 2003.

Phrase-Based Tibetan-Chinese Statistical Machine Translation

YONG Cuo^{1,2}, Xiaodong SHI^{1*}, NYIMA Tashi^{2*}, Yidong CHEN¹

School of Information Science and Technology, Xiamen University, Xiamen, P.R.China¹

School of Information Science and Technology, Tibet University, Lhasa, P.R.China²

yongtso@163.com; mandel@xmu.edu.cn; nmzx@utibet.edu.cn; ydchen@xmu.edu.cn

Abstract—Statistical machine translation has made great progress in recent years, and Tibetan-Chinese machine translation has many needs. A phrase-based translation model is suitable for machine translation between Tibetan and Chinese, which have similar morphological changes. This paper studies the key technologies of phrase-based Tibetan-Chinese statistical machine translation, including phrase-translation models and reordering models, and proposes a phrase-based Tibetan-Chinese statistical machine translation prototype system. The method proposed in this paper has better accuracy than Moses, the current mainstream model, in the CWMT 2013 development set, and shows great performance improvement.

Keywords—Machine translation; Statistics; Phrase; Tibetan-Chinese

I. INTRODUCTION

Machine translation studies the use of computers to automatically convert between natural languages. It is one of the important research directions in the field of artificial intelligence and natural language processing [Koehn et al. 2003]. Machine translation is a key technology that breaks through the language barrier when transmitting information between countries. It is particularly important for strengthening cultural exchange and promoting foreign trade under the national strategy of the One Belt and One Road Initiative.

People have gradually realized that artificial translation rules have become a bottleneck in machine-translation research. Since the 1980s, the rise and rapid development of corpus linguistics have been refreshing. At the MT Summit IV conference held in Japan in 1993, the British scholar John Hutchins pointed out in a guest lecture that machine-translation research had entered its third generation. A major indicator is the introduction of corpus methods, including statistical and instance-based methods. Different from other natural language-processing research, the corpus used for machine-translation research is generally a parallel corpus, i.e., it contains translations between multiple languages. A common one is a bilingual parallel corpus consisting of two languages. Because the bilingual corpus contains contrasting translation information between two different languages, it has high research value and practical value in the field of natural language processing. Statistical machine translation has made great progress in recent years, with many significant achievements in international evaluation.

In China, the study of machine translation began in the 1950s. Several institutes and universities developed machine-translation systems such as Russian-Chinese, English-Chinese, Chinese-English, Japanese-Chinese, and

Chinese-Japanese, and performed much research on the development of natural-language understanding of Chinese.

Tibetology is studied worldwide. Therefore, research on machine translation between Tibetan and other languages is of high concern. With the implementation of the One Belt and One Road Initiative and the advancement of informationization in Tibetan areas in China, there is wide demand for Tibetan-Chinese machine-translation technology in journalism, education, academic research, publishing, information security management, and cultural exchange. Hence, it is particularly important to study Tibetan-Chinese machine translation, which will not only enrich machine-translation theory, but also promote the development of Tibetan information technology, laying a solid foundation for the ultimate development of a Tibetan-Chinese two-way machine-translation system.

II. RELATED WORK

The study of machine translation in Tibetan and other languages started late, in the 1990s. Based on the research and development practice of the “863” project “BanZhida Chinese-Tibetan Official Document Machine Translation System”, Cai [2009] and Cai and Hua [2005] discussed the principle of combining lexical information with grammatical rules, and proposed a dichotomy of syntactic analysis centering on verbs. Combining their study with Tibetan language corpora segmentation specifications, they also discussed the establishment and design of a Tibetan grammar information dictionary for the segmentation and labeling of Tibetan corpora, and focused on the content construction, annotation of grammar information, index structure, and search algorithms of the dictionary. Chen and Yu [2003] introduced and evaluated related research work on the development of Tibetan-language information processing in China—the Tibetan language operating system, Tibetan information technology standards, Tibetan information processing, and comprehensive applications. They specifically mentioned their two “863” projects—a Chinese-Tibet science and technology machine translation system and a practical Chinese-Tibetan machine translation system, which adopted a conversion-translation model and a rule-based machine translation method. Based on the actual needs of constructing Tibetan corpora, Cai and Ji [2005] proposed a corpus-based Tibetan word-class annotation and classification method. Suo et al. [2004] proposed a British-Tibetan translation system based on rules and corpora, and outlined the design ideas and principles of the system. Cao and Suo [2009] proposed a language model and structural design of a British-Tibetan machine-translation system based on rules and corpora. He et al. [2015] analyzed Tibetan case-auxiliary words, added semantic information of the

Tibetan ontology based on the Tibetan phrase tree library, and proposed a Tibetan-Chinese machine-translation method that integrates Tibetan semantic information. Wang [2016] proceeded from the linguistic characteristics of the Tibetan language and carried out a study of identifying functional blocks of Tibetan sentences for an practical Tibetan-Chinese machine translation system.

In summary, Tibetan-Chinese machine-translation technology is mainly based on rules and linguistic materials, and does not involve phrase-based statistical machine translation. In particular, the Tibetan-Chinese statistical machine-translation research foundation is weak in the research on reordering models. Therefore, the key technology research and achievements of Tibetan-Chinese machine translation have strong academic research value and a wide range of application prospects, for both the actual needs of the society and the development of Tibetan natural-language processing research.

III. PHRASE-BASED TIBETAN-CHINESE STATISTICAL MACHINE

A relatively mature statistical translation method is a phrase-based translation method, which improves on IBM's word-based translation model. Word- and phrase-based machine translation use words and phrases, respectively, as the basic unit of translation. Taking the phrase as a basic unit does not need to consider the grammatical information between the words in a phrase, and therefore the obtained translation result is more accurate and reasonable than word-based translation. A phrase-based translation model is suitable for machine translation between Tibetan and Chinese, which share similar morphological changes.

A. Phrase Translation Model

A phrase translation model reflects the probability of translating a source-language phrase to a target-language phrase. Through translation model training, namely, word alignment and phrase extraction, Tibetan-Chinese phrase-translation probabilities are obtained from Tibetan-Chinese parallel training corpora.

1) Word Alignment

Word-alignment training obtains word-alignment models from sentence-aligned corpora. Using GIZA++ to perform bidirectional training between Tibetan and Chinese after pre-processing Tibetan-Chinese training corpora, bidirectional alignment results are obtained.

2) Phrase Extraction

Phrases are extracted from Tibetan-Chinese word-alignment results, translation probabilities are calculated, and the Moses statistical machine-translation system is used to obtain a phrase translation model. A Perl script optimizes bidirectional alignment results to obtain an alignment matrix. Phrases are extracted from the alignment matrix to obtain a phrase table, the probability of phrase-translation for each phrase pair is calculated using maximum likelihood estimation, and an IBM Model 1 is used to obtain the lexical translation probability and a Tibetan-Chinese phrase-translation probability table.

3) Decoding

Using the currently most popular and stable Moses decoder, a beam-search algorithm with dynamic programming is used to implement the search function. The decoder reads the configuration file and test file and outputs a file of translation results. The translation outcome is improved by adjusting the feature weights of the model.

4) Translation Rules

To guarantee the quality of translation, pruning and clustering methods are used to reduce the number of translation rules. The goal is to reduce the number of translation rules to about 1/4 so that the translation speed can be increased to more than 100 words per second, with only a slight decrease in quality. Specific strategies include:

- To introduce part-of-speech rules for abstraction, rather than the current phrase-system translation rules that has only a specific vocabulary. As long as the parts of speech are the same, translation rules can be matched.
- To remove translation rules with too small a probabilities.
- To remove reordering rules with too small probabilities.

B. Language Model

This paper uses the traditional n-gram language model with SRILM [Stolcke 2002] to train the five-gram language model.

C. Reordering Model

The reordering model adopts a new reordering algorithm, which can better solve the syntactic differences between Tibetan and Chinese due to the difference in verbal order.

All available information for the two sets of phrases (phrases to be ordered and phrases for reference) needed for reordering, including the vocabulary itself, parts of speech, and even other lexical semantic information, is assembled. The two sets of phrases may be adjacent or non-adjacent. The reordering operation is expanded to include M, S, L, R, F, and I, which respectively represent monotonous, switch, left discontinuity, right discontinuity, end of sentence, and beginning of sentence. It is represented by the following equation (1):

$$p(o|e, f) = \prod_{i=1}^k p(o|\overline{e_{i-n}}, \overline{f_{i-n}}), \overline{e_i}, \overline{f_i})$$

The difference between our reordering model and another model is shown in Figure 1.

Due to the sparseness of the data, we obtain more reliable reordering calculations through equivalence-like functions:

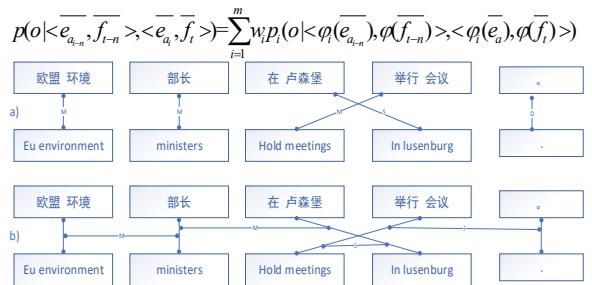


Figure 1. (a) Galley and Manning 2008 Model, (b) Our model.

The algorithm combines the reordering model with the linear decoding process from left to right. Our model differs from the previous model, as shown in Figure 2.

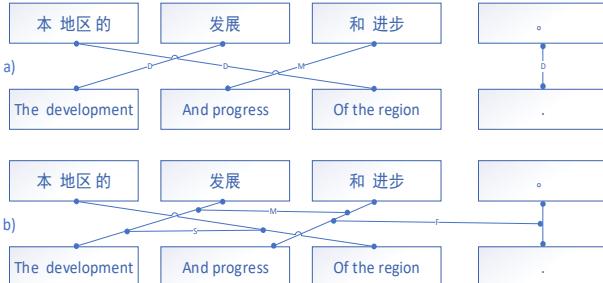


Figure 2. (a) Galley and Manning 2008 Model, (b) Our model.

Without considering the end of the sentence and sentence order, there are six reordering methods for adjacent phrases, as shown in Figure 3.

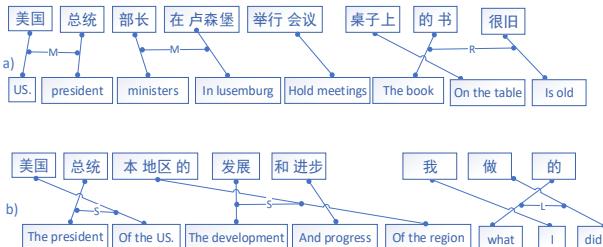


Figure 3. Six main ordering ways.

The reordering algorithm is as follows.

Algorithm 1 Computing Reordering Probabilities

```

find bilingual phrases using grow-diag-final-and method
foreach adjacent phrase pair  $\langle s_{i-1}, t_{i-1}, s_i, t_i \rangle$  from left to right
  If  $t_{i-1}, t_i$  are adjacent,
  then If  $t_{i-1}$  is to the left of  $t_i$ ,
    then  $o \leftarrow m$ 
    else  $o \leftarrow s$ 
  else     If  $t_{i-1}$  is to the left of  $t_i$ ,
        then if there are phrases between  $t_{i-1}$  and  $t_i$  that align
        to the left of  $s_{i-1}$ ,
        then  $o \leftarrow r$ 
        else  $o \leftarrow m$ 
        else if there are phrases between  $t_{i-1}$  and  $t_i$ 
        that align to the left of  $s_{i-1}$ ,
        then  $o \leftarrow l$ 
        else  $o \leftarrow s$ 
  add  $\langle s_{i-1}, t_{i-1}, s_i, t_i, o, 1 \rangle$  to the reordering frequency table
  foreach equivalence function  $\emptyset_i$ 
  add  $\langle \emptyset_i(s_{i-1}), \emptyset_i(t_{i-1}), \emptyset_i(s_i), \emptyset_i(t_i), o, 1 \rangle$  to the reordering freq
  table
Normalize the raw frequencies into probabilities
  
```

Experiments showed that our reordering algorithm had better accuracy than the current mainstream model (Moses). In the CWMT 2013 development set, there was a great

performance improvement. Performance comparison is shown in Table 1.

TABLE1. PERFORMANCE COMPARISON

	BLEU SBP	BLEU	LM
Moses phrase model	0.5562	0.5847	1016 M words, 5-gram
Our model	0.5819	0.6109	10 M words, trigram

D. System Architecture and Implementation

The Tibetan-Chinese machine translation system employs a hierarchical phrase model. A new semantic model is used to extract the hierarchical rules to solve the generalization problem of the model. Based on Tibetan sentence analysis, the syntax (boundary) restriction is used to further reduce the redundancy of rules.

The phrase-based Tibetan-Chinese statistical machine translation prototype system contains five major modules. Its overall architecture and translation engine are illustrated in Figures 4 and 5. In terms of word segmentation, the Chinese word-segmentation tool segtag developed by the Natural Language Processing Laboratory of Xiamen University is used for Chinese, and the Sunshine Tibetan-language word-segmentation tool developed by our laboratory is used for Tibetan.

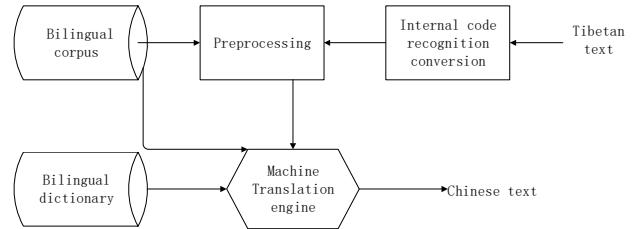


Figure 4. The system architecture

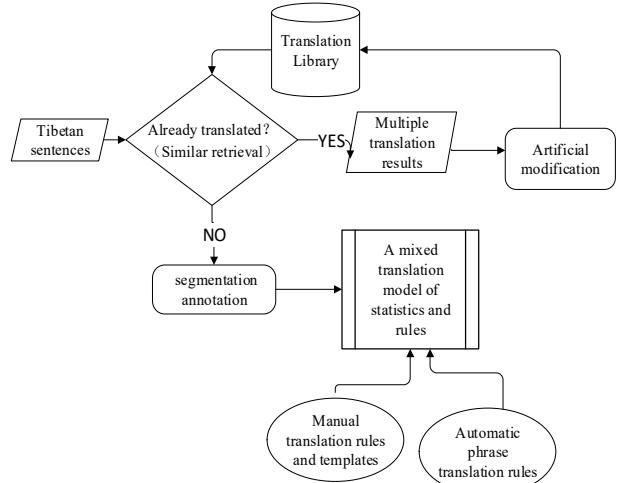


Figure 5. Translation engine schematic

Given a source-language sentence f , the statistical machine translation process based on a logarithmic display model searches for the target translation with the highest probability, e :

$$e^* = \arg \max \sum_{m=1}^M \lambda_m h_m(e, f)$$

where $h_m(e, f)$ is the characteristic function and λ_m is the feature weight. In the phrase-based model, the characteristic functions used are:

- 1) Phrase translation probability (two directions);
- 2) Word translation probability (two directions);
- 3) Language model;
- 4) Reordering based on source phrases;
- 5) Length penalty.

Thus, there are seven features. The decoding search strategy is a column-search algorithm, which ultimately generates a 1-Best result. At present, the weight of each characteristic function is set by experience.

Training is performed using the GIZA++ toolkit, and grow-diag-final heuristics are used for word-alignment expansion to generate phrases. When constructing the phrase table, the word translation probability of GIZA++ is fused to avoid the situation where the translation of certain words is not in the phrase table.

The phrase-based Tibetan-Chinese statistical machine translation system currently has about 75% readability of translation results in news and government documents.

IV. CONCLUSION AND FUTURE WORK

Statistical machine translation has made great progress in recent years, and Tibetan-Chinese machine translation technology has many needs. Since a phrase-based translation model is suitable for machine translation between Tibetan and Chinese, which have similar morphological changes, this paper has studied the key technologies of phrase-based Tibetan-Chinese statistical machine translation, including phrase-translation and reordering models, and we have designed and implemented a phrase-based Tibetan-Chinese statistical machine translation prototype system.

At present, the performance of the system in translating news and government documents is acceptable. However, due to the limited scale of Tibetan-Chinese parallel corpora, the quality of translation in other areas must be improved. The system uses a relatively ancient hierarchical phrase model, with much room for improvement.

Research on neural networks has progressed greatly in recent years. In the future, a Tibetan-Chinese statistical machine translation method integrating hierarchical phrase models and neural network translation models [Bahdanau et al. 2014; Cho et al. 2014; Devlin et al. 2014; Kalchbrenner and Blunsom 2013; Wang et al. 2017] should be studied to further improve the quality of Tibetan-Chinese machine translation.

V. ACKNOWLEDGMENTS

This work was supported by the National Key Research and Development Program of China (No. 2017YFB1402200) and the National Team and Key Laboratory Construction Program of Computer and Tibetan Information Technology of the Education Department of Tibet Autonomous Region.

Xiaodong SHI and NYIMA Tashi are the corresponding authors.

VI. ADDITIONAL AUTHORS

Additional authors: GAMA Trashi (Tibet University, email: 80957997@qq.com), Yang GUO (Tibet University, email: sdfg33445@qq.com).

REFERENCES

- [1] BAHDANAU, D., CHO, K. AND BENGIO, Y. 2014 Neural machine translation by jointly learning to align and translate, *Comput. Res. Repos.* <https://arxiv.org/abs/1409.0473>.
- [2] CAI, R.J. AND JI, T.J. 2005 Researches of speech classification methods based on tibetan repertoire, *Journal of Northwest University for Nationalities (Natural Science)* 26, 2, 39-42.
- [3] CAI, Z.T. 2009 Design of Tibetan segmentation dictionary and its algorithm study, *Journal of Computer Applications* 7, 2019-2021.
- [4] CAI, Z.T. AND HUA, G.J. 2005 Research of banzhida Chinese-Tibetan document translation system based on the dichotomy of syntax analysis, *J. Chin. Inf. Proc.* 19, 6, 9-14.
- [5] CAO, Y.L. AND SUO, N.D.Z. 2009 English-Tibetan machine translation system model and population structure design, *Journal of Southwest University for Nationalities (Natrual Science Edition)* 2, 365-370.
- [6] CHEN, Y.Z. AND YU, S.W. 2003 Research status and prospect of Tibetan information processing technology, *China Tibetology* 4, 97-107.
- [7] CHO, K., van MERRIENBOER, B.M., GULCEHRE, C., BAHDANAU, D., BOUGARES, F., SCHWENK, H. AND BENGIO, Y. 2014 Learning phrase representations using encoder-decoder for statistical machine translation, *Comput. Res. Repos.* <https://arxiv.org/abs/1406.1078?context=cs.NE>.
- [8] DEVLIN, J., ZBIB, R. AND HUANG, Z. 2014 Fast and robust neural network joint models for statistical machine translation, In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics Association for Computational Linguistics, Baltimore, USA, 1370-1380.
- [9] HE, X., WAN, F., YU, H. AND WU, X. 2015 Machine translation technology based on Tibetan semantic parsing, *Comput. Eng. Appl.* 15, 134-137,173.
- [10] KALCHBRENNER, N. AND BLUNSM, P. 2013 Recurrent continuous translation models, In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing Association for Computational Linguistics, Seattle, USA, 18-21.
- [11] KOEHN, P., OCH, F.J. AND MARCU, D. 2003 Statistical phrase-based translation, In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology Edmonton, Edmonton, Canada, 48-54. doi:10.3115/1073445.1073462
- [12] STOLCKE, A. 2002 SRILM-an extensible language modeling toolkit, In 7th International Conference on Spoken Language Processing ISCA, Denver, USA.
- [13] SUO, N.D.Z., MA, N.C. AND CAO, Y.L. 2004 Regulation and corpus-based English to Tibetan machine translation system design model, *Terminology standardization and Information Technology*, 4, 37-42.
- [14] WANG, M., LU, Z., ZHOU, J. AND LIU, Q. 2017 Deep neural machine translation with linear associative unit, *Comput. Res. Repos.* <https://arxiv.org/abs/1705.00861>.
- [15] WANG, T.H. 2016 An MT-oriented research on recognition of Tibetan syntactic functional chunk, Master's thesis. Computer Science and Technology Beijing Institute of Technology, Beijing, China.

A General Procedure for Improving Language Models in Low-Resource Speech Recognition

Qian Liu*, Wei-Qiang Zhang*, Jia Liu*, Yao Liu†

* Beijing National Research Center for Information Science and Technology

Department of Electronic Engineering, Tsinghua University

Beijing 100084, China

Email: q-liu18@mails.tsinghua.edu.cn, {wqzhang, liuj}@tsinghua.edu.cn

† China General Technology Research Institute

Beijing 100084, China

Email: liuya088@mail.ustc.edu.cn

Abstract—It is difficult for a language model (LM) to perform well with limited in-domain transcripts in low-resource speech recognition. In this paper, we mainly summarize and extend some effective methods to make the most of the out-of-domain data to improve LMs. These methods include data selection, vocabulary expansion, lexicon augmentation, multi-model fusion and so on. The methods are integrated into a systematic procedure, which proves to be effective for improving both n-gram and neural network LMs. Additionally, pre-trained word vectors using out-of-domain data are utilized to improve the performance of RNN/LSTM LMs for rescoring first-pass decoding results. Experiments on five Asian languages from Babel Build Packs show that, after improving LMs, 5.4-7.6% relative reduction of word error rate (WER) is generally achieved compared to the baseline ASR systems. For some languages, we achieve lower WER than newly published results on the same data sets.

Keywords-language modeling; speech recognition; low-resource languages; data augmentation;

I. INTRODUCTION

Limited speech and transcripts often lead to poor performance of automatic speech recognition (ASR) systems [1]. In the past decade, special attention has been paid to ASR in this low-resource condition. IARPA BABEL program¹ is aimed to improve the performance of ASR and keyword search (KWS) with limited transcribed speech. Additionally, Open Keyword Search (OpenKWS) and Open Speech Analytic Technologies (OpenSAT) evaluation series² encourage researchers to explore novel methods for ASR in low-resource condition.

Language model (LM) is one of the key components of ASR systems, and the performance of LM is crucial for ASR systems. Researchers have made great efforts to improve LMs in low-resource speech recognition. The most commonly used and the simplest method is to acquire more training text data (out-of-domain data) from other resources [2]. With the explosive growth of data on the Internet, a large amount of textual data is available, even for most of the minority languages [3]. Besides, texts generated by machine translation have also been experimented for low-resource language modeling [4]. Nowadays, translation softwares have been able to support

translation of many low-resource languages, with which we can easily translate in-domain texts from a common language, such as English, into the target language, such as Georgian. The translated texts prove to be helpful for data augmentation. However, the improvement is not so notable in previous work [4]. In addition, LMs based on sub-word units or even characters are proposed [5][6]. In this way, the training data can be considered to be more adequate and out-of-vocabulary (OOV) rate is reduced. However, these methods are not suitable for word-based ASR systems.

After an investigation into previous work, we find that current methods of utilizing out-of-domain data are still unsystematic and not thoroughly summarized. There is still no general procedure to take the most advantage of out-of-domain data to improve LMs. In this paper, we summarize and extend some effective methods, which prove to be generally effective for improving LMs across different languages and ASR systems.

The rest of the paper is organized as follows. In the following section, we introduce our general procedure for improving the word-level LMs, and explain how each step is performed in detail. The whole procedure is shown in Fig. 1. In Section III, we take Georgian as an example to improve the LMs step by step, along with detailed analyses and evaluation. Experiments on other Asian languages in BABEL program are demonstrated in Section IV. Finally, we conclude our work in Section V.

II. METHODS

A. Out-of-Domain Data Acquisition

For data acquisition, the most efficient method is to get more texts from the websites in the language that we need. After properly processed, high-quality texts can be retrieved [2][3]. Translated text can also be useful, but the text quality may be unsatisfactory because of the limit of translation performance [7]. As for text generated by LMs, they are not so effective in low-resource condition, because sentences with grammatical errors are likely to be generated [8]. Based on the previous work and our validation, we highly recommend web texts as the first data source and translated text as an auxiliary for data augmentation.

¹<https://www.iarpa.gov/index.php/research-programs/babel>

²<https://www.nist.gov/itl/iad/mig/opensat>

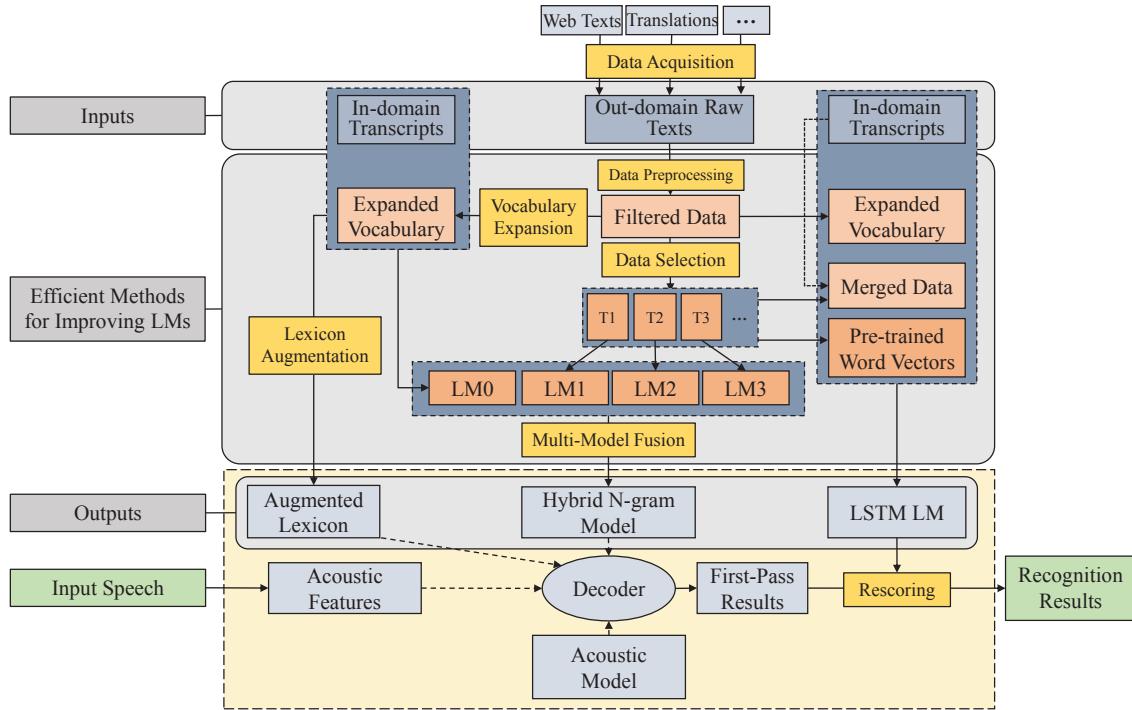


Figure 1. A general procedure for improving LMs in low-resource ASR. **Inputs:** in-domain transcripts and out-of-domain raw text. **Outputs:** an augmented lexicon, a hybrid n-gram LM and an improved LSTM LM. **T1-T3:** text corpus selected by different methods.

B. Data Preprocessing and Selection

The raw text from the Internet cannot be directly used as training materials, because some may be useless and will worsen the LMs in our experiments. What we can do is to select texts based on the similarity to the in-domain transcripts and get rid of irrelevant texts [2][9]. In our work, useless characters and symbols are directly deleted. As for punctuation marks, they can also be removed after sentence segmentation [9]. We regard data selection as a task of text similarity analysis. More methods than previous work [10] for data selection are explored in Section II-B, and three methods achieve satisfactory results and are chosen for improving LMs.

C. Vocabulary Expansion

In low-resource condition, words from in-domain transcripts are not enough to build a powerful vocabulary. Due to the limited size of vocabulary, the OOV rate on development/test data is high and leads to poor performance of LMs. According to previous work, when the vocabulary size is larger, the LM usually performs better because of lower OOV rate. However, if the vocabulary size is too large, LM can be unnecessarily complex and consume more time while decoding [11]. Therefore, efforts should be made to expand the vocabulary appropriately.

The most common method is to increase vocabulary size [9] based on word frequency. In our work, we select a certain number of words with higher frequency in out-of-domain data and merge with original vocabulary. The specific vocabulary size is determined by some primary

experiments on development data, which will be demonstrated in Section III.

D. Lexicon Augmentation

After increasing the vocabulary size, the pronunciation lexicon should be modified accordingly. It is vital that the phonemes in the lexicon should be the same as the original lexicon, because the acoustic models are trained based on these phonemes and the acoustic models should remain invariant since our work focuses on improving LMs. For alphabetic languages like Georgian, a grapheme-to-phoneme (G2P) lexicon modeling method is commonly used to generate lexicon for new words [12]. The graphemes are mapped into a set of phonemes as the basic units for acoustic modeling. We include this grapheme-based method in our procedure to generate pronunciation lexicon for newly added words.

E. Multi-Model Fusion

The selected text corpus in Section II-B can be further used to improve LMs. For neural network LMs, we can simply combine the selected texts with original transcripts as training data [9]. However, for n-gram LMs, we notice that little gains are realized with the same method, especially when the size of selected texts are much more than in-domain transcripts. Therefore, we investigate a multi-model fusion method for n-gram LMs [2] based on language model interpolation. It proves that combining more models with different training corpora can help achieve significant improvements.

In Section II-B, we select a few groups of text using three methods. Then n-gram LMs can be trained separately

instead of training a single LM. These models are also trained with the augmented vocabulary in Section II-C. We interpolate the auxiliary models linearly with the original LM into a hybrid n-gram LM. The interpolation coefficients can be determined by minimizing the perplexity (PPL) on development data [9].

F. Word Vector Pre-Training

Neural Network Language Models (NNLMs) have been widely used for language modeling [13][14][15]. However, when NNLMs are used for low-resource language modeling, the improvement is limited because the parameters are poorly estimated for lack of training data.

Traditionally, word vectors in NNLMs are trainable parameters and are trained with other parameters together. We notice that a few word-embedding methods are used to pre-train the word vectors in NLP tasks. Skip-Gram and CBOW (Continuous Bags of Words) are two widely used methods [16]. We experiment on these two methods to pre-train word vectors in RNN/LSTM LMs, and two kinds of usage of the pre-trained vectors are proposed for different considerations. The first method is to take the pre-trained word-vectors as the final representation and set them as non-trainable parameters. Because if the vocabulary size is extremely large, the number of parameters in the neural network for word vectors will be huge. The other method is to initialize NNLMs with pre-trained word vectors instead of random initialization. The experiments are done on RNN LM and LSTM LMs, and results show that the RNN/LSTM LMs achieve slightly better performance when word vectors are used for parameter initialization.

G. Rescoring

In our work, the hybrid n-gram LM after model fusion in Section II-E are used for first-pass decoding. Then we use the improved RNN/LSTM LMs to rescore the first-pass lattices as shown in [17]. The experiments show that rescoring with LSTM LM can slightly further improve the speech recognition performance.

III. EXPERIMENTS ON GEORGIAN

In this section, we take Georgian as an example to improve the LMs systematically according to the procedure above. We do a series of experiments to demonstrate how each step is performed and evaluate the improvement of each step. For LMs with the same vocabulary size, perplexity (PPL) and Word Error Rate (WER) are both used to evaluate the performance. As for LMs with different vocabulary size, only WER is a reasonable metric [18]. More experiments are done on other low-resource Asian languages in Section IV to test the generality of our procedure.

A. Data

The in-domain Georgian speech and transcripts are from OpenKWS 2016 Surprise Language build pack, including 80-hour training speech in which only 40-hour speech is transcribed, and 10-hour test speech with transcripts. Out-of-domain texts are collected and preliminarily filtered by

BBN WebText Collection System [19]. Details about the in-domain and out-of-domain texts is shown in Table I.

Table I
DETAILS OF DATA FOR GEORGIAN LANGUAGE MODELING. **TIME**: TIME LENGTH OF SPEECH. **UTTS**: NUMBER OF UTTERANCES. **TOKENS**: NUMBER OF TOKENS. **VOCAB**: NUMBER OF UNIQUE WORDS. **TRANS**: IN-DOMAIN TRANSCRIPTS. **WEB**: TEXTS FROM WEBSITES.

	Source	Time	Utts	Tokens	Vocab
Training set	trans	40h	37.7k	314k	30.3k
	web	-	623k	22.8M	1.83M
Test set	trans	10h	9.2k	77.4k	-

B. Baseline ASR System

In order to evaluate the generality of our procedure for improving LMs, we have three baseline systems with different acoustic models. SGMM based acoustic models [20] are commonly used before DNN acoustic models are proposed. DNN and TDNN are two kinds of acoustic models well supported in Kaldi Toolkit [21]. As for acoustic feature, we use bottleneck feature [22] in all three baseline systems.

The pronunciation lexicon of Georgian is not provided in OpenKWS 2016 data, so we extract all unique words from the in-domain transcripts as the original vocabulary, and generate the lexicon with the help of Morfessor [23]. Since the conventional n-gram LM cannot model zero-frequency words in the training data, researchers proposed several smoothing algorithms, including Good-Turing and Kneser-Ney algorithms [24]. An maximum-entropy based n-gram LM also performs well [25]. To find the best method of language modeling, we train several n-gram LMs in a few preliminary experiments using in-domain transcripts. A 3-gram Maximum-Entropy LM (ME3) performs the best among the models and is used as the baseline LM.

In this section, we take an SGMM based ASR system as an example to demonstrate the procedure for improving LMs. Without improving LMs, the WER of SGMM baseline system is 50.7%. Experiments on systems with other acoustic models are demonstrated in Section IV.

C. Procedure of Improving LMs

For out-of-domain data mentioned in Table I, we first filter invalid symbols as mentioned in Section II-B. As for data selection, three methods are compared in the experiments.

The first method is to compute the cross-entropy (CE) difference between in-domain and out-of-domain texts as described in [26] and [10]. More specifically, two 3-gram Maximum-Entropy LMs are trained separately on in-domain and out-of-domain text using the vocabulary of in-domain data. For each sentence in the out-of-domain data, we calculate the cross-entropy using these two LMs and get the difference. The sentences with lower cross-entropy difference are assumed more similar to the in-domain transcripts. XenC [27] tool is used in this method. The other two methods split the out-of-domain texts into

several documents, and transform the documents into vectors using TF-IDF and Doc2Vec [28] respectively. Text similarity is evaluated by cosine similarity based on document vectors. Finally, a certain number of sentences more similar to the in-domain transcripts are selected. It is worth noting that random selection is used as the baseline for comparison.

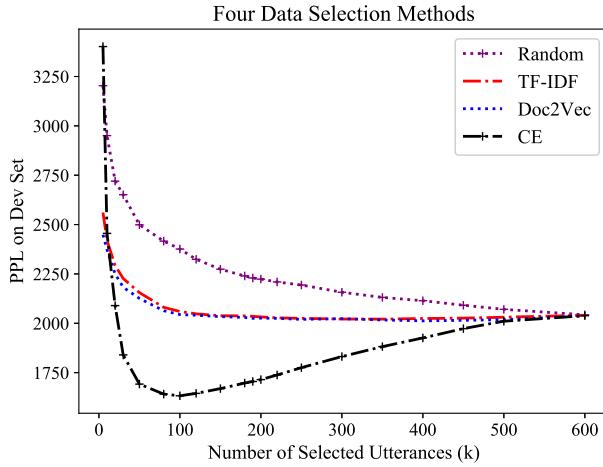


Figure 2. Performance of four data selection methods. Utterances selected from websites are used to train LMs, and evaluate the performance with PPL on development data.

Four groups of n-gram LMs are trained with the sentences selected by different selection methods. The results are shown in Fig. 2. For TF-IDF and Doc2Vec methods, about 200k-300k sentences are enough to reach the lowest PPL, and more data will not further lower the PPL on development data. As for cross-entropy based method, it performs the best and the first 100k sentences achieve the lowest PPL. It is worth noting that the LM performs worse when more data is added. It actually proves that training data should be carefully selected according to the similarity to in-domain data and is not the more the better. Finally, these selected sentences are all kept for further usage in multi-model fusion.

For vocabulary expansion, we directly select words with higher word frequency from the out-of-domain text as described in Section II-C. Meanwhile, we augment the pronunciation lexicon accordingly. Table II shows the results of different vocabulary sizes. As the vocabulary size increases, the OOV rate is significantly reduced and WER is absolutely reduced 3.0% at most. When the vocabulary size is more than 200k, the OOV rate remains around 4.0% and WER remains 47.7%, so 200k words can be assumed enough for vocabulary expansion.

We have the in-domain training transcripts and three groups of texts selected by different methods. Therefore, four different LMs can be trained separately. For language model interpolation, we test with two, three and four n-gram models respectively. The weights (interpolation coefficients) for each model are computed with SRILM Toolkit [29] to minimize the PPL on development data. The results in Table III show that the hybrid model with all

Table II
PERFORMANCE OF LMS WITH DIFFERENT VOCABULARY SIZES.
ADDED VOCAB: THE NUMBER OF WORDS FROM OUT-OF-DOMAIN TEXTS. **MERGED VOCAB:** FINAL VOCABULARY SIZE.

Added Vocab	Merged Vocab	OOV Rate	WER
0(Baseline)	30.3k	8.71%	50.7%
20k	40.5k	7.14%	50.0%
80k	91.0k	5.46%	48.6%
150k	155.3k	4.57%	47.9%
200k	204.0k	4.30%	47.7%
300k	302.0k	4.12%	47.7%
500k	500.2k	3.98%	47.7%

four models performs the best, and WER further reduces 0.8% absolutely.

Table III
PERFORMANCE OF MODEL INTERPOLATION. **TRANS:** IN-DOMAIN TRANSCRIPTS. **CE/DOC2VEC/TF-IDF:** THREE DATA SELECTION METHODS. **PPL:** PERPLEXITY OF LMS ON DEVELOPMENT DATA. **LM0+LM1 AND SO ON:** A HYBRID LM AFTER LINEAR INTERPOLATION.

LMs	Source	Utts	Tokens	PPL	WER
LM0	trans	37.7k	314k	429.30	47.7%
LM1	CE	100k	1.18M	1632.8	-
LM2	Doc2Vec	200k	6.72M	2012.2	-
LM3	TF-IDF	200k	9.38M	2021.5	-
LM0+LM1				393.11	47.1%
LM0+LM1+LM2				390.28	47.0%
LM0+LM1+LM2+LM3				387.32	46.9%

For RNN and LSTM LMs, we select texts from out-of-domain data using the same methods in Section II-B and merge with in-domain training transcripts. The vocabulary is the same as the improved n-gram LM above. Meanwhile, word vectors are pre-trained using Skip-Gram or CBOW methods. Table IV shows that lower PPL can be achieved by LMs initialized with pre-trained word vectors. Another conclusion is that Skip-Gram performs a bit better than CBOW for model initialization.

Table IV
PERPLEXITY ON DEVELOPMENT DATA OF RNN/LSTM LMS USING PRE-TRAINED WORD VECTORS. USAGE OF PRE-TRAINED WORD VECTORS: **A**) NON-TRAINABLE PARAMETERS. **B**) MODEL INITIALIZATION.

	Methods	PPL		
		Baseline	A	B
RNN LM	CBOW	223.18	238.15	222.76
	Skip-Gram		236.22	221.27
LSTM LM	CBOW	217.29	230.53	217.54
	Skip-Gram		228.30	214.70

With hybrid n-gram LM, the WER of Georgian SGMM ASR system is reduced from 50.7% to 46.9%. Then we utilize the improved LSTM LM to do lattice rescoring, and the WER is further reduced to 46.7%. In summary, after all steps in the procedure, the WER of Georgian ASR system is 4.0% absolutely lower and **7.9%** relatively lower than the baseline system.

IV. EXPERIMENTS ON FIVE BABEL LANGUAGES

In order to evaluate the general performance of the whole procedure for improving LMs, we experiment on

Table V
PERFORMANCE OF THE METHODS ACROSS DIFFERENT BABEL LANGUAGES. TRANS: IN-DOMAIN TRANSCRIPTS. WEB: OUT-OF-DOMAIN TEXTS FROM WEBSITES. TDNN: CHAIN TDNN ACOUSTIC MODEL. WER_{pub}: PUBLISHED RESULTS OF WER ON THE SAME DATA SET. ARR: AVERAGE RELATIVE REDUCTION OF WER COMPARED TO BASELINE SYSTEMS AFTER IMPROVING LMs.

Languages	Vocab Size		Tokens		WER (Baseline)			WER (after improving LMs)			WER _{pub}	ARR
	trans	web	trans	web	SGMM	DNN	TDNN	SGMM	DNN	TDNN		
Assamese	22.0k	200k	451.6k	1.08M	52.0	52.6	48.5	48.6	49.2	45.3	**	6.5%
Vietnamese	6.20k	300k	989.1k	4.85M	49.9	49.0	45.1	46.4	45.8	42.4	**	6.5%
Tamil	68.8k	200k	486.4k	6.79M	66.9	66.2	64.9	63.0	62.4	61.8	**	5.4%
Mongolian	20.8k	300k	402.6k	2.48M	56.7	54.5	50.4	53.3	51.1	47.2	48.7 (in [30])	6.2%
Georgian	30.3k	200k	314.1k	9.38M	50.7	50.0	44.5	46.7	46.2	41.3	42.2 (in [30])	7.6%

five low-resource languages in total. The in-domain training transcripts are provided by IARPA BABEL program, and the out-of-domain raw texts are collected and formatted by Leipzig Corpora Collection [31]. Additionally, we have three baseline systems with different setup of acoustic models, so we can evaluate the general performance of improved LMs across different ASR systems.

For each baseline system, we improve the LMs according to the procedure shown in Fig. 1. Table V summarizes the results of experiments on five languages. Some related state-of-the-art results on the same data set are shown in the WER_{pub} column. Significant gains are achieved for each language and baseline system. We evaluate the improvements with an intuitive metric: average relative reduction (ARR) of WER for three baseline systems. ARRs for five languages range from 5.4% to 7.6%. The best performance is on Georgian with a 7.6% ARR. The results prove that the procedure for improving LMs is generally effective in low-resource speech recognition.

V. CONCLUSIONS

We investigate and refine some existing methods to improve the performance of LMs in low-resource speech recognition. Experiments show that high-quality texts can be retrieved from out-of-domain data after pre-processing and selection, which can be used for vocabulary expansion, lexicon augmentation and multi-model fusion. For RNN/LSTM LMs, word vectors pre-trained by out-of-domain texts are used for model initialization. We use the hybrid n-gram LM for the first-pass decoding, and we use improved LSTM LM for lattice rescoring. Experiments show that significant improvements are generally achieved for five languages and three different ASR systems. We believe that it can be taken as a general procedure for improving LMs for low-resource speech recognition.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grant No. U1836219 and the National Key R&D Program of China.

REFERENCES

- [1] A. R. Syed, A. Rosenberg, and M. Mandel, “Active learning for low-resource speech recognition: Impact of selection size and language modeling data,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5315–5319.
- [2] A. Gorin, R. Lileikyte, G. Huang, L. Lamel, J.-L. Gauvain, and A. Laurent, “Language model data augmentation for keyword spotting in low-resourced training conditions.” in *Interspeech*, 2016, pp. 775–779.
- [3] G. Mendels, E. Cooper, V. Soto, J. Hirschberg, M. J. Gales, K. M. Knill, A. Ragni, and H. Wang, “Improving speech recognition and keyword search for low resource languages using web data,” in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.
- [4] G. Huang, T. F. D. Silva, L. Lamel, J. L. Gauvain, A. Gorin, A. Laurent, R. Lileikyte, and A. Messouadi, “An investigation into language model data augmentation for low-resourced stt and kws,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5790–5794.
- [5] H. Sak, M. Sarclar, and T. Gngr, “Morphology-based and sub-word language modeling for turkish speech recognition,” in *International Conference on Acoustics Speech and Signal Processing (ICASSP)*. IEEE, 2010, pp. 5402–5405.
- [6] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, “Character-aware neural language models.” in *AAAI*, 2016, pp. 2741–2749.
- [7] G. Huang, A. Gorin, J.-L. Gauvain, and L. Lamel, “Machine translation based data augmentation for cantonese keyword spotting,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 6020–6024.
- [8] I. Sutskever, J. Martens, and G. E. Hinton, “Generating text with recurrent neural networks,” in *International Conference on Machine Learning (ICML)*, 2011, pp. 1017–1024.
- [9] Z. Zhang, W.-Q. Zhang, K.-X. Shen, X.-K. Yang, Y. Tian, M. Cai, and J. Liu, “Thuee language modeling method for the openkws 2015 evaluation,” in *International Symposium on Signal Processing and Information Technology*. IEEE, 2015, pp. 534–538.
- [10] T. Fraga-Silva, A. Laurent, J.-L. Gauvain, L. Lamel, V.-B. Le, and A. Messouadi, “Improving data selection for low-resource stt and kws,” in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 153–159.
- [11] T. Alume, D. Karakos, W. Hartmann, R. Hsiao, Z. Le, N. Long, S. Tsakalidis, and R. Schwartz, “The 2016 bbn georgian telephone speech keyword spotting system,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5755–5759.
- [12] M. Bisani and H. Ney, “Joint-sequence models for grapheme-to-phoneme conversion,” *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2008.

- [13] A. Ragni, E. Dakin, X. Chen, M. J. F. Gales, and K. Knill, “Multi-language neural network language models,” *Conference of the International Speech Communication Association*, pp. 3042–3046, 2016.
- [14] T. Mikolov, M. Karafiat, L. Burget, J. Cernock, and S. Khudanpur, “Recurrent neural network based language model,” in *Eleventh Annual Conference of the International Speech Communication Association*, 2010, pp. 1045–1048.
- [15] M. Sundermeyer, R. Schlter, and H. Ney, “Lstm neural networks for language modeling,” in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.
- [16] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv preprint arXiv:1301.3781*, 2013.
- [17] S. Kumar, M. Nirschl, D. Holtmann-Rice, H. Liao, A. T. Suresh, and F. Yu, “Lattice rescoring strategies for long short term memory language models in speech recognition,” in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2017, pp. 165–172.
- [18] S. Chen, D. Beeferman, and R. Rosenfeld, “Evaluation metrics for language models,” in *Procdarpa Broadcast News Transcription & Understanding Workshop*, 1998.
- [19] L. Zhang, D. Karakos, W. Hartmann, R. Hsiao, R. Schwartz, and S. Tsakalidis, “Enhancing low resource keyword spotting with automatically retrieved web documents,” in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.
- [20] D. Povey, L. Burget, M. Agarwal, P. Akyazi, F. Kai, A. Ghoshal, O. Gembek, N. Goel, M. Karafit, and A. Rastrow, “The subspace gaussian mixture modela structured model for speech recognition,” *Computer Speech & Language*, vol. 25, no. 2, pp. 404–439, 2011.
- [21] D. Povey, A. Ghoshal, G. Boulian, L. Burget, O. Gembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, “The kaldi speech recognition toolkit,” in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE Signal Processing Society, 2011.
- [22] D. Yu and M. L. Seltzer, “Improved bottleneck features using pretrained deep neural networks,” in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.
- [23] S. Virpioja, P. Smit, S.-A. Grönroos, M. Kurimo *et al.*, “Morfessor 2.0: Python implementation and extensions for morfessor baseline.” Aalto University, 2013.
- [24] S. F. Chen and J. Goodman, “An empirical study of smoothing techniques for language modeling,” *Computer Speech & Language*, vol. 13, no. 4, pp. 359–394, 1999.
- [25] T. Alumae and M. Kurimo, “Efficient estimation of maximum entropy language models with n-gram features: an srilm extension,” in *Eleventh Annual Conference of the International Speech Communication Association*, 2010, pp. 1820–1823.
- [26] R. C. Moore and W. Lewis, “Intelligent selection of language model training data.” in *Proceedings of the Meeting of the Association for Computational Linguistics*, 2010, pp. 220–224.
- [27] A. Rousseau, “Xenc: An open-source tool for data selection in natural language processing,” *The Prague Bulletin of Mathematical Linguistics*, vol. 100, pp. 73–82, 2013.
- [28] Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” in *International Conference on Machine Learning (ICML)*, 2014, pp. 1188–1196.
- [29] A. Stolcke, “Srilm-an extensible language modeling toolkit,” in *Seventh International Conference on Spoken Language Processing*, 2002.
- [30] J. Trmal, M. Wiesner, V. Peddinti, X. Zhang, P. Ghahremani, Y. Wang, V. Manohar, H. Xu, D. Povey, and S. Khudanpur, “The kaldi openkws system: Improving low resource keyword search,” in *Interspeech*, 2017, pp. 3597–3601.
- [31] D. Goldhahn, T. Eckart, and U. Quasthoff, “Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages,” in *Eighth International Conference on Language Resources and Evaluation*, 2012, pp. 759–765.

Effect of Music Training on the Production of English Lexical Stress by Chinese English Learners

Hui Feng

School of Foreign Languages and
Literature,
Tianjin University,
Tianjin, China
fenghui@tju.edu.cn

Jie Lian*

College of Foreign Languages,
Taiyuan University of
Technology,
Taiyuan, China
lianjie0330@163.com

Yingjie Zhao

School of Foreign Languages and
Literature,
Tianjin University,
Tianjin, China
victor_zhao@tju.edu.cn

Abstract—Under the guidance of the Theory of Multiple Intelligences, this study aims to find whether music training can improve English stress production among Chinese English learners without music background. Major findings are as follows. (1) In stress production, music training has significant influence on the stress production by Chinese English learners. Specifically, after music training, there has been evident improvement in pitch and intensity in the training group in distinguishing stressed and unstressed syllables in disyllabic pseudowords. Besides, the accuracy of the production of unfamiliar words in the training group has increased by 11.5% on average, compared with that of the control group which has little change. In addition, little effect of music training on duration proportion in stressed syllables is found in this experiment. (2) Chinese English learners' perception of music can be positively transferred to their production of English lexical stress. Such findings provide further proof for the effect of music training on the production of English lexical stress, and propose a method for Chinese English learners to improve their English pronunciation.

Keywords-music training; stress production; second language speech; theory of Multiple Intelligences

I. INTRODUCTION

For many years, English education in China has focused on memorizing words and grammar knowledge on reading and writing levels, with little attention paid to pronunciation and fluency. With the development of globalization, the basic communicative function of language has become increasingly important. Tone and intonation of language thus have become the fundamental elements for better intelligibility in international communication. However, traditional listening and reading imitation has been proved to be unable to improve the accuracy of English speech production properly (Yang, 2010).

Music and language are two methods unique for human beings to express emotions. The relations between these two are becoming major research topics for linguists, psychologists and neurologists in recent years. It is also believed that musical ability affects language ability and many experiments proved that music training has a positive influence on language study (e.g. Besson & Schon, 2001; Ezquerro, 2012; Marques et al., 2007; Mashayekh & Hashemi, 2011; Pei, 2010; Slevc & Miyake, 2006; Zhang & Chang, 2015). Thus, large potential to study the specific

effect of music training on foreign language pronunciation learning in order to find an effective way to make the best of musical elements in second language speech teaching to improve intonation and pronunciation acquisition.

Proposed by Gardner (1983), the Theory of Multiple Intelligences argues that intelligence is multiple which accounts for a group of abilities instead of one ability, and these abilities interact with each other. In addition, every kind of intelligence is cultivable, creative and practical. Gardner's theory is further tested by Patel (2006, 2008) on the neurological level that music and language share one processing mechanism and neural resources in human brain.

Gardner's theory takes into account the influence of individual difference on teaching, and emphasizes the comprehensive development of all kinds of intelligences and sufficient display of individual talents. Therefore, under the Theory of Multiple Intelligences and Patel's further test, this paper mainly discusses whether music training has a positive effect on the production of English lexical stress for Chinese English learners. The study focuses on investigating the effects of music training on the realization of stressed syllables in English lexical words.

This study puts forward two research questions: 1) Can music training improve Chinese English learners' performance in stress production before and after music training? 2) In which way can Chinese English learners improve their stress production? The detailed hypotheses are as follows.

Hypothesis 1: After music training, there is significant improvement in the production of English lexical stress by Chinese English learners.

Hypothesis 2: English lexical stress produced by Chinese English learners are realized by increasing pitch and intensity and extending duration of the stressed syllables.

II. METHODS

A. Participants

In this study, 20 participants (gender balanced) in the training group (PT) and 10 participants (gender balanced) in the control group (PC) are invited to the production experiments. All participants are from North China with standard Chinese (Mandarin) as their language for daily communication. They all know English phonetic symbols and they have no discernable accent in Chinese. None participant has ever learned to play any musical instrument

* Corresponding author

or been trained in singing or ever received any formal music training for at least three months, but they all have the ability to distinguish musical melody and rhythm. All participants have strong motivation to improve their English pronunciation, and are willing to follow the instructions at each stage of the experiment. They are all in their junior year or above, which means they have no compulsive English class to attend. They have not received any English pronunciation training course in the past 6 months and are not attending any such course while they participate in this experiment.

B. Stimuli

Reading materials are composed of three parts: pre-training recording material, training material and post-training recording material. In the selection of recording materials, the following factors are taken into consideration: 1) words in different syllable patterns; 2) familiarity of English words. In this study, 60 words (including disyllabic words and polysyllabic words) are selected randomly from GRE (Graduate Record Examination) wordlists that none of the participants are familiar with. These 60 unfamiliar GRE words are divided evenly and randomly into pre-training and post-training sessions, with stressed syllables and phonetic symbols marked out.

Pseudowords are created on the basis of phonotactic restrictions in a target language (Glaze, 1928), with no real meaning in the lexicon which can not only avoid familiarity of previously known pronunciation, but also imposes strict control on syllabic structures. In this experiment, pseudowords are designed in disyllabic structures “V.CV” and “CV.CV”, with voiced alveolar plosive [d] and voiceless alveolar plosive [t] and five fundamental vowels [a][i][u][e][o]. Since “e” in the words ended with “de” are not pronounced in English, “ade, ede, ide, ode, ude, tade, tede, tide, tode, tude” are removed. In total, there are 40 tokens of pseudowords. Later, 40 pseudowords are doubled to 80 tokens with stressed syllable marked in capitalized letters, such as “Ada /'ʌdə /” and “aDA /ə'də/”.

C. Procedure

Before and after music training, all participants in PT and PC are invited to record 30 unfamiliar words in random order. Three English native speakers are invited to assess whether the stress of each word is correctly produced. The mean accuracy scores are used to evaluate the stress production of participants before and after the music training.

After pre-training recording, PT are given weekly music training and recorded randomly after training, including reading pseudowords independently and reading pseudowords in a fixed sentence structure. The music training lasts for 8 weeks including the pitch and loudness perception of musical notes produced from piano, violin, flute and triangle. As for PC, 80 pseudowords tokens are only recorded in the first and last week of music training.

Two months after the recording and production experiments, 8 participants in PT are invited to a follow-up experiment to read 80 pseudowords and 30 unfamiliar words again to test the long-term effect of music training on English stress production. In the follow-up experiment, recordings are scored by three English native speakers to assess the stress placement.

D. Data Extraction and Processing

All sound recordings are first automatically annotated, then syllable boundaries of each word are modified manually in language analysis software Praat. In this study, fundamental frequency (F0), intensity (INT) and duration (DUR) of each target word are extracted; the units of pitch, intensity and duration are hertz (Hz), decibel (dB) and millisecond (ms) respectively.

For disyllabic pseudowords, $\Delta F0$, ΔINT and $\% DUR$ are calculated to measure the acoustic differences between stressed and unstressed syllables:

$$\Delta F0_i = F_{0i} - f_{0i} \quad (1)$$

$$\Delta INT_i = INT_i - int_i \quad (2)$$

$$\% DUR_i = \frac{DUR_i}{Dur_w} \times 100\% \quad (3)$$

where F_{0i} , INT_i , DUR_i are F0, intensity and duration of the stressed syllables; f_{0i} , int_i are F0 and intensity of the unstressed syllables; Dur_w is the entire duration of the target word. $\Delta F0$ is the difference between F0 in stressed syllable and F0 in unstressed one in a disyllabic pseudoword; ΔINT is the difference between intensity in stressed syllable and intensity in unstressed one in a disyllabic pseudoword; $\% DUR$ is the proportion of the length of stressed syllable in the whole word length.

III. RESULTS AND DISCUSSION

In stress perception, small variation is found in the accuracy of English stress perception after music training. Before music training, all participants’ ability in English lexical stress perception is generally at a high level (accuracy around 90-95%) with no significant individual differences, while after music training, stress perception accuracy in the training group is still around 90-95% with little variation.

In this study, participants’ stress production data are collected by recording pseudowords and lexical words. Before music training, it is noticed that 7 participants in the training group have difficulties in stress realization on the condition that they know where the stressed syllable should be, but they do not know how to realize the intended stress. They tend to produce higher pitch in the first syllable, stronger intensity and longer duration in the second syllable, which makes it even hard for listeners to identify stress placement. Therefore, in this part, the 20 participants in the training group are divided into Group A (7 participants with difficulty in stress realization) and Group B (13 participants with little difficulty in stress realization).

A. Production of Pseudowords

Generally, in disyllabic words, the stressed syllables are pronounced with higher frequency, stronger intensity and longer duration compared with the unstressed syllables (Fry, 1958). As for the production of F0 and intensity in pseudowords in this study, $\Delta F0$ and ΔINT are the main parameters to indicate the prominence of pitch and intensity of the stressed syllables in disyllabic tokens. The mean $\Delta F0$ and ΔINT changes of Group A and Group B in PT before and after music training are shown in Fig. 1, data are analyzed in t-test.

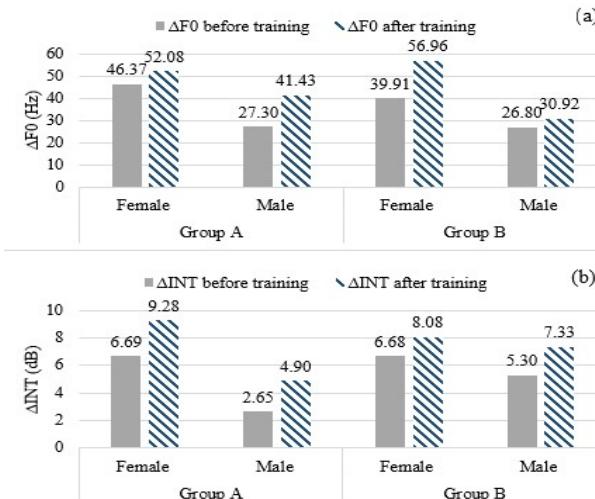


Figure 1. Mean $\Delta F0$ and ΔINT changes of PT before and after music training

Fig. 1 reveals that, overall, there is significant improvement in $\Delta F0$ ($t=3.25$, $P=.047$) and ΔINT ($t=8.27$, $P=.003$) in disyllabic pseudowords produced by the two groups of PT after music training. Female participants in Group B make greater progress in $\Delta F0$ than that of female participants in Group A (17.05Hz vs. 5.71Hz), while male participants in Group A performed better in F0 production than male participants in Group B (14.1Hz vs. 4.1Hz). Besides, participants in Group A have greater enhancement in ΔINT than that of participants in Group B (2.58dB vs. 1.40dB for female, 2.25dB vs. 2.03dB for male). That is to say, in intensity production, the improvement of participants in Group A is more significant.

Z-score normalization (standard score) method is utilized in this part to eliminate the effect of gender differences in F0 and intensity. The normalized mean $\Delta F0$ and ΔINT , and %DUR of tokens with word-initial stress and tokens with word-final stress produced by PT and PC before and after music training are listed in Table I.

Results in Table I indicate that for participants in the training group, there is significant improvement ($t=4.34$, $P=.022$) in $\Delta F0$ and ΔINT of disyllabic tokens after music training. On average, PT have improved by 22.03% in $\Delta F0$ and 79.67% in ΔINT in stressed syllables in tokens with word-initial stress; they have improved by 60.44% in $\Delta F0$ and 16.59% in ΔINT in stressed syllables in tokens with word-final stress. However, for the control group, there is little change ($t=1.84$, $P=.161$) in $\Delta F0$ and ΔINT of disyllabic tokens. Such findings suggest that there is significant enhancement in the prominence of pitch and intensity of the stressed syllables produced by the training group after they received music training.

As for duration production, on average, before music training, all participants put less than half of time (PT 47.34%, PC 46.16%) on the stressed syllables when they pronounce tokens with word-initial stress, while they produce much longer (PT 69.52%, PC 68.77%) stress in tokens with word-final stress which may resulted from the phenomenon that final-stressed syllables will extend duration to realize focus. The proportion results show little changes in duration proportion of stressed syllables after music training in both groups. Such findings suggest that

music training has little influence on the duration production of stressed syllables in disyllabic tokens.

TABLE I. $\Delta F0$, ΔINT and %DUR of disyllabic tokens production

	$\Delta F0$	Initial stress		Final stress		rate(%)
		before	after	before	after	
PT	ΔINT	0.39	0.71	79.67	0.30	0.35
	%DUR	47.34	48.46	2.37	69.52	68.23
						-1.86
PC	$\Delta F0$	1.75	1.75	0.16	1.57	1.52
	ΔINT	0.83	0.84	0.44	0.67	0.66
	%DUR	46.16	45.78	-0.82	68.77	69.05
						0.41

Notes: here, rate= $100\% \times (\text{after} - \text{before}) / \text{before}$.

In the follow-up experiment, 8 participants still maintain the improvement of pitch and intensity in stress production of disyllabic tokens as they did at the end of the music training (average improvement 40.4% vs. 44.7% without significant statistical difference, $t=1.872$, $P=0.473$).

To sum up, compared with the control group which receives no music training, there is significant improvement in stress prominence of F0 and intensity between stressed and unstressed syllables in disyllabic pseudowords produced in the training group after music training. Little effect of music training on duration proportion in stressed syllables is discovered in this experiment.

B. Production of Lexical Words

Participants in both the training group and the control group are invited to read 30 unfamiliar words with stress and phonetic symbols marked out before music training, and read another group of 30 unfamiliar words after training. Audio recordings are sent to three English native speakers to assess whether the stress placement is correct.

The accuracy of different groups in PT before and after music training is calculated and presented in Fig. 2. It can be indicated from Fig. 2 that overall accuracy of unfamiliar words has increased by 11.5% (=88.83%-77.33%) which means there is overall enhancement in PT's accuracy of reading unfamiliar words after music training (88.83% vs. 77.33%; $t=6.97$, $P=.000$) with statistical significance. Group B performs better than Group A in reading unfamiliar words (82.82% vs. 67.14% before training, 91.03% vs. 84.76% after training) with no statistical significance, while Group A makes greater progress in words accuracy (17.62% vs. 8.20%).

Accuracy (numbers of the words with correct stress pronunciation) in pre-training and post-training records of the training group and the control group are analyzed in SPSS for paired-samples t-test. The results of t-test show that, there exist significant differences in the training group in the accuracy of reading unfamiliar words before and after training ($t=2.09$, $P=.000$), while in the control group, the difference in the accuracy of reading unfamiliar words is not remarkable ($t=2.26$, $P=.343$). Such findings indicate that PT's accuracy in reading unfamiliar words have improved after music training.

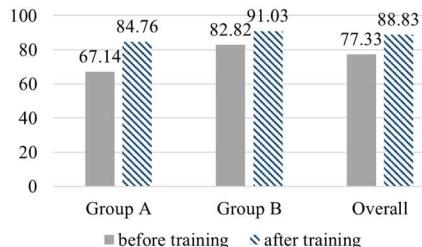


Figure 2. Mean accuracy (%) of reading unfamiliar words by PT

In the follow-up experiment, eight PT's recordings are scored by three English native speakers to identify the stress placement. The accuracy in the follow-up experiment is still over 82%.

C. Effect of Music Perception on Stress Production

After musical pitch and loudness distinguishing training, PT produce stress by increasing $\Delta F0$ and ΔINT between the stressed and unstressed syllables, which means that there is significant improvement in participants' stress production in disyllabic pseudowords after music training. In this study, the mean accuracy improvement in musical pitch and loudness identification before and after music training is reckoned as the music perception ability of PT. The rate of $\Delta F0$ and ΔINT in disyllabic pseudowords for each participant (variations between mean $\Delta F0$ and ΔINT before training, and mean $\Delta F0$ and ΔINT after training) is defined as variation of the improvement of speech production. Variation values of accuracy of musical pitch and loudness as well as $\Delta F0$ and ΔINT in disyllabic pseudowords are normalized to eliminate random differences. Fig. 3 presents the correlation between perception of music and production of pseudowords.

As is shown in Fig. 3(a), variation value of pitch perception of music is basically in positive linear correlation with variation value of F0 production of pseudowords ($r=.874$, high correlation). In other words, most PT with higher perception in musical pitch produce greater $\Delta F0$ in stressed syllables of disyllabic pseudowords, which indicates that in terms of pitch, the improvement in music perception can be mostly transferred to stress realization in speech production. Fig. 3(b) indicates that, variation value of loudness perception of music is basically in positive linear correlation with variation value of intensity production of pseudowords ($r=.746$, moderate correlation). That is to say, the majority of PT with higher perception in musical loudness produce larger ΔINT in stressed syllables of disyllabic pseudowords, which indicates that in terms of loudness, the improvement in music perception can be largely transferred to the stress realization in speech production.

Therefore, it can be summarized that after music training, PT with greater progress in musical pitch and loudness perception could produce stress in a more significant way by increasing F0 and intensity in the stressed syllables. As for PT with little improvement in music perception, their enhancement in the production of stressed syllables is not quite obvious. In other words, PT with higher sensitivity in music are more likely to benefit from music training to improve their stress production in speech.

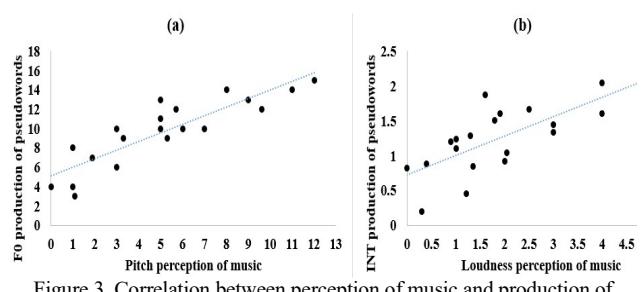


Figure 3. Correlation between perception of music and production of pseudowords

IV. CONCLUSION

Based on Gardner's Theory of Multiple Intelligences, this study aims to find whether Chinese English learners without music background can improve their production of English stress after a set of music training schedule by conducting phonetic experiments. Major findings are as follows. (1) In stress production, there is significant influence of music training on the stress production by Chinese English learners. After two-month music training, the training group make great progress in distinguishing stressed and unstressed syllables in disyllabic pseudowords. They mainly achieve it by increasing pitch (around 42%) and intensity (around 87%) in the stressed syllables. Such improvement is not found in the control group. Besides, the accuracy of the production of unfamiliar words in the training group has increased by 11.5% on average, compared with the control group which has little change. In addition, the proportion results show small variation in duration proportion of the stressed syllables after music training in both PT and PC, which means that little effect of music training on duration proportion in stressed syllables is discovered in this experiment. (2) Chinese English learners' perception of music can be positively transferred to the production of English lexical stress. After music training, participants in the training group with greater progress in music perception can produce stressed syllables in a more significant way by increasing pitch and intensity.

Such findings provide further proof for the effect of music training on the production of English lexical stress, and propose a learning method for Chinese English learners to improve their English pronunciation. The set of music training which has been proved effective can be applied to language pronunciation classes, and teachers can use these musical methods to improve the efficiency of language teaching.

ACKNOWLEDGMENTS

The research is supported partially by the Major Program of National Social Science Foundation of China (No. 15ZDB103), and the Social Science Foundation of Tianjin, China (No. TJWW17-010 and TJWW19-009).

REFERENCES

- [1] J. Yang, "A study of the effects of repeated listening and imitation on the production of English stress patterns," *Foreign Languages Research*, vol. 4, pp. 9-16, April 2010.
- [2] M. Besson, D. Schon, "Comparison between language and

- music,” Annuals New York Academy of Sciences, vol. 930, pp. 232-258, 2001.
- [3] A. O. Ezquerro, “Exploring the relationship between musical aptitude and degree of comprehensibility in L2 speech: Assessment of a group of English learners of Spanish in the English secondary school system,” Porta Linguarum, vol. 17, pp. 223-236, 2012.
- [4] C. Marques, S. Moreno, S. L. Castro, M. Besson, “Musicians detect pitch violation in a foreign language better than nonmusicians: Behavioral and electrophysiological evidence,” Journal of Cognitive Neuroscience, vol. 19, pp. 1453-1463, 2007.
- [5] M. Mashayekh, M. Hashemi, “The impacts of music on language learners’ performance,” Procedia - Social and Behavioral Sciences, vol. 30, pp. 2186-2190, 2011.
- [6] Z. Pei, An Empirical Study on the Effects of Musical Aptitude on EFL Phonological Learning. Nanjing: Nanjing University, 2010.
- [7] R. Slevc, A. Miyake, “Individual differences in second-language proficiency: Does musical ability matter?” Psychological Science, vol. 17, pp. 675-681, 2006.
- [8] H. Gardner. Frames of Mind: The Theory of Multiple Intelligence. New York: New York Basic Books, 1983.
- [9] A. D. Patel, “Musical rhythm, linguistic rhythm, and human evolution,” Music Perception, vol. 24, pp. 99-104, 2006.
- [10] A. D. Patel, Music, Language, and the Brain. Oxford: Oxford University Press, 2008.
- [11] J. A. Glaze, “The association value of non-sense syllables,” Pedagogical Seminary and Journal of Genetic Psychology, vol. 35, pp. 255-269, 1928.
- [12] D. Fry, “Experiments in the perception of stress,” Language and Speech, vol. 1, pp. 120-152, 1958.
- [13] H. Zhang, H. Chang, “Effects of musical training on Chinese learners’ perception of English suprasegmental,” Second Language Learning Research, vol. 1, pp. 73-83, 2015.

Carrier Sentence Selection with Word and Context Embeddings

Chak Yan Yeung, John Lee and Benjamin Tsou

Department of Linguistics and Translation

City University of Hong Kong

chak.yeung@my.cityu.edu.hk, jsylee@cityu.edu.hk, rlbtsou@cityu.edu.hk

Abstract—This paper presents the first data-driven model for selecting carrier sentences with word and context embeddings. In computer-assisted language learning systems, fill-in-the-blank items help users review or learn new vocabulary. A crucial step in automatic generation of fill-in-the-blank items is the selection of carrier sentences that illustrate the usage and meaning of the target word. Previous approaches for carrier sentence selection have mostly relied on features related to sentence length, vocabulary difficulty and word association strength. We train a statistical classifier on a large-scale, automatically constructed corpus of sample carrier sentences for learning Chinese as a foreign language, and use it to predict the suitability of a candidate carrier sentence for a target word. Human evaluation shows that our approach leads to substantial improvement over a word co-occurrence heuristic, and that context embeddings further enhance selection performance.

Keywords-computer-assisted language learning; carrier sentence; word embeddings; context embeddings;

I. INTRODUCTION

Learners of a foreign language often wish to practice and reinforce their linguistic knowledge with exercises. Traditional textbooks, however, can provide only a limited quantity of exercises. In the field of intelligent computer-assisted language learning (ICALL), the task of automatic exercise generation can address this need [1]. Automatic generation can potentially provide users not only with an increased quantity of exercises, but also exercises that fit their needs better in terms of pedagogical level and personal interest.

Many ICALL systems offer fill-in-the-blank (FIB) items, also known as cloze or gap-fill items, as exercises. FIBs are constructed on the basis of *carrier sentences* [2], which are also referred to as seed sentences [1]. As shown in Table I, the system blanks out one word — called the *target word* — in the carrier sentence, and asks the learner to fill in the blank, typically by selecting from multiple choices that include the target word and several distractors.

FIB items may be used for multiple purposes. When supporting language assessment, they should be able to discriminate among students at different proficiency levels. For this purpose, it would be appropriate to select challenging carrier sentences, for example those with complex syntactic structure, difficult vocabulary, or unusual usage of the target word.

In the context of language learning, which forms the focus of this paper, FIB items are intended to help users review the target word and understand it better. Therefore, relatively short, simple carrier sentences that illustrate a

Fill-in-the-blank item	Component
He stayed home because of the wet ____.	Carrier sentence
A. air	Distractor
B. climate	Distractor
C. weather	Target word
D. whether	Distractor

Table I
COMPONENTS OF AN EXAMPLE FILL-IN-THE-BLANK ITEM WITH TARGET WORD “WEATHER”

common usage of the target word are generally considered the most effective [3], [4].

Carrier sentences are traditionally composed by hand. In order to reduce the cost of manual composition and to take advantage of authentic material for language learning, there has been increasing interest in automatic selection of carrier sentences from large corpora [2], [5], [6]. This paper investigates data-driven methods to assess the suitability of candidate carrier sentences for language learning.

II. CARRIER SENTENCE SELECTION

The growing size of available digital corpora means that one can find sample sentences for almost any target word. It does not suffice, however, to randomly select a sentence to serve as carrier sentence. Consider the candidate carrier sentences in Table II for the word “weather”. Sentence (a) is probably too complex for the user to understand, given that the user is learning the word “weather” and therefore unlikely to be an advanced speaker of English. Sentence (b) is easy to understand, but is not a good carrier sentence since it does not provide a good illustration of the target word. Sentence (c), a straightforward sentence with the collocation “weather” and “wet”, is the most appropriate carrier sentence among the three.

One line of research is to adopt readability assessment algorithms to the sentence level [7]. Automatic readability assessment assigns each paragraph or sentence a grade, or score, that indicates its linguistic complexity [8], [9], [10]. This task is related to carrier sentence selection in that it can help filter out sentences with difficult vocabulary or structure, e.g., sentence (a) in Table II. Readability assessment, however, cannot identify simple sentences that do not provide a good illustration of the meaning of the target word. Sentence (b) in Table II is one such example. Proper assessment for carrier sentences, similar to the selection of example sentences for dictionaries, requires

Suitability	Candidate carrier sentence
×	(a) Humid ___ is imminent as the low-pressure system approaches.
×	(b) He likes to talk about the ___.
✓	(c) He stayed home because of the wet ___.

Table II

EXAMPLE CANDIDATE CARRIER SENTENCES FOR THE TARGET WORD
“WEATHER”

taking into account the relation between the target word and the rest of the sentence. To identify sentence (c) in Table II as a promising candidate, for example, the system needs to recognize that the target word is associated with one of its properties (“wet”) and a consequence (“stayed home”).

Because of the lack of large-scale, annotated datasets for carrier sentences, previous approaches either relied on heuristics [11], [12], [13] or applied machine learning to mostly surface features such as sentence length and vocabulary difficulty. This paper presents and evaluates the first data-driven model for selecting carrier sentence that exploits word and context embeddings. We compile a large-scale corpus, of close to 868K carrier sentences in Chinese harvested from online material, to train a statistical classifier. Experimental results show that combining context embeddings and word-occurrence statistics leads to optimal performance, outperforming baseline classifiers as well as the best heuristic reported in previous work.

The rest of the paper is organized as follows. The next section summarizes previous work. Section IV describes our dataset. Section V presents our classification approach. Section VI reports cross-validation results and motivates our feature selection. Section VII reports results on a human-annotated evaluation dataset. Finally, Section VIII concludes and outlines future work.

III. PREVIOUS WORK

This section first summarizes heuristic and rule-based approaches on carrier sentence selection (Section III-A), and then describes more recent work on machine learning approaches (Section III-B).

A. Heuristic approaches

Volodina et al. [11] proposed an algorithm that uses weighted heuristics to score each candidate sentence in Swedish. These heuristics specify the range of acceptable sentence lengths; penalize the presence of words below a word frequency threshold; and the preferred location of the target word. In manual evaluation, 56.6% of the sentences were considered “acceptable”. The fourth requires the presence of finite verbs. A subsequent study considered additional heuristics involving context independence and sentence structure, as well as machine learning for L2 complexity. In an evaluation, 80% of the selected sentences were satisfactory in terms of context independence, and 64% adhered to the expected CEFR level [14].

A more recent study further investigated the use of lexical similarity and word co-occurrence measures [13].

Type	Suitable	Unsuitable
# sentences	867,838	867,838
# target words	19,845	19,845
# sentences / target word	43.73	43.73
Sentence length	12.03	26.20

Table III

STATISTICS ON THE SUITABLE AND UNSUITABLE CARRIER SENTENCES IN THE TRAINING SET.

The algorithm first filtered out unsuitable sentences with heuristics similar to those proposed by [11]. Two criteria were then evaluated for sentence selection. The first optimizes lexical similarity, as approximated by word2vec scores between the target word and the other words in the sentence. The second focuses on word co-occurrence, as measured by pointwise mutual information (PMI) between the target word and other words in the sentence. Human evaluation shows that the PMI criterion identified carrier sentence of higher quality.

B. Machine learning approaches

Recent research has also begun to investigate machine learning approaches. A Croatian corpus of 1094 sentences was annotated on a four-point scale from “Very good” to “Very bad”. A Random Forest regressor was trained on this corpus to predict the quality of example sentences, using 23 variables corresponding to sentence and word length, frequency and other properties, proportions of words in certain parts-of-speech (POS) such as pronouns, proper nouns and conjunctions, as well as syntactic complexity, as measured by parse tree depth. The best model achieved 89.3% precision in selecting the top three sentences [15].

For German, a supervised machine learning approach was developed to refine the results of a heuristics-based approach for selecting example sentences for dictionary headwords [6]. An SVM classifier, trained on a corpus of 13,000 example sentences annotated as “Good” or “Bad”, used features based on lemma and part-of-speech subsequences, as well as sub-tree patterns from constituent trees. Experimental results show that the classifier achieved 68% accuracy [16].

IV. DATA

We constructed a training set consisting of an equal number of suitable carrier sentences and unsuitable ones (Section IV-A), and a human-annotated test set (Section IV-B). We will use the training set to select features for the classifier (Section VI), and then test the classifier on the human dataset (Section VII).

A. Training set

Since there is no large-scale corpus of sentences annotated with their suitability as carrier sentences, we constructed our training set automatically from online resources. Detailed statistics of this set are shown in Table III.

1) *Suitable carrier sentences*: Many websites offer exercises for learners of Chinese as a foreign language, usually in the form of flashcard-style questions and fill-in-the-blank items.¹ However, these resources offer only limited amounts of example sentences. To collect a large dataset, we used sentences on the *Zaoju Cidian* website², which provides example sentences for Chinese word usage, as samples of suitable carrier sentences.

To identify candidate target words for which language learners are likely to request fill-in-the-blank exercises, we retrieved the 40,000 most frequent words in Chinese Wikipedia, excluding words on the stopword list.³ We retained only those target words for which the *Zaoju Cidian* website provides at least 30 example sentences. This process harvested a total of there 19,845 target words, with an average of 43.73 example sentences each.

2) *Unsuitable sentences*: For each target word, we obtained an equal number of sentences from Chinese Wikipedia to serve as samples of unsuitable carrier sentences. We recognize that some Wikipedia sentences can be suitable for this purpose; indeed, some of the sentences that were deemed suitable by human raters in the test set (Section IV-B) were taken from Wikipedia.⁴ While the noise could have been reduced by deliberately choosing low-quality sentences, that would have hurt the classifier’s ability to distinguish between sentences that are merely grammatical and fluent, and those truly suitable as language learning material. We therefore accepted the noise to enable automatic compilation of a large-scale, open-source corpus. Our evaluation results show the effectiveness of this corpus despite the noise.

A significant difference between the sentences from *Zaoju Cidian* and Wikipedia is their length. The average length among the former, with only 12.0 words, is much shorter. Sentence length is an attribute that can be easily controlled and does not require machine learning. We therefore did not use sentence length as a feature (Section V). Instead, we sampled Wikipedia sentences whose length is closest to the average length of the *Zaoju Cidian* sentences.

B. Test set

We used the human-annotated dataset in [13] as test set. This dataset contains 100 target words, each with four carrier sentences: one taken from a textbook, and the other three drawn from Chinese Wikipedia by three different heuristics. To each sentence, two human judges assigned a Word Score (“good”, “fair”, or “unacceptable”) that assesses how well it illustrates a typical usage of the target word, and a Sentence Score on the same scale to assess the grammaticality and fluency of the sentence.

¹E.g., Clavis Sinica (clavisinica.com) and Du Chinese (duchinese.net).

²Accessed at www.ichacha.net/zaoju/ in November 2018.

³We scraped sentences from a dump downloaded in April 2016. The stopword list was taken from <https://github.com/stopwords-iso/stopwords-zh/blob/master/stopwords-zh.txt>.

⁴The training set excludes sentences in the human-annotated test set.

C. Evaluation metric

We converted the two scores in the test set (Section IV-B) into a gold label — either “suitable” or “unsuitable” — for evaluating our classification approach. Sentences annotated as “good” by both judges for both the Word Score and the Sentence Score were labelled as “suitable”; those annotated as “unacceptable” by both judges for *either* the Word Score *or* the Sentence Score were labelled as “unsuitable”.

By this metric, the test set contains 226 “suitable” carrier sentences and 72 “unsuitable” ones, covering all 100 target words.

V. APPROACH

Given an input sentence S that contains the word t , our task is to predict whether S is “suitable” or “unsuitable” as carrier sentence for the target word t . Let W represent the set of words in S , excluding t and those on a stopword list⁵. We investigated two main classes of features.

A. Features on vocabulary difficulty

The following features characterize the level of difficulty of the words in the sentence.

1) *Vocabulary difficulty*: To ensure the learner can understand the sentence, carrier sentences with easier vocabulary items are preferred. Word frequency is often used as a proxy for its difficulty level. We split the 40,000 most frequent words in Chinese Wikipedia into 200 buckets, each with 200 words, then calculate the proportions of words in W that fall into each bucket.

2) *Relative vocabulary difficulty*: Vocabulary difficulty is relative to the user’s proficiency. Rather than attempting to assess the user’s vocabulary proficiency, we determine its upper limit with the following assumption: since the user requested an exercise for word t , he is likely to find that word difficult, and therefore he can be expected to have difficulty understanding other words at the same difficulty level or above. For example, a carrier sentence designed to teach the word “weather” should not assume the learner to know more advanced words, such as “approach”.

A simple rule-based approach would be to reject a sentence if any word $w \in W$ has a lower word frequency than the target word t [13]. This heuristic can lead to unnecessary rejection of good candidate sentences. Instead, our feature allows the classifier to learn the acceptable proportion of words in W that are harder than t from training data.

B. Features on sentential context

The following features characterize the relation between W and the target word t , following the intuition that in a good carrier sentence, one or several words $w \in W$ should form a construction with t that illustrates a typical usage.

⁵We used the list of 748 Chinese function words from <https://gist.github.com/dreampuf/5548203>.

1) *Word co-occurrence*: We measure co-occurrence with pointwise mutual information (PMI). For each word $w \in W$, we calculated $pmi(w, t)$, the PMI score between w and t , as estimated on sentences in Chinese Wikipedia. This feature then takes the highest PMI score.

Jiang and Lee [13] selected sentences that maximized $pmi(w, t)$, and reported that this heuristic yielded the best performance. We will refer to this baseline as the “**word co-occurrence heuristic**”.

2) *Example-based word co-occurrence*: Sentence S is likely to be a good carrier sentence if it contains word occurrences that are also attested among the suitable sentences in our dataset with target word t . Specifically, for each word $w \in W$, we calculate the proportion of suitable sentences that contain w . This feature takes the maximum proportion.

3) *Word embeddings*: Word embeddings have been shown to be effective in measuring word similarity and relatedness in a large range of NLP tasks. For each word $w \in W$, we calculate the cosine similarity between the word embeddings of w and t , taking the maximum score as the feature. We used word embeddings trained by Skipgram with negative sampling on Baidu Encyclopedia [17].

4) *Context embeddings*: Context embeddings have also been shown to be useful in predicting word similarity and relatedness [18]. For each word $w \in W$, we calculate the cosine similarity between the context embeddings of t and the word embeddings of w . This asymmetric measure predicts how likely t or similar words can be found in the context of w . The word embeddings and context embeddings were both trained by Skipgram with negative sampling on Baidu Encyclopedia [17].

VI. EXPERIMENT

To assess the usefulness of the features proposed in Section V, we perform cross-validation on our training set (Section IV-A) to measure their performance in classifying candidate carrier sentences as suitable or unsuitable.

A. Set-up

To ensure our evaluation data represent target words at various levels of difficulty, we split the 19,845 target words in our dataset (Section IV) into four difficulty levels:

- “Level 1”, the easiest, consists of the 5000 words with highest frequencies in Chinese Wikipedia;
- “Level 2” consists of the next 5,000 most frequent words;
- “Level 3” consists of the next 5,000;
- “Level 4” consists of the remaining 4,845 words, and therefore is the most difficult.

We then randomly selected 1,000 words from each of the four difficulty levels for use in this experiment. We trained a logistic regression model with scikit-learn and performed word segmentation with the Stanford Chinese parser [19].

B. Results

Table V shows the overall classification accuracy on 10-fold cross validation, as well as the breakdown into the four difficulty levels. The figures represent the average accuracy for each of the 4,000 target words, i.e. each target word has equal weight.

Vocabulary difficulty achieved 67.35% accuracy when used alone, with each additional feature significantly improving the classification accuracy⁶ and producing the best performance at 83.02% accuracy. Word co-occurrence is shown to be already a strong predictor, yielding an accuracy of 81.13%. The addition of the example-based word co-occurrence statistics from our dataset further improved the accuracy to 82.77%, indicating that our corpus contains word co-occurrence information about specific target words and their carrier sentences that is not covered by PMI scores. Word embeddings features increased the accuracy to 82.91% and the use of context embeddings further improved the accuracy to 83.02%. This matches the previous observation that context embeddings are useful in predicting asymmetric association between words [18].

C. Impact of target word difficulty

Accuracy in carrier sentence quality assessment depends on the difficulty of the target word. When only vocabulary difficulty features were used, the more difficult the target word, the easier it is to predict whether a carrier sentence is suitable for it. In our dataset, most words in a suitable carrier sentence are at a lower difficulty level than the target word. This requirement can be more easily realized when the target words are more advanced. In contrast, easier target words have a smaller pool of carrier sentence candidates that can fulfill this requirement.

The pattern was reversed when word co-occurrence and embeddings features were added in. This could be because more advanced target words tend to have more characteristic word usage, so it is more likely for the carrier sentences and the Wikipedia sentences to have similar contexts, and thus lessening the use of co-occurrence and embeddings features. In contrast, easier target words can be found in a larger pool of different contexts, and so co-occurrence and embeddings features can play a bigger role in distinguishing whether a carrier sentence is suitable.

VII. EVALUATION ON HUMAN DATA

We now evaluate our classifier on human-annotated data. We trained multiple logistic regression models with scikit-learn [20] using features listed in Section V. Table V shows classification results on the test set, contrasting the results obtained when using single features and feature sets; when using the word-occurrence heuristic, which yielded the best result in [13]; and hand-crafted textbook materials.

⁶At $p \leq 0.001$ by McNemar’s test for all cases.

Feature set	All	Level 1	Level 2	Level 3	Level 4
Vocabulary difficulty	67.35%	66.18%	67.72%	67.72%	67.86%
+ Relative vocabulary difficulty	68.09%	64.87%	67.89%	69.76%	69.98%
+ Word co-occurrence	81.13%	82.93%	81.87%	80.98%	78.79%
+ Example-based word co-occurrence	82.77%	84.82%	83.70%	82.80%	79.82%
+ Word embeddings	82.91%	85.13%	83.93%	82.94%	79.63%
+ Context embeddings	83.02%	85.23%	84.16%	83.11%	79.63%

Table IV
CARRIER SENTENCE CLASSIFICATION ACCURACY WITH BREAKDOWN TO DIFFERENT DIFFICULTY LEVELS

Model	Feature(s)	Precision	Recall	F-score
Single-feature	Vocabulary difficulty	0.83	0.49	0.61
	Relative vocabulary difficulty	0.84	0.40	0.54
	Word co-occurrence	0.76	0.92	0.83
	Example-based word co-occurrence	0.83	0.08	0.15
	Word embeddings	0.71	0.53	0.60
	Context embeddings	0.68	0.46	0.55
Feature sets	Word co-occurrence + Relative vocabulary difficulty	0.78	0.85	0.81
	Word co-occurrence + Example-based word co-occurrence	0.76	0.88	0.81
	Word co-occurrence + Word embeddings	0.77	0.93	0.84
	Word co-occurrence + Context embeddings	0.77	0.94	0.85
Heuristic	Word co-occurrence heuristic	0.50	n/a	n/a
Human	Textbook	0.70	n/a	n/a

Table V
CLASSIFIER PERFORMANCE ON IDENTIFYING SUITABLE CARRIER SENTENCES IN THE TEST SET

A. Baseline and human performance

Using our metric (Section IV-B), the word co-occurrence heuristic (Section V-B) performs at 0.50 precision. The human-crafted sentences, drawn from textbooks, were considered “suitable” 70% of the time. Note that the recall for these two approaches cannot be estimated.

B. Word occurrence only

Among single-feature models, word co-occurrence achieved the best F-score (0.83), at 0.76 precision and 0.92 recall. This model, significantly better than any other individual features⁷, is consistent with previous observation of the effectiveness of the use of PMI [13]. However, this model achieves substantially higher precision (0.76) than the word co-occurrence heuristic (0.50) since it was able to learn the lower-bound of suitable PMI scores from the training data in order to prevent false positives. Direct comparison with the heuristic, however, is not possible since its recall is not known.

The addition of the Relative vocabulary difficulty feature degraded the performance of the word co-occurrence model to an F-score of 0.81.

C. Enhancement with context embeddings

Word embeddings were useful in supplementing the word co-occurrence feature, increasing the F-score to 0.84. It was Context embeddings, however, that combined with word co-occurrence to achieve the best results (0.85 F-score), at 0.77 precision and 0.94 recall. The improvement over “Word-occurrence” model is however not statistically significant.⁸, which likely reflected the limited size of the test set.

Since context embeddings predict asymmetric associations between words [18], they indicate how likely the target words can be found in the context of the words in the carrier sentences. They thus in effect predict if there are cue words in the sentence to elicit the correct answer for the blank (i.e., the target word), serving as a good proxy for the ability of the sentence to illustrate the usage of the target word.

VIII. CONCLUSIONS

We have presented the first data-driven model on carrier sentence selection that exploits word and context embeddings. Using a statistical classifier trained on a large corpus of automatically annotated sample sentence, our best model achieved 85% F-score in predicting the suitability of a sentence for a target word. Our experimental results show that context embeddings improve upon word co-occurrence statistics in this task, outperforming the previous best heuristic based on pointwise mutual information statistics.

We have applied this model to select carrier sentences from the LIVAC (Linguistic Variation in Chinese Speech Communities) synchronous corpus [21], as part of our project to build a computer-assisted language learning system for Chinese as a foreign language. In future work, we wish to pursue two directions. First, we intend to investigate carrier sentence selection methods that distinguish between multiple senses of a word [22]. Second, we plan to investigate other features to further improve the results and to evaluate the model on other languages.

ACKNOWLEDGMENT

This work was supported by the Innovation and Technology Fund (Ref: ITS/389/17) of the Innovation and

⁷At $p \leq 0.001$ by McNemar’s test for all cases.

⁸At $p \leq 0.23$ by McNemar’s test.

Technology Commission, the Government of the Hong Kong Special Administrative Region; and by a Strategic Research Grant (#7004941) from City University of Hong Kong. We thank Ka Po Chow for his assistance with this research.

REFERENCES

- [1] E. Sumita, F. Sugaya, and S. Yamamoto, “Measuring Non-native Speakers Proficiency of English by Using a Test with Automatically-Generated Fill-in-the-Blank Questions,” in *Proc. 2nd Workshop on Building Educational Applications using NLP*, 2005.
- [2] S. Smith, P. V. S. Avinesh, and A. Kilgarriff, “Gap-fill Tests for Language Learners: Corpus-Driven Item Generation,” in *Proc. 8th International Conference on Natural Language Processing (ICON)*, 2010.
- [3] T. M. Haladyna, S. M. Downing, and M. C. Rodriguez, “A Review of Multiple-Choice Item-Writing Guidelines for Classroom Assessment,” *Applied Measurement in Education*, vol. 15, no. 3, pp. 309–333, 2002.
- [4] W. Xu, “A Research on Blanked Cloze Exercises in Intermediate TCSL Comprehensive Textbooks Taking Four Textbooks as Examples [in Chinese],” in *L (Proc. 5th Forum of CFL Graduate Students)*. Beijing, China: School of Chinese as a Second Language, Peking University, 2012.
- [5] A. Kilgarriff, M. Husák, K. McAdam, M. Rundell, and P. Rychlý, “GDEX: Automatically Finding Good Dictionary Examples in a Corpus,” in *Proc. EURALEX*, 2008.
- [6] J. Didakowski, L. Lemnitzer, and A. Geyken, “Automatic Example Sentence Extraction for a Contemporary German Dictionary,” in *Proc. EURALEX*, 2012.
- [7] I. Pilán, E. Volodina, and R. Johansson, “Rule-based and Machine Learning Approaches for Second Language Sentence-level Readability,” in *Proc. 9th Workshop on Innovative Use of NLP for Building Educational Applications*, 2014.
- [8] E. Pitler and A. Nenkova, “Revisiting readability: a unified framework for predicting text quality,” in *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2008.
- [9] R. J. Kate, X. Luo, S. Patwardhan, M. Franz, R. Florian, R. J. Mooney, S. Roukos, and C. Welty, “Learning to Predict Readability using Diverse Linguistic Features,” in *Proc. 23rd International Conference on Computational Linguistics (COLING)*, 2010, pp. 546–554.
- [10] Y.-T. Sung, J.-L. Chen, J.-H. Cha, H.-C. Tseng, T.-H. Chang, and K.-E. Chang, “Constructing and validating readability models: the method of integrating multilevel linguistic features with machine learning,” *Behavior Research Methods*, vol. 47, pp. 340–354, 2015.
- [11] E. Volodina, R. Johansson, and S. J. Kokkinakis, “Semi-automatic Selection of Best Corpus Examples for Swedish: Initial Algorithm Evaluation,” in *Proc. Workshop on NLP in Computer-Assisted Language Learning*, 2012.
- [12] I. Pilán, E. Volodina, and R. Johansson, “Automatic Selection of Suitable Sentences for Language Learning Exercises,” in *Proc. EUROCALL*, 2013.
- [13] S. Jiang and J. Lee, “Carrier Sentence Selection for Fill-in-the-blank Items,” in *Proc. 4th Workshop on Natural Language Processing Techniques for Educational Applications*, 2017, pp. 17–22.
- [14] I. Pilán, E. Volodina, and L. Borin, “Candidate Sentence Selection for Language Learning Exercises: from a Comprehensive Framework to an Empirical Evaluation,” *Traitemen Automatique des Langues (TAL) Journal, Special issue on NLP for Learning and Teaching*, vol. 57, no. 3, pp. 67–91, 2017.
- [15] N. Ljubešić and M. Peronja, “Predicting Corpus Example Quality via Supervised Machine Learning,” in *Proc. Electronic Lexicography in the 21st Century Conference (eLex)*, 2015, pp. 477–485.
- [16] L. Lemnitzer, C. Pöltz, J. Didakowski, and A. Geyken, “Combining a Rule-based Approach and Machine Learning in a Good Example Extraction Task for the Purpose of Lexicographic Work on Contemporary Standard German,” in *Proc. Electronic Lexicography in the 21st Century Conference (eLex)*, 2015, pp. 21–31.
- [17] S. Li, Z. Zhao, R. Hu, W. Li, T. Liu, and X. Du, “Analogical Reasoning on Chinese Morphological and Semantic Relations,” *arXiv preprint arXiv:1805.06504*, 2018.
- [18] F. T. Asr, R. Zinkov, and M. Jones, “Querying Word Embeddings for Similarity and Relatedness,” in *Proc. Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 675–684.
- [19] R. Levy and C. D. Manning, “Is it harder to parse Chinese, or the Chinese Treebank?” in *Proc. ACL*, 2003.
- [20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, “Scikit-learn: Machine learning in python,” *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.
- [21] B. K. Tsou and O. Y. Kwong, “LIVAC as a Monitoring Corpus for Tracking Trends beyond Linguistics,” *Journal of Chinese Linguistics Monograph Series*, no. 25, pp. 447–472, 2015.
- [22] A. Geyken, C. Pöltz, and T. Bartz, “Using a Maximum Entropy Classifier to Link “Good” Corpus Examples to Dictionary Senses,” in *Proc. Electronic Lexicography in the 21st Century Conference*, 2015, pp. 304–314.

Multiple-source Entity Linking with Incomplete Sources

1st Qingguo Liu
Beijing University of Posts and Telecommunications Beijing, China
 798011435@qq.com

2nd Shui Liu
Tencent Videos Beijing, China
 16237059@qq.com

3rd Lemao Liu
Tencent AI Lab Beijing, China
 lemaoliu@gmail.com

4th Bo Xiao*
Beijing University of Posts and Telecommunications Beijing, China
 xiaobo@bupt.edu.cn

Abstract—This paper introduces a new entity linking task from a well-known online video application in industry, where both entities and mentions are represented by multiple sources but some of them may be missing. To address the issue of incomplete sources, it proposes a novel neural approach to model the linking relationship between a pair of an entity and a mention. To verify the proposed approach to this task, it further creates a large scale dataset including 70k examples. Experiments on this dataset empirically demonstrate that the proposed approach is effective over a baseline and particularly it is robust to the missing sources in some extent.

Keywords-deep learning; entity link; entity disambiguation; incomplete Sources;

I. INTRODUCTION

Entity linking aims at grounding a mention describing an entity to the target entity in a given knowledge base, and it plays an important role in many downstream NLP tasks such as language understanding and question answering [1], [2]. In conventional entity linking tasks, where mentions and entities are usually represented by a single source such as a sentence [3], one challenge is that a single source is insufficient to capture different aspects of the meaning for an ambiguous entity [4], [5]. Consequently, it is promising to acquire multiple sources to describe both mentions and entities at various levels of granularity and then build a multiple-source linking model. Unfortunately, in real world scenarios, it is costly to collect a large number of mentions and entities with multiple sources, and it is more practical that some sources for mentions and entities are missing while for others are not [5].

This paper introduces a fresh multiple-source entity linking task from an online video application in industry, where sources for entities and mentions are *incomplete*. A natural solution to such incompleteness is the expectation maximization algorithm which treats a missing source as a latent variable and marginalize it through expectation [6]. However, the missing source in our task is a structure variable (i.e. sentence or picture), which takes value from a large space, and thus it is inefficient to sample a value for approximating its expectation [7].

In this paper, we propose an efficient approach to address the aforementioned issue of data incompleteness. The key idea to our approach is a novel neural model representing the sources in a way such that it is less sensitive to the missing sources. To this end, given a

mention and an entity, we firstly represent each pair of sources, one in the mention and the other in the entity, into several vectors and then we employ max-pooling [8] to summarize all those vectors for prediction. Crucially, to make the max-pooling effective, we design a regularization term over the vectors such that their components are comparable. To further verify the proposed approach, we create a large scale dataset ¹including around 70k mention-entity pairs from the log of the video application, each of which is coupled with a label indicating whether the mention is linked to the corresponding entity. We make two-fold contributions as follows:

- We introduce a novel multiple-source entity linking task with incomplete sources from a real world application and create a dataset to facilitate its further research.
- We propose a simple yet effective model to alleviate the issue of data incompleteness, which is robust to the missing sources and delivers substantial improvements over a strong baseline.

II. TASK AND DATASET

A. Task

Suppose $\mathbf{m} = \langle m_1, m_2 \rangle$ denotes a mention represented by two sources m_1 and m_2 , $\mathbf{e} = \langle e_1, e_2 \rangle$ denotes a 2-source entity,² and \mathcal{KB} is a knowledge base \mathcal{KB} consisting of a large number of entities.

Since it is generally cost to collect all sources for mentions and particularly entities in \mathcal{KB} in practice, we propose a new task of **multiple-source entity linking**: linking \mathbf{m} to a particular entity \mathbf{e} in \mathcal{KB} , where some sources are missing for \mathbf{m} and entities from \mathcal{KB} . To solve the task, one typically resorts [9], [10], [11] to a binary model $P(z = 1 | \mathbf{m}, \mathbf{e}; \theta)$ parametrized by θ , which indicates the probability of linking \mathbf{m} to an entity, and find the target entity according to the following steps:

- Retrieve a subset \mathcal{E} from \mathcal{KB} for \mathbf{m} and then rank the top entity $\hat{\mathbf{e}}$ over \mathcal{E} :

$$\hat{\mathbf{e}} = \arg \max_{\mathbf{e} \in \mathcal{E}} P(z = 1 | \mathbf{m}, \mathbf{e}; \theta) \quad (1)$$

¹The dataset will be available at <https://github.com> once this paper is accepted.

²In our dataset there are two sources in total, but our approach in §3 is general to arbitrary number of sources.

- Link \mathbf{m} to $\hat{\mathbf{e}}$ if $P(z = 1 | \hat{\mathbf{e}}, \mathbf{m}; \theta) > 0.5$; otherwise, do not link \mathbf{m} to any entity in \mathcal{KB} .

In this way, the entity linking task resembles a binary classification task whose key solution is to build a binary model $P(z|\mathbf{m}, \mathbf{e}; \theta)$ (see §3), and then optimize its parameter θ via maximal likelihood estimation over a training dataset as follows:

$$\max_{\theta} \sum_i \log P(z^i | \mathbf{m}^i, \mathbf{e}^i; \theta) \quad (2)$$

B. Dataset

To establish the new task, we have to provide a dataset, which is used to train our model $P(z|\mathbf{m}, \mathbf{e}; \theta)$ and testify the learned model.

Our dataset consists of a large number of tuples $\langle \mathbf{m}, \mathbf{e}, z \rangle$, which are collected from the log of a large scaled video search website (top 3 video website) in the world. In our dataset, m_1 is the title of the mention and m_2 is a picture, both of which are two sources describing the mention; e_1 and e_2 are a sentence and picture which describe the entity.

In our dataset, each mention \mathbf{m} is provided by a user to the video search website while an entity \mathbf{e} is the search result from a knowledge base maintained by experts for the video application. The label z for the pair $\langle \mathbf{m}, \mathbf{e} \rangle$ is automatically generated through a heuristic method by analyzing the search log for the mention. Our data is collected from the log of a large scaled video search website (top 3 video website) in the world, in which most of the queries can exactly match a certain name of professionally-generated video content (PGC, such as film, cartoon, variety show or tv program). However, certain part of the clicks are on the PGC related user-generated video content (UGC, such as behind the scenes, movie clips, fragment of PGC). Thus, the queries which exactly match the name of PGC, in the task of entity linking, can be considered as name entity, and the clicked UGC can be considered as documents to be linked. To construct a qualified data set, several heuristic rules are applied to mining the query and click video pair from log data, such as the co-occurrence number of query and UGC, the matching rate between query and the name of PGC, the displayed position of UGC, the click though rate (CTR) over PGC, the total number of query. A real sample from the dataset is shown in Table I

Our dataset contains 70k examples, in which positive examples versus negative ones is 2 to 3. The missing rates of m_2 , e_1 and e_2 are 0.13%, 51.62% and 0.23%, respectively³; and the average lengths of m_1 and e_1 are 22.4 and 251.0. We randomly sample 500 examples from the dataset and manually validate their labels. We find only 3.4% of their labels is not correct, which indicates the quality of our dataset is relatively high. It is worth mentioning that every example in our dataset definitely include the mention title, i.e. m_1 , although other sources for \mathbf{m} and \mathbf{e} may be missing.

³Because m_2 and e_2 can be automatically extracted from video by programs, but e_1 can only be manually annotated

Table I
A REAL SAMPLE FROM THE DATASET.

label(z)	1
mention title(m_1)	怦然星动票房过亿,曝偶像自白特辑!李易峰戏里戏外男神升级。(Fall in love like a star box office over 100 million,idol confession was exposed! Yifeng Li is a male god both inside and outside the drama.)
mention cover(m_2)	
entity description (e_1)	电影《怦然星动》讲述的是明星与经纪人之间的爱情故事。杨幂饰演的田心与李易峰饰演的苏星宇相识于微时,五年后两人已获事业.....(The movie "fall in love like a star" is about a love story between a star and his agent. Mi Yang as tian xin and Yifeng Li as su xingyu met at a young age, five years after two people have been career.....)
entity cover(e_2)	

III. METHODOLOGY

Our network can be roughly divided into two parts. The first part is the Input Representation. We first complete the embedding of the input text and image, and then extract similarity feature vectors between mention and entity in different dimensions. The second part is Label Prediction. We use a fully connected network with Shared weights to achieve semantic alignment of similarity feature vectors. Then, a robust feature fusion method is used to fuse the four similarity feature vectors. Finally, the fused vectors are fed into the full-connected network to output the results. The overall structure of our network is shown in the Figure 1.

A. Input Representation

On our task, the inputs of our model are mention \mathbf{m} and entity \mathbf{e} , both of which include a sentence and a picture as two sources according to our dataset in §2. For simplicity, we suppose the first source (m_1 and e_1) is a sentence while the second one (m_2 and e_2) is a picture. Since the

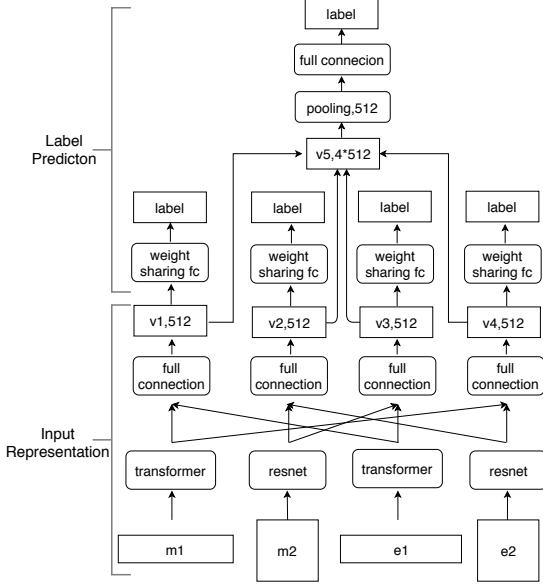


Figure 1. Network structure

sources in \mathbf{m} and \mathbf{e} may be incomplete in practice, we make the following modifications: if the missing source is a sentence, we consider it as a specialized sentence including a single word “`<NULL>`”; if the missing source is a picture, we consider it as a zero matrix with the size identical to that of an ordinal picture. We represent the sentence and picture by using two different networks, i.e. Transformer with multiple-head self attention [12] and Resnet [13].

Specifically, to represent a sentence, we firstly insert a special word `<cls>` into the beginning of this sentence, then employ a Transformer over the resulting sentence to obtain a matrix, and finally we take the row vector from the matrix with respect to `<cls>`, following BERT [14]. Suppose the representation vectors for sentences m_1 and e_1 are respectively denoted by $T(m_1; \theta_T)$ and $T(e_1; \theta_T)$ parametrized by θ_T . To represent the picture, we firstly perform a series of operations such as scaling, clipping, standardization, and then we use a Resnet34 model to obtain a vector. Suppose the representation vectors for pictures m_2 and e_2 are respectively denoted by $R(m_2; \theta_R)$ and $R(e_2; \theta_R)$ parametrized by θ_R . For easier notations, we remove all parameters and denote those four vectors by $T(m_1)$, $T(e_1)$, $R(m_2)$ and $R(e_1)$.

Additionally, for capturing the similarity between \mathbf{m} and \mathbf{e} , we use feedforward neural networks over all pairs of sources from \mathbf{m} and \mathbf{e} and obtain four new vectors. These new vectors are the final representation vectors of \mathbf{m} and \mathbf{e} and are denoted as follows:

$$v_1 = F(T(m_1), T(e_1); \theta_S^1), \quad (3)$$

$$v_2 = F(R(m_2), R(e_2); \theta_S^2), \quad (4)$$

$$v_3 = F(R(m_2), T(e_1); \theta_S^3), \quad (5)$$

$$v_4 = F(T(m_1), R(e_2); \theta_S^4), \quad (6)$$

where F is a feedforward network and θ_S^i is a parameter for v_i such that all v_i has the same dimension ($i = 1, 2, 3, 4$).

B. Label Prediction

To predict the label z for \mathbf{m} and \mathbf{e} , one can summarize all the source representation feature vectors v_1, v_2, v_3 and v_4 together and then apply a sigmoid function σ over the summarized vector as follows:

$$P(z | \mathbf{m}, \mathbf{e}; \theta) = \sigma(\theta_w^\top g(v_1, v_2, v_3, v_4)) \quad (7)$$

where g is a network to summarize vectors v_i , θ_w is a parameter to weight the summarized vector into a scalar.

To specify the definition g , one natural method is to set g as a fully connected neural network F , which is used as the **baseline** in this paper. Unfortunately, baseline is not robust enough to handle incomplete sources in our task because F is very sensitive to its input v_i for all i .

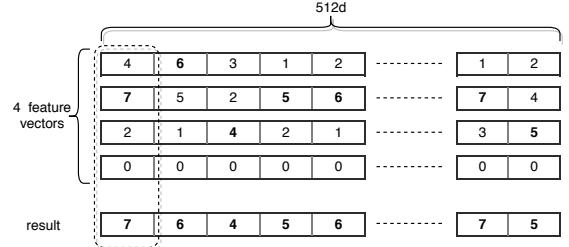


Figure 2. max-pooling instructions

To alleviate this issue, we use max-pooling to summarize v_i , i.e. $g(v_1, v_2, v_3, v_4) = \max(v_1, v_2, v_3, v_4)$, where \max is an element-wise operator over each component of four vectors. Since \max is insensitive to all components in v_i except the largest one, it has the potential to be robust to the missing sources. However, this simple method leads to decreased performance compared with baseline. One possible reason is that each component in different v_i are incomparable by using \max . As a result, we build four auxiliary models with shared parameter θ_a which predicts the label using each v_i as follows:

$$P_a(z | v_i) = \sigma(\theta_a^\top v_i) \quad (8)$$

Then we train our model regularized by the auxiliary models as follows:

$$\max_{\theta} \sum_j \log P(z^j | \mathbf{m}^j, \mathbf{e}^j; \theta) + \sum_{i=1}^4 \log P_a(z^j | v_i^j) \quad (9)$$

where $P(z | \mathbf{m}, \mathbf{e})$ is defined in Eq.(7) with $g = \max$, θ is the overall parameters, and v_i^j is similarly defined in Eq.(6) for the j th example $\langle \mathbf{m}^j, \mathbf{e}^j, z^j \rangle$ in the training set.⁴ It is worth mentioning that for testing, we only use the $P(z^j | \mathbf{m}^j, \mathbf{e}^j)$ to predict the label.

IV. EXPERIMENT

A. Settings

We randomly divide the created dataset into 50k training set, 10k validation set and 10k test set. A two-layer full-connected network including activation function [15]

⁴To train baseline, we set $g = F$ and then remove the regularization terms in Eq. (9).

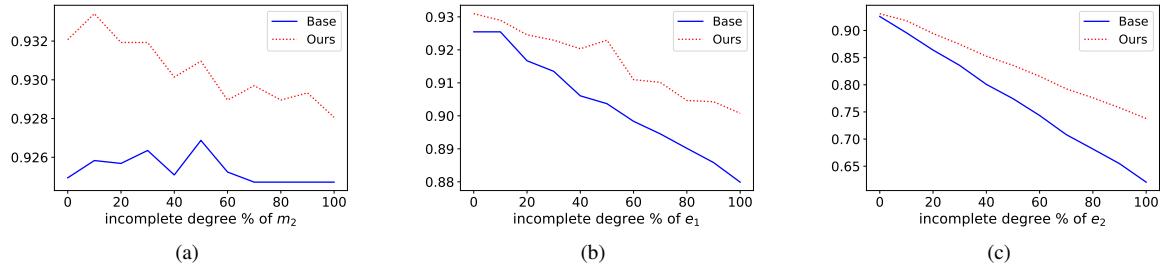


Figure 3. Model accuracy varies with the imperfection of data

and BN [16] is applied to generate the final result. We implement baseline and our models using the Pytorch1.0 framework [17]. All hyperparameters for both models are summarized as shown below. For the sentences(m_1, e_1), they are sampled such that the length is ≤ 128 tokens. We use Token Embeddings and Position Embeddings to encode the sequence. The transformer we use is a six-layer encoder network. Each layer of encoder contains a multi-head self-attention and a feed forward network. The representation($T(m_1), T(e_1)$) obtained from transformer is 768-dimensional. For pictures(m_2, e_2), we scale and crop them to 227*227, then use Resnet34 to generate 2048d representation vectors($R(m_2), R(e_2)$). The similarity feature vectors(v_1, v_2, v_3, v_4) we get from feedforward neural networks are all 512d. The θ_w in eq6 is a two-layer full-connected network. The first layer is 512*128, and the second layer is 128*2. There are Relu activation function and BN between the two layers. We initialize the parameters in Transformer and Resnet with pretrained models from BERT [14] and Resnet34 [13], and then fine-tune all parameters on our training set. During the training, we set the batch size to 48. We choose SGD [18] with momentum of 0.9 as our optimizer. The learning rates for embedding and other parameters are $1e - 4$ and $1e - 3$, respectively. The stopping epoch is selected according to the performance on the validation set for both models.

B. Main Results and Analysis

Table II
THE ACCURACY OF MODELS

Methods	Accuracy (%)
Base	91.83
Ours	92.81
w/o Reg	91.61
w/o MP	92.45

The accuracy of our method (Ours) and baseline (Base) on the test set. “w/o Reg” denotes our method without auxiliary regularization and “w/o MP” is our method by replacing max-pooling with a fully connected feedforward network.

Table II summarizes the main results between our models and baseline. It is clear to see that our model gains about 1 points over baseline. In addition, by removing the regularization terms in Eq. (8), the performance is even worse than that of baseline, which shows that making component comparable is critical for our model as mentioned in §3.2. Furthermore, on the basis of fully

Table III
THE ANALYSIS OF SENTENCE LENGTH

Bins	Length	[1,20]	[20,30]	[30, +∞)
	Percents	27.31%	70.36%	2.33%
Methods	Base	90.11	92.45	93.13
	Ours	91.03	93.55	90.56

The accuracy of our method (Ours) and baseline (Base) on each bin of the test set according to the sentence length and number of missed sources in a mention.

connected network, regularization gains about 0.4 points over baseline.

As we can see in table III, our model shows better performance on short and medium length sentences, while baseline performs better on long sentences. We speculate that long sentences contain more information, while the full-connected network will encourage that by its additive scoring function. Luckily, such an exception case accounts only for 2.3% and thus it does not decrease the positive effects of our model on average.

C. Robustness

We investigate the robustness of our model by examining the accuracy drops if we artificially remove sources from mentions and entities. To this end, we firstly extract 13.5k examples from the overall data as the new test set, each of which include all four sources; and split the remaining data into training and validation sets, which are used to retrain baseline and our model. Note that for both training and validation sets, their sources may be missing. Then we randomly remove one source for each example in the new test set according to a probability taken from 10%, 20%, 30%, 40%, 50%, 60%, 70% and 100%, leading to 21 test sets with at most one missing source for each example.

Figure 3 depicts the accuracy of both models on the test sets according to randomly remove one source from m_2 , e_1 and e_2 . ⁵We find that our model is more robust to the missing source than baseline as more examples include a missing source. Furthermore, removing e_2 decreases more accuracy compared with removing other sources. This finding suggest that e_2 is more important for making accurate prediction.

⁵We do not conduct experiment on randomly removing m_1 , since it always exists in real data.

V. CONCLUSION

This paper introduced a new entity linking task from industry, where mentions and entities include multiple sources and some sources may be missing. It also presented an efficient and effective approach which is robust to the missing sources for this task. To facilitate research on this task, it created a large dataset. Empirical experiments on this dataset demonstrated the proposed approach delivers gains over a strong baseline.

REFERENCES

- [1] R. Das, M. Zaheer, S. Reddy, and A. McCallum, “Question answering on knowledge bases and text using universal schema and memory networks,” *arXiv preprint arXiv:1704.08384*, 2017.
- [2] J. Welbl, P. Stenetorp, and S. Riedel, “Constructing datasets for multi-hop reading comprehension across documents,” *Transactions of the Association of Computational Linguistics*, vol. 6, pp. 287–302, 2018.
- [3] S. Chopra, M. Auli, and A. M. Rush, “Abstractive sentence summarization with attentive recurrent neural networks,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 93–98.
- [4] I. Yamada, H. Shindo, H. Takeda, and Y. Takefuji, “Joint learning of the embedding of words and entities for named entity disambiguation,” *arXiv preprint arXiv:1601.01343*, 2016.
- [5] Y. Eshel, N. Cohen, K. Radinsky, S. Markovitch, I. Yamada, and O. Levy, “Named entity disambiguation for noisy text,” *arXiv preprint arXiv:1706.09147*, 2017.
- [6] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.
- [7] C. Kong, M. Gao, C. Xu, W. Qian, and A. Zhou, “Entity matching across multiple heterogeneous data sources,” in *International Conference on Database Systems for Advanced Applications*. Springer, 2016, pp. 133–146.
- [8] Y.-L. Boureau, J. Ponce, and Y. LeCun, “A theoretical analysis of feature pooling in visual recognition,” in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 111–118.
- [9] S. Upadhyay, N. Gupta, and D. Roth, “Joint multilingual supervision for cross-lingual entity linking,” *arXiv preprint arXiv:1809.07657*, 2018.
- [10] S. Murty, P. Verga, L. Vilnis, I. Radovanovic, and A. McCallum, “Hierarchical losses and new resources for fine-grained entity typing and linking,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 97–109.
- [11] A. Sil, G. Kundu, R. Florian, and W. Hamza, “Neural cross-lingual entity linking,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.
- [13] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [15] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 315–323.
- [16] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.
- [17] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” 2017.
- [18] S. Ruder, “An overview of gradient descent optimization algorithms,” *arXiv preprint arXiv:1609.04747*, 2016.

Neural Machine Translation Strategies for Generating Honorific-style Korean

Lijie Wang*, Mei Tu*, Mengxia Zhai*, Huadong Wang*, Song Liu*, Sang Ha Kim[†]

* Samsung Research China - Beijing (SRC-B), China

[†] Samsung Research (SR), Seoul, Korean

{lijie.wang, mei.tu}@samsung.com, mengx.zhai@partner.samsung.com

{huadong.wang, s0101.liu, sangha01.kim}@samsung.com

Abstract—Expression with honorifics is an important way of dressing up the language and showing politeness in Korean. For machine translation, generating honorifics is indispensable on the formal occasion when the target language is Korean. However, current Neural Machine Translation (NMT) models ignore generation of honorifics, which causes the limitation of the MT application on business occasion. In order to address the problem, this paper presents two strategies to improve Korean honorific generation ratio: 1) we introduce honorific fusion training (HFT) loss under the minimum risk training framework to guide the model to generate honorifics; 2) we introduce a data labeling (DL) method which tags the training corpus with distinctive labels without any modification to the model structure. Our experimental results show that the proposed two strategies can significantly improve the honorific generation ratio by 34.35% and 45.59%.

Keywords-honorific fusion training; data labeling; korean honorific

I. INTRODUCTION

Neural Machine Translation has made great progress in recent years, the translation results have reached or even exceeded the level of human beings in some fields, especially after combining the LSTM /GRU [1, 2] or self-attention mechanism [3]. In NMT, the encoder encodes the source language into a high level of semantic representation and decoder generates the target translation, the systems comply with this kind of structure and achieve good performance in most cases. However, the generated translation fails to express target language phenomenon sometimes, e.g. honorific-style expressions.

Expressions containing honorifics are used to show respect towards a speaker's audience. It also indicates the level of formality and politeness of a situation. Therefore, honorifics are incredibly important in Korean culture and widely used for conversation and relationship-building, ignorance of using honorifics can be seen as very impolite, it is necessary to use honorifics in order to show respect to people like elders, those in superior positions. The following examples show honorific-style and non-honorific-style expressions when translating from English to Korean:

- Source: “What did you do last weekend?”
- Honorific target:

ji-nan ju-mar-e mu-eos-eul ha-syeot-seum-ni-kka?

지난 주말에 무엇을 하셨습니까?

- Non-honorific target:

ji-nan ju-mar-e mwol haen-ni?

지난 주말에 뭘 했니?

The two targets above are correct and the Korean words with bold are honorifics. The first target is a polite honorific-style expression, while the second target is a non-honorific-style expression that will be not appropriate for a formal occasion. Since the current encoder-decoder structures in NMT cannot capture the honorific information in the source sentence, it is hard to generate the honorific-style target as the final translation according to the context representation of the encoder. From the perspective of training, the reason that the current NMT techniques are difficult to generate honorifics mainly comes from two folds:

- Source sentence and target sentence contain asymmetric information. The current NMT can't distinguish the honorifics from the training data. The honorific and non-honorific sentences are mixed together. One source sentence corresponds to both honorific and non-honorific sentences sometimes. The data consistency is not good, thus making it difficult to learn.

- The translation model itself does not have a reward and punishment mechanism for the honorifics and non-honorifics, which leads to the translation paying no attention to the choice of honorifics.

There're two obvious existing methods to solve the above problems. The first method is to pick out the parallel corpus with honorific-style targets; the second method is to modify the translation results to honorific-style expressions by hard rule. However, the honorific-style data is only about 30% of the whole corpus, which is not enough to achieve good translation performance. While the hard-rule modification will always lead to disfluent translation. On the other hand, a complement rule-database is hard to build, making the second method less practical. Considering the data utilization rate and feasibility, we propose two strategies to generate the honorific-style translation:

- The first strategy is honorific fusion training which introduces honorific fusion training loss into the training loss function. The honorific fusion training loss is integrated under the minimum risk training (MRT) [4] framework and can help the translation model paying more attention to honorifics during training

- The second strategy is to annotate distinctive labels on the training corpus. From the perspective of information symmetry, the source sentence is marked with honorific or non-honorific tag according to target sentence. The encoder can capture the distinctive feature from the source sentence, which contributes to generating honorific or non-

honorific translation based on the addictive tag in source side.

II. RELATED WORK

The encoder-decoder NMT architectures with attention mechanism like GNMT [5], ConvS2S [6], Transformer [3] have achieved state-of-the-art in recent years. However there're still some challenges [7] at the model and data level after excluding decoding speed, previous work has focused on how to optimize the data and model space. Dual learning [8] and unsupervised NMT [9, 10] are proposed to solve the problem of insufficient training data; Deliberation net [11] is proposed to solve decoding problem that the beam search is not globally optimal [12]; MRT [4] and Adversarial-NMT [13] are proposed to update the parameter gradient based on sampling obtained by search. Moreover, the current NMT models are optimized by maximum likelihood estimation to make the translation result more fidelity. However it's uncertain whether the style of translation result is appropriate in some cases, this paper tries to address this issue at both model and data level according to the previous work.

From the perspective of the model, we learn from the text style transfer which is a sequence generation method in order to apply constraints to the NMT loss to guide the generation of results containing expected styles. Text style transfer [14, 15, 16] is a rephrasing task without changing the internal meaning of the context, and generative adversarial networks [17], back-translation [18] and Auto-Encoder [19] have been proposed to yielded state-of-the-art results. Despite the success of NMT and text style transfer, previous work is rarely involved how to accurately translate the source language to another target language that contains the desired style. Our solution is not only to translate the source language correctly, but also make the generated translation results meet the specific style rather than simple rephrasing.

From the perspective of data, the source language doesn't contain the characteristics of the target sentence style. Therefore, we borrow the method of labeling the language direction in one-to-many translation [20], and tag source language with the style label of the corresponding target language before training. After training, the model can learn the style corresponding to different tags.

III. OUR METHOD

In this section, we proposed two methods to improve the honorific generation ratio. The first method introduces honorific fusion training which depends on an honorific classifier to help the NMT model to generate more honorific results. The second method adopts data labeling to add corresponding label at the end of the source Chinese sentence depending on whether the target is an honorific-style result or not before training, then the translation model can learn the honorific mode during training.

A. Honorific Fusion Training

Given parallel corpus X, Y, where $X = \{x_1, x_2, \dots, x_s\}$, $Y = \{y_1, y_2, \dots, y_s\}$ and (x_i, y_i) is the i-th aligned sentence

pair, $M(\theta)$ represents the parameters of NMT model. The training goal is to minimize the total loss function in formula (1) with model parameter $M(\theta)$, θ is the parameter of NMT model:

$$Loss_{Total}(\theta) = \gamma Loss_{MT}(\theta) + (1 - \gamma) Loss_{HE}(\theta) \quad (1)$$

Where $Loss_{MT}$ refers maximum likelihood estimation (MLE) loss, while $Loss_{HE}$ indicates our proposed honorific fusion training loss. γ is a hyper-parameter that controls the trade-off between honorific fusion training loss and MLE loss.

Honorific fusion training loss is defined as expected loss with respect to the posterior distribution in formula (2):

$$\begin{aligned} Loss_{HE}(\theta) &= \sum_{i=1}^S E_{y'_i|x_i;\theta}[Loss_D(y'_i)] \\ &= \sum_{i=1}^S \sum_{y'_i \in y(x_i;\theta)} P(y'_i|x_i;\theta) Loss_D(y'_i) \end{aligned} \quad (2)$$

Where the $Loss_D(y'_i)$ is the honorific loss of the model prediction y'_i , $y(x_i;\theta)$ is the all possible candidate translations for x_i with $M(\theta)$. $P(y'_i|x_i;\theta)$ is the probability of generating y_i given x_i and θ .

The proposed honorific loss is to measure how the model prediction y'_i is not likely to be an honorific-style sentence. In this paper, we use a pre-trained honorific classifier D to compute it with formula (3). D is based on convolutional neural network (CNN). $D(y'_i)$ is the softmax output of classifier. Sentences used for training are labeled by rules in Section III.B. We follow [21] to build the honorific classifier, where the label 1 stands for honorific-style sample and 0 stands for non-honorific-style sample.

$$Loss_D(y'_i) = -\log D(y'_i) \quad (3)$$

In formula (2), the expectations are intractable to calculate due to the exponential search space of $y(x_i;\theta)$. Using a subset of the full search space to approximate the posterior distribution to alleviate this problem in formula (4):

$$Loss_{HE}(\theta) = - \sum_{i=1}^S \sum_{j=1}^K Q(y'_{ij}|x_i;\theta) \log(D(y'_{ij})) \quad (4)$$

Where the j-th sampled sentence is y'_{ij} , K candidate sentences is sampled as a subset of the full search space, $Q(y'_{ij}|x_i;\theta)$ is a distribution defined on the subset in formula (5):

$$Q(y'_{ij}|x_i;\theta) = \frac{P(y'_{ij}|x_i;\theta)}{\sum_{j=1}^K P(y'_{ij}|x_i;\theta)} \quad (5)$$

The overall framework of our honorific fusion training is shown in Figure 1. The honorific fusion training contains MLE and honorific fusion training loss as follows:

- 1) Input sentence (x_i, y_i) into NMT to get probability matrix of prediction and evaluate the gap between reference and predicts, then get $Loss_{MT}(\theta)$ by cross entropy.

- 2) Generate K candidate sentences $\{y'_{i1}, y'_{i2}, \dots, y'_{iK}\}$ by polynomial sampling with playback K times at each decoding step according to the output distribution [22].
- 3) Compute the approximating posterior probability of the sampled K sentences $Q(y'_{ij}|x_i; \theta)$ ($1 \leq j \leq K$) according to formula (5).
- 4) Compute honorific loss $Loss_D(y'_i)$ according to formula (3).
- 5) The total loss of sentence pair (x_i, y_i) is as follows:

$$Loss_{Total}(\theta) = \gamma Loss_{MT}(\theta) - (1 - \gamma) \sum_{j=1}^K Q(y'_{ij}|x_i; \theta) \log D(y'_{ij}) \quad (6)$$

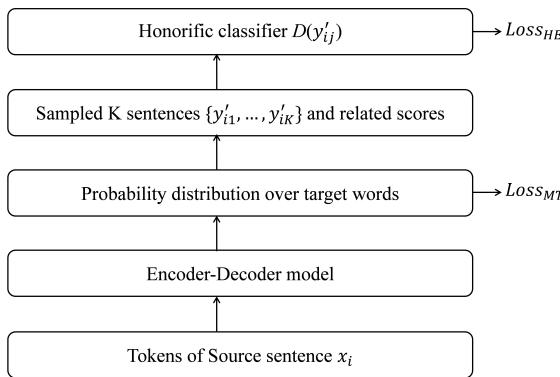


Figure 1. The honorific fusion training framework

The Algorithm 1 shows the full details of the proposed honorific fusion training. In this paper, the NMT model is pre-trained with only MLE loss firstly. The model is initialized with the pre-trained NMT parameters instead of training from scratch in honorific fusion training. The classifier D is also pre-trained upon the honorific and non-honorific corpus. The classifier is built following [21]. It is based on CNN with multiple convolution kernels. The classification performance on our test set will be reported in the experimental part. The honorific and non-honorific corpus are obtained from the target side of the parallel training corpus. Hard rules are applied to determine whether a sentence is an honorific-style sentence or not, which will be described in Session III.B. The hyper-parameters γ and K will be discussed in the experimental part. Based on pre-trained NMT model and classifier D, the honorific fusion training won't take too much training time in the fine-tune stage, and decoding time remains the same.

B. Data Labeling

Data labeling is used to automatically distinguish the target language in a one-to-many translation task [20], the target direction is labeled at the end/start of the source sentence to indicate the translation direction. The second strategy is to add distinctive labels on the training data

Algorithm 1 Honorific fusion training.

Input: Chinese-Korean parallel sentence: x, y and the target sentence length is T, K is maximum sampling times, γ is the hyper-parameter controlling trade-off for loss.
Output: $M(\theta)$

- 1: Load pre-trained NMT model $M(\theta)$, honorific classifier D
- 2: $i \leftarrow 1$
- 3: **while** $i \leq T$ **do**
- 4: $j \leftarrow 1$
- 5: **while** $j \leq K$ **do**
- 6: Sampling $y'_{ij} \sim P(y'_i|x, y_{<i}; \theta)$, $P_{ij} = P(y'_{ij}|x, y_{<i}; \theta)$
- 7: **end while**
- 8: **end while**
- 9: $P_j = \prod_{i=1}^T P_{ij}$
- 10: $Q(y'_j|x; \theta) = \frac{P_j}{\sum_{j=1}^K P_j}$ //score of j-th sampled result
- 11: $Loss_{HE}(\theta) = -\sum_{j=1}^K Q(y'_j|x; \theta) \times \log D(y'_j)$ //honorific fusion training loss
- 12: $Loss_{MT}(\theta) = CE(y, y'_i)$ //the cross-entropy loss
- 13: $Loss_{Total}(\theta) = \gamma Loss_{MT}(\theta) + (1 - \gamma) Loss_{HE}(\theta)$
- 14: Update NMT model by gradient $\nabla_{\theta} Loss_{Total}(\theta)$

to generate honorific-style translation. The data labeling method can capture honorific feature from the target sentence by tagging the labels at the end of source sentence. Label <p> or <np> is added at the end of the source sentence according to the aligned target sentence. If it is an honorific-style sentence, the corresponding label <p> is added at the end of source sentence, or else <np> is added. We did nothing else to the model except for the data labeling. When testing, we add label <p> at the end of every input sentence to generate honorific-style expression. In addition, the training time and decoding time won't increase too much with just a simple label added at the end of every sentences.

Table I
SPEECH LEVELS IN KOREAN

Name	Politeness	Formality
Hasipsio-che (하십시오체)	High	High
Haeyo-che (해요체)	High	Low
Hao-che (하오체)	Neutral	High
Hage-che (하게체)	Neutral	Neutral
Haera-che (해라체)	Low	High
Hae-che (해체)	Low	Low

The criteria above determine what are the honorifics. According to the Korean grammar¹ published by the National Institute of Korean Language, the Korean language experts summarize six speech levels based on the level of politeness and formality in the Korean dialogue as shown in Table I. Each level has its own unique set of verb endings which are used to indicate the level of formality and politeness.

¹<http://www.ilovekorean.net/files/pdf/korgrammar.pdf>

Korean language experts classify the Korean sentence under high level of politeness into an honorific-style expression. As shown in Table I, both Hasipsio-che (another name is hapsyo-che or '합쇼체') and Haeyo-che meet the requirement of honorific. Referring to related paper [23], the verb endings of the above two honorific types are summarized as shown in Table II.

Table II
VERB ENDING OF TWO HONORIFIC TYPES

Type	Verb Ending with Hasipsio-che	Verb Ending with Haeyo-che
Declarative	-ㅂ니다	-아요/어요
Interrogative	-ㅂ니까	-아요/어요
Imperative	-(으)십시오/-으시지요	-아요/어요
Propositive	-(으)십시오	-아요/어요

IV. EXPERIMENTS

A. Datasets and Metrics

The paper focuses on improving honorific generation ratio which is defined as the number of sentences containing honorifics divided by the total number of sentences. Our translation task is conducted on Chinese into Korean direction. Our baseline and honorific fusion training experiments make use of an MT training set of about 7.2M Chinese-Korean parallel sentences, including 3.9M purchased data and 3.3M script crawled data. Before training, Chinese have been segmented and processed into subword units using byte-pair encoding (BPE) [24], eventually the source Chinese vocabulary size is about 4.4k. For Korean, tokenizer² in Moses and BPE are used to obtain the target Korean vocabulary and the size is about 4.2K. The randomly extracting 2,133 Chinese sentences are translated into Korean 4 references by crowdsourcing as the Normal Test Set (NTS), then we extract the parallel sentences in which the references containing honorifics in any of them are about 1,474, we set it as the Honorific Test Set (HTS). BLEU [25] score and honorific generation ratio are adopted when evaluating.

When training honorific classifier, 1M source Chinese sentences are randomly selected, the pre-trained and best performance NMT model is used to translate the selected Chinese into Korean and we get best 5 translation results of every sentences. Based on the classification criteria of honorific in section III.B, the Korean are classified to honorific-style sentences and non-honorific-style sentences with label 1 and 0. Finally, 5M labeled Korean sentences with 30% of them contain honorifics are obtained to train the honorific classifier.

When training NMT model with data labeling, based on the classification criteria of honorific in section III.B, the 7.2M processed Chinese-Korean parallel sentences are labeled at the end of corresponding source Chinese sentences according to the raw Korean sentences. According to statistics, honorific sentences account for about 30%.

²<https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl>

B. Setup

Transformer base model [3] is used as the baseline model. We use 2 Tesla P40 GPUs to pre-train the NMT model for honorific fusion training. The NMT experiments are carried out based on the OpenNMT-py [26] and the honorific classifier³ is based on [21]. We use Adam [27] with $\beta_1 = 0.9$, $\beta_2 = 0.998$ and $\epsilon = 10^{-9}$ when training NMT. We vary learning rate with *warmup_steps* 4000 over the course of training.

C. Experiment Results

As shown in Table III, the accuracy of honorific classifier is 99.99% on a test set of 7k, the BLUE of honorific fusion training model ($\gamma=0.7$, K=50), and data labeling model in NTS are a little bit lower than baseline model, although the honorific generation ratio is much higher than baseline model. The reason that the BLEU of proposed two methods are a little bit lower than baseline model, we think it's caused by the references. Since the references are not for honorific expressions, and it's unfair to compare honorific-style results with NTS, then we test in HTS, the results show both honorific fusion training and data labeling method can not only improve BLEU, but also improve the honorific generation ration by 34.35% and 45.59% respectively . In order to compare the effectiveness of the proposed methods a step further, we use manual evaluation to further analyze the results in Table IV.

Table III
COMPARISON OF HONORIFIC FUSION TRAINING AND DATA LABELING

Test Set	Model	BLEU	HG Ratio
NTS	Baseline	34.21	44.82%
	HFT	33.57	79.23%
	DL	34.17	99.40%
HTS	Baseline	35.53	54.21%
	HFT	36.04	88.74%
	DL	37.20	99.80%

NTS represents Normal Test Set; HTS represents Honorific Test Set; baseline represents Transformer baseline model; HFT represents the honorific fusion training model; DL represents the data labeling model; HG Ratio represents honorific generation ratio.

D. The Human Evaluation

In this paper, we random take 300 Chinese sentences and their corresponding Korean results. We adopt Mean opinion score (MOS) as subjective evaluation from "excellent" to "bad" with range 5 to 1. In order to consider the correctness of the translation and the honorific-style expression, when the result Korean sentence is not an honorific-style expression, the sentence score will be subtracted by 0.2 points; when the result honorific-style Korean sentence doesn't conform to correct grammars, the sentence score will be subtracted by 1. The final average MOS is shown in Table IV, the MOS of honorific fusion training and data labeling method exceed the baseline model, and honorific-style results also increase greatly.

³<https://github.com/junwang4/CNN-sentence-classification-pytorch-2018>

Table IV
HUMAN EVALUATION

Model	MOS	HG Ratio
Baseline	4.41	48.67%
HFT	4.43	81.00%
DL	4.48	95.00%

During manual evaluation, we find the two proposed methods have their own characteristics. Honorific fusion training model will miss honorific verb endings in few cases to ensure the accuracy of the results, while DL will ensure the generation of honorifics to the greatest extent. There're some positive and negative impact on final results when using method DL. In few cases, the DL model will add respectful title for the character although we don't deliberately make the model to learn such honorific features, but in few cases the DL model will add some suffix '요 (yo)' to represent the honorific feature for the translation of single word, for example when translation '1.5 欧元 (ou yuan)', the result will be '1.5유로요 (yuro- yo)' , however it's grammatically wrong. We think the phrase translation ability of DL method in some cases should be further improved.

It is concluded that the result of data labeling model surpasses honorific fusion training model. Data labeling method not only has good effect but also has a simpler process without modifying the NMT model; the honorific fusion training provides a way to generate a specific style of the translation. We notice the limitation that the current methods we proposed can only generate honorific-style sentence endings. Next, we will further try to generate honorific words in the middle of sentences including nouns and pronouns, and try to verify the effectiveness of our methods in honorific expression in other languages like Japanese.

E. The Effect of Hyperparameters to HFT

In honorific fusion training experiments, the accuracy of honorific classifier is 99.99% on a test set of 7k. We adjust the hyper-parameter γ which is the parameter that controls trade-off between MLE and honorific fusion training loss, we evaluate in NTS. In the process, maximum sampling times K remaining unchanged and results are shown in Table V ,and the test set is NTS. The BLEU score continues to increase from 32.07 to 34.17 when K = 10 and γ increases from 0.1 to 0.9 at interval 0.2 (When $\gamma = 1$, the model turns to the standard Transformer baseline model), but the honorific generation ratio decreases from 84.76% to 54.95%. The increase of honorific generation ratio accompanied by the decrease of BLEU, BLEU decreases slightly but honorific generation ratio increases by 24.75% when $\gamma = 0.7$; in order to raise the honorific generation ration as much as possible without lowering the BLEU too much, then γ is set to 0.7 and change maximum sampling times K to observe the influence of hyper-parameter K.

In Table VI, when K increase from 1 to 400 and $\gamma = 0.7$, BLEU decreases from 34.30 to 33.35, and honorific

Table V
ADJUSTMENT OF HYPER-PARAMETER γ

Model	BLEU	HG Ratio
Baseline	34.21	44.82%
HFT ($\gamma=0.1$, K=10)	32.07	84.76%
HFT ($\gamma=0.3$, K=10)	32.93	81.58%
HFT ($\gamma=0.5$, K=10)	33.23	79.18%
HFT ($\gamma=0.7$, K=10)	33.62	69.57%
HFT ($\gamma=0.9$, K=10)	34.17	54.95%

generation ratio increases to 79.23% when K is set to 50, the honorific generation ratio increases by 34.41%, then both BLEU and honorific generation ratio will be basically stable.

Table VI
ADJUSTMENT OF HYPER-PARAMETER K

Model	BLEU	HG Ratio
Baseline	34.21	44.82%
HFT ($\gamma=0.7$, K=1)	34.30	44.49%
HFT ($\gamma=0.7$, K=5)	33.82	64.56%
HFT ($\gamma=0.7$, K=10)	33.62	69.57%
HFT ($\gamma=0.7$, K=50)	33.57	79.23%
HFT ($\gamma=0.7$, K=100)	33.45	77.59%
HFT ($\gamma=0.7$, K=200)	33.36	78.11%
HFT ($\gamma=0.7$, K=400)	33.35	78.06%

As shown in Table V and Table VI, when hyper-parameter K is fixed, if γ becomes bigger, the proportion of MLE loss becomes larger, the BLEU will be higher and the honorific generation ratio will be lower because the model is more likely to notice the word associated with NTS reference and pays less attention to generate honorifics; when γ is fixed as the sample size K become bigger, the honorific generation ratio will be higher, but the BLEU will be a little bit lower in NTS but higher in HTS as shown in table III.

V. CONCLUSION

In this paper, we have introduced two training strategies to improve the honorific generation ratio in Chinese-Korean translation from the perspective of model and data respectively. The first strategy adopts honorific fusion training loss under MRT framework to guide NMT. The second strategy adopts data labeling which will label the source sentences with corresponding honorific labels of the target sentences. Experiments show both strategies can increase honorific generation ratio significantly.

REFERENCES

- [1] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation. 9(8), 1735–1780 (1997)
- [2] Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
- [3] Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems (2017)

- [4] Shen, S., Cheng, Y., He, Z., He, W., Hua, W., Sun, M., Liu, Y.: Minimum risk training for neural machine translation. In: Proceedings of ACL 2015, pp. 1683–1692 (2015)
- [5] Wu, Y., Schuster, M., Chen, Z., et al.: Google’s neural machine translation system: bridging the gap between human and machine translation. arXiv preprint arXiv:16008144 (2016)
- [6] Gehring, J., Auli, M., Grangier, D., et al.: Convolutional sequence to sequence learning. arXiv preprint arXiv:17003122 (2017)
- [7] Koehn, P., Knowles, R.: Six challenges for neural machine translation. arXiv preprint arXiv:17003872 (2017)
- [8] He, D., Xia, Y., Qin, T., Wang, L., Yu, N., Liu, T., Ma, W.: Dual learning for machine translation. In: Proceedings of NIPS 2016 (2016)
- [9] Artetxe, M., Labaka, G., Agirre, E., et al.: Unsupervised neural machine translation. arXiv preprint arXiv:17111041 (2017)
- [10] Lample, G., Ott, M., Conneau, A., Denoyer, L., Ranzato, M.: Phrase-based & neural unsupervised machine translation. arXiv preprint arXiv:1804.07755 (2018)
- [11] Xia, Y., Tian, F., Wu, L., Lin, J., Qin, T., Yu, N., Liu, T.: Deliberation networks: Sequence generation beyond one-pass decoding. In Proc. of NIPS (2017)
- [12] Freitag, M., Al-Onaizan, Y.: Beam search strategies for neural machine translation. CoRR, abs/1702.01806 (2017)
- [13] Wu, L., Xia, Y., Zhao, L., Tian, F., Qin, T., Lai, J., Liu, T.: Adversarial neural machine translation. arXiv preprint arXiv:1704.06933 (2017)
- [14] Prabhumoye, S., Tsvetkov, Y., Salakhutdinov, R., Black, W.: Style Transfer Through Back-Translation. In Proceedings of ACL (2018)
- [15] Jhamtani, H., Gangal, V., Hovy, E., Nyberg, E.: Shakespeareanizing modern language using copyenriched sequence-to-sequence models. arXiv preprint arXiv:1707.01161 (2017)
- [16] Shen, T., Lei, T., Barzilay, R., Jaakkola, T.: Style transfer from non-parallel text by cross-alignment. In: Advances in Neural Information Processing Systems. (2017)
- [17] Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al.: Generative adversarial nets. In NIPS (2014)
- [18] Edunov, S., Ott, M., Auli, M., Grangier, D.: Understanding back-translation at scale. arXiv preprint arXiv:1808.09381 (2018)
- [19] Hinton, G., and Salakhutdinov, R.: Reducing the dimensionality of data with neural networks. Science, 313 (5786), 504–507 (2006)
- [20] Wang, Y., Zhang, J., Zhai, F., Xu, J., Zong, C.: Three Strategies to Improve One-to-Many Multilingual Translation , Association for Computational Linguistics, (2018)
- [21] Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
- [22] Chatterjee, S., Cancedda, N.: Minimum error rate training by sampling the translation lattice. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 606–615. Association for Computational Linguistics (2010)
- [23] Yin, F.: A contrastive Study on the honorific forms in Korean and Chinese, Jilin University (2016)
- [24] Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, pp. 1715–1725 (2016)
- [25] Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 311–3 Association for Computational Linguistics (2002)
- [26] Klein, G., Kim, Y., Deng, Y., Senellart, J., Rush, A.: OpenNMT: Open-Source Toolkit for Neural Machine Translation. In Proceedings of ACL (2017)
- [27] Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1406.4297 (2014)

Ranking Like Human: Global-View Matching via Reinforcement Learning for Answer Selection

Yingxue Zhang*, Ping Jian*†, Ruiying Geng*, Yuansheng Song* and Fandong Meng‡

*Beijing Institute of Technology, China

Email: {zhangyingxue, pjian, rygeng, yssong}@bit.edu.cn

†Beijing Engineering Research Center of High Volume Language Information Processing
and Cloud Computing Applications, China

‡ Pattern Recognition Center, WeChat AI, Tencent Inc, China

Email: fandongmeng@tencent.com

Abstract—Answer Selection (AS) is of great importance for open-domain Question Answering (QA). Previous approaches typically model each pair of the question and the candidate answers independently. However, when selecting correct answers from the candidate set, the question is usually too brief to provide enough matching information for the right decision. In this paper, we propose a reinforcement learning framework that utilizes the rich overlapping information among answer candidates to help judge the correctness of each candidate. In particular, we design a policy network, whose state aggregates both the question-candidate matching information and the candidate-candidate matching information through a global-view encoder. Experiments on the benchmark of WikiQA and SQuAD demonstrate that our RL framework substantially improves the ranking performance.

Keywords-Answer Selection; Reinforcement Learning;

I. INTRODUCTION

Answer Selection (AS) is an important component of open-domain Question Answering (QA). Given a question and a set of candidate answers, the objective is to select all the correct answers from the candidate set. It is also a ranking task, whose objective is to rank the correct answers in front of incorrect ones. Pre-deep learning methods address this task through modeling the question-candidate similarity (Q-C similarity), including syntactic or semantic features based methods [1–3], and structural kernels based methods [4, 5]. Recently, neural networks have shown outstanding performance on this task including attentive networks [6, 7] and compare-aggregate networks [8, 9] and so on.

These previous works typically model each pair of the question and the answer candidate independently. However, when modeled together with candidates, the question is usually too brief to provide adequate matching information, which makes it difficult to make the right decision. While in practice, a question usually corresponds to several correct candidates. These correct candidates carrying essential information are always similar to each other and easily distinguished from incorrect answers. Imagine when people encounter a complicated candidate answer, if they fail to extract enough meaningful information from the question for a safe decision, they will try to get strong hints from other easily identified answers.

Consider the question and its answer candidates given in Table I. The phrase “warehouse” in candidate C_1 also appears in the question, making C_1 an easily identified correct answer. While, other candidates have little matching information with the question, making it difficult to judge their correctness solely with the evidence provided by the question. Obviously, there is some overlapping information between C_1 and C_2/C_3 , such as the “*operational*” of C_1 and C_2 , the “*data*” of C_1 and C_3 , while the incorrect answer C_4 is quite different from other answer candidates. Therefore, a comparison with C_1 can help distinguish other answer candidates. We observe that, when ranking the current candidate, comparing it with other candidates as humans always do can help make more accurate ranking decisions.

However, comparing the current candidate with all other candidates suffers from much noise from the incorrect answers in the set, making it challenging to leverage these overlapping information to improve ranking performance. To address the above problems, we construct a reinforcement learning (RL) framework [10] to model AS as a sequential ranking problem, where the RL agent compares the current candidate with the most reliable candidate predicted previously. In this way, we can utilize effective matching information between candidates while reducing much effect from noise of incorrect answers (We use “the highest-confidence candidate” to represent most reliable candidate predicted previously). On top of basic sentence modeling neural networks, two innovations are introduced in our framework. First, we propose a global-view comparison for answer candidates. In particular, for each answer candidate, we find the highest-confidence candidate from its previous candidates (which have been ranked in earlier time steps), then the system encodes the current candidate with both the question and the highest-confidence candidate as the global matching information. Second, we design a policy network for ranking incrementally, whose state is the global matching information mentioned above. The policy network decides whether the current candidate is correct (action) at each state, then obtains a reward for guiding the learning. The agent gets a positive reward only when the current decision improves the overall performance of ranking. Our RL framework can be easily applied to various basic neural networks,

Table I
A REAL CASE IN WIKIQA.

Question		Candidate	Label
what are warehouse spreadsheets used for?	C_1	The data stored in the warehouse are uploaded from the operational systems .(such as marketing , sales etc. , shown in the figure to the right)	True
	C_2	The data may pass through an operational data store for additional operations before they are used in the DW for reporting.	True
	C_3	The access layer helps users retrieve data .	True
	C_4	The combination of facts and dimensions is sometimes called a star schema.	False

producing more accurate results than models that merely consider the current question-candidate pair and make local decisions.

Our major contributions are three-fold.

- We propose a RL framework for Answer Selection to utilize the matching information among candidates, which can be easily applied on various sentence modeling neural networks.
- The proposed RL framework allows models to extract global matching information from both the question-candidate pair and the highest-confidence candidate identified in earlier states through the global-view encoder;
- We apply the proposed RL framework to several sentence modeling neural networks. Experimental results show that it gains significant and consistent improvements on the benchmarks of WikiQA and SQuAD.

II. RELATED WORK

A. Answer Selection

Answer Selection (AS) has enjoyed wide popularity in natural language processing. Most previous methods typically model the AS as a text matching problem, including conventional feature engineering based approaches [1–3] and deep learning models [6, 7]. Feature engineering based methods tend to design intricate linguistic features, such as syntactic features extracted from dependency and constituency tree [3], or semantic features extracted from WordNet [11], which is time-consuming and inefficient. Most recent AS systems are based on deep neural networks, which have earned a much promising break. These DNN based models can be roughly divided into two categories, the attention-based models [12, 13] and the compare-aggregate based models [8, 14]. Some recent studies focus on employing external resource to improve the ranking performance. [15] propose the EviNets that extracts relevant evidence from external text corpora by a search system and retrieves KB triples based on entities tagged by a entity linking system before each prediction, which improves the performance of answer selection but is also time-consuming. [16] propose a kernel-based method that utilizes external resources such as constituency/dependency tree and semantic features from WordNet to model the similarity between question-candidate pairs, which requires intricate feature engineering work. In this paper, we propose a reinforcement

learning algorithm where we assign the highest-confidence candidate and we utilize it to support answer selection. Compared with previous methods, our model needs neither intricate feature engineering work nor external resources.

B. Reinforcement Learning

Reinforcement Learning (RL) [17] has shown promising results in many natural language processing tasks [18]. [19] regard the pronoun resolution as a sequence decision problem. They argued that modeling useful information of preceding potential antecedents is crucial for classifying later zero pronoun-candidate antecedent pairs and used policy gradient to optimize the task. [20] explore a deep reinforcement learning strategy to generate the false-positive indicator for distant supervision relation extraction. [21] use reinforcement learning to optimize a new objective function with a reward defined by the property of the NLI datasets to make full use of the discourse information. In this work, we propose a RL framework for Answer Selection, which makes full use of the answer candidates to select correct answers. Our RL framework allows models to extract global matching information from both the Q-A pair and the answer candidates identified in earlier states through the global-view comparison mechanism.

III. MODEL

As figure 1 shows, the proposed framework consists of two components: (1) The global-view encoder, which contains two matching modules, the question-candidate matching module and the candidate-candidate matching module. We concatenated these two views of matching information as the RL state and feed it into the RL agent.(2) The RL agent, which follows a policy to decide which action is chosen at each state and obtains a reward for policy learning.

A. Global-View Encoder

The highest-confidence candidate We first define the highest-confidence candidate which is used to help identify the correctness of the current candidate. Suppose we are predicting the t -th candidate C_t , the highest-confidence candidate is the candidate which is most likely to be positive in $\{C_1, C_2, \dots, C_{t-1}\}$ predicted in earlier time steps. Since the first candidate C_1 has no antecedents, we use the question to initialize the highest-confidence candidate. After obtaining the highest-confidence candidate, we encode two views of matching information with two matching modules.

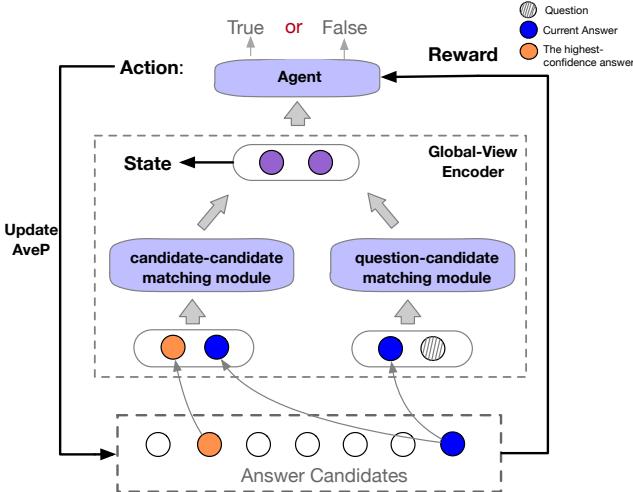


Figure 1. The architecture of proposed RL framework. The global-view encoder consists of two matching modules: the question-candidate matching module and the candidate-candidate matching module.

The matching Module The question-candidate matching module and the candidate-candidate matching module are two separate matching modules with the same network architecture. Each one has its own parameters. We apply the compare-aggregate network proposed by [9] as the matching module, which is effective for a wide range of sequence matching problems. We briefly introduce its components:

(1) The embedding module. We initialize the question Q as $\{\mathbf{x}_{q_1}, \dots, \mathbf{x}_{q_{|Q|}}\}$ and initialize the current candidate X_C as $\{\mathbf{x}_{c_1}, \mathbf{x}_{c_2}, \dots, \mathbf{x}_{c_{|C|}}\}$. Here, $\mathbf{x}_{q_1}, \mathbf{x}_{q_2}, \dots, \mathbf{x}_{q_{|Q|}}$ and $\mathbf{x}_{c_1}, \mathbf{x}_{c_2}, \dots, \mathbf{x}_{c_{|C|}}$ are pre-trained word embeddings. $|Q|$ and $|C|$ respectively represent the numbers of words in the question and the candidate.

(2) The attention module. It generates a question-aware candidate representation $H = \{h_{c_1}, h_{c_2}, \dots, h_{c_{|C|}}\}$ through the attention mechanism [22], where each h_c is an attention-weighted sum of the vectors of Q :

$$\alpha_{ij} = \mathbf{x}_{c_i} \cdot \mathbf{x}_{q_j} \quad (1)$$

$$p_{ij} = \frac{e^{\alpha_{ij}}}{\sum_{j=1}^{|Q|} e^{\alpha_{ij}}} \quad (2)$$

$$h_{c_i} = \sum_{j=1}^{|Q|} p_{ij} \mathbf{x}_{q_j} \quad (3)$$

where \cdot represents dot product, and h_{c_i} represents the attention representation of the i -th word of the candidate answer.

(3) The comparison module. In this module, each pair of \mathbf{x}_{c_i} and h_{c_i} are combined into a vector \mathbf{v}_{c_i} through a comparison function \mathbb{F} , which can be “element-wise multiplication” and “element-wise subtraction”. In this paper, we use “element-wise multiplication” as the comparison function:

$$\mathbf{v}_{c_i} = \mathbf{x}_{c_i} \odot h_{c_i} \quad (4)$$

where the \odot represents the element-wise multiplication.

(4) The aggregate module. The vectors $\mathbf{V} = \{\mathbf{v}_{c_1}, \mathbf{v}_{c_2}, \dots, \mathbf{v}_{c_{|C|}}\}$ obtained after comparison are aggregated into a final vector \mathbf{M} for the classification. The aggregate module can be various networks. In this paper, we apply the Bidirectional Gated Recurrent Unit network (BiGRU) [23] to the aggregate module. We feed \mathbf{V} into the BiGRU and concatenate the last hidden states of the BiGRU in two directions to obtain the final vector M .

Global-View Matching The global-view encoder extracts adequate matching information through global-view comparison. It consists of two parts: the Q-C CompAgg module that compares the current candidate C_{cur} with the question, and the C-C CompAgg module compares C_{cur} with C_{pre} , where C_{pre} represents the highest-confidence candidate. When dealing with a specific candidate, we obtain the matching information as:

$$M_{qc} = \text{CompAgg}(Q, C_{cur}) \quad (5)$$

$$M_{cc} = \text{CompAgg}(C_{pre}, C_{cur}) \quad (6)$$

These two matching information are combined as follows and sent to the RL agent.

$$s_t = \text{Concat}(M_{qc}, M_{cc}) \quad (7)$$

B. Policy Network for Answer Selection

From a global perspective of ranking, we design a policy network which can leverage the information of reliable candidates identified in the earlier state.

The policy network adopts a stochastic policy $\pi(a_t | s_t; \Theta)$, where s_t and a_t represent the state and the action at time t respectively. Here, we briefly introduce the state, the action, and the reward used to guide the policy learning at time t .

State The state is generated as Eq.(7), which combines the current Q-C matching information and the previous reliable candidate information.

Action The action space is defined as $\{pos, neg\}$, where the “pos” indicates the current candidate is correct while the “neg” indicates it is incorrect. The degree of “pos”, i.e., $p(a_t = pos | s_t)$ corresponds to the confidence of the current candidate, which is used to determine the ranking position of the current candidate. If the current candidate reaches the highest confidence so far, it will replace the previous highest-confidence candidate C_{pre} and become a new C_{pre} for later time steps. Specifically, the agent makes an action based on the current state through an MLP.

$$h_1(s_t) = \tanh(W_1 s_t + b_1) \quad (8)$$

$$p(a_t | s_t) = \text{softmax}(W_2 h_1(s_t) + b_2) \quad (9)$$

Rewards After sampling an action with the probability at each state, the agent obtains a reward for guiding the policy learning. Since the task is modeled as a sequential ranking problem, we design the reward as the difference of the Average Precision(AveP) before and after each action. We use AP_t to represent the AveP at time step t , then the function of the reward $R(a_{1:T})$ is as follows:

$$R(a_t) = \begin{cases} 0 & t = 1 \\ AP_t - AP_{t-1} & 1 < t \leq T \end{cases} \quad (10)$$

Table II
THE STATISTICS OF WIKIQA AND SELQA.

	WikiQA		SelQA	
	#Q	#QA	#Q	#QA
Train	873	8,627	5,529	66,438
Dev	126	1,130	785	9,377
Test	243	2,351	1,590	19,435

where,

$$AP_t = \frac{1}{N} \sum_{n=1}^N \frac{n}{index(n)} \quad (11)$$

Here, N represents the numbers of correct answers in the first t candidates. “ $index(n)$ ” stands for the ranking position of the n -th correct answer.

Objective Function We optimize the parameters of the policy network using REINFORCE algorithm[17], which aims to maximize the expected reward:

$$\begin{aligned} J(\theta) &= \mathbb{E}_{a_{1:T} \sim p_\theta(a_t|Q, C_{cur}, C_{pre})} R(a_{1:T}) \\ &= \sum_t \sum_a p_\theta(a_t|Q, C_{cur}, C_{pre}) R(a_t) \end{aligned} \quad (12)$$

Hence, we update the policy network with the following gradient:

$$\nabla_\theta J(\theta) = \nabla_\theta \sum_{t=1}^T R(a_t) \log p_\theta(a_t|Q, C_{cur}, C_{pre}) \quad (13)$$

IV. EXPERIMENTS

A. Datasets and Evaluation Metric

We conduct experiments on two datasets: SelQA and WikiQA.

WikiQA Dataset The WikiQA dataset [24] is an open domain question-answering dataset constructed from real queries of Bing and Wikipedia. We remove all questions with no correct answers as predecessors [9, 24] do. The distribution statistics of the question and candidates are shown in table II.

SelQA Dataset The SelQA dataset [25] consists of questions generated through crowd-sourcing and sentence length answers that are drawn from the ten most prevalent topics in the English Wikipedia. SelQA is much larger than WikiQA. As table II shows, the numbers of its Q-A pairs is about nine times of WikiQA’s.

Evaluation Metric We evaluate models by Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR), which are commonly used metrics in this task.

B. Experimental Setup

We introduce experimental setups including baselines and implementation details in this section. We set up the following baselines and comparison systems to show the effectiveness of our method.

Shen et al. 2017 [26] propose an adaptive convolutional filter generation framework for question-answer sentence pairs modeling, by leveraging a meta network to generate input-aware filters.

Table III
RESULTS ON WIKIQA AND SELQA.

Model	WikiQA		SelQA	
	MAP	MRR	MAP	MRR
Shen et al.v2017 [26]	71.07	73.04	89.14	89.83
Tay et al. 2018 [27]	71.20	72.70	-	-
Nicosia et al. 2018 [28]	72.24	73.91	-	-
Shao et al. 2019 [29]	69.41	70.77	-	-
our model	73.39	74.45	90.39	90.95

Tay et al. 2018 [27] propose a novel deep learning architecture, HyperQA, for fast and efficient question-answer ranking and retrieval.

Nicosia et al. 2018 [28] take advantage of small amounts of labelled data that model semantic phenomena in text to encode matching features directly in the word representations.

Shao et al. 2019 [29] design a Transformer-based neural network for answer selection, where they deploy a bidirectional long short-term memory (BiLSTM) behind the Transformer to acquire both global information and sequential features in the question or answer sentence.

To further prove the effectiveness of our RL framework, we also build some matching neural networks to do a fully comparison:

ABCNN: An attention-based CNN proposed by [7].

CompAgg-GRU: The CompAgg-GRU is the matching neural network described in SectionIII-A, which is used as the matching module in our framework.

CompAgg-CNN: CompAgg-CNN is also based on the compare-aggregate framework described in SectionIII-A. But, we replace the BiGRU based aggregation module with CNN based one.

We conduct two experimental settings for these matching models to see whether they achieve higher performance after applying our RL framework. One experimental setting is using these models to simply encode the current question-candidate pair independently for answer selection and trained by supervised learning. The other is putting these models into our RL framework. The two settings use the same set of hyper-parameters. The comparison results are shown in Table III.

Implementation Details We initialize word embeddings with 300-dimensional-GloVe vectors [30]. The mini-batch contains Q-A pairs of 10 different questions. We set the hidden size of GRU to 128, the kernel window sizes of CNN in CompAgg-CNN to [1, 2, 3, 4, 5] with 150 hidden units, the learning rate to $1e^{-4}$ which decayed after every epoch by a factor of 0.95. All these hyper-parameters are optimized on the development set.

C. Main Results and Discussion

Table III shows the results on the benchmarks of WikiQA and SelQA. Our method achieves comparable performance to the recent powerful answer selection model. Compared with the baselines in Table III, we achieve 1.15% improvements of MAP and 0.54% improvements of MRR on WikiQA, and also achieves 1.25% improvements of MAP and 1.12% improvements of MRR on SelQA.

Table IV
COMPARISON OF MATCHING MODELS WITH AND WITHOUT OUR RL FRAMEWORK

Model	WikiQA		SelQA	
	MAP	MRR	MAP	MRR
ABCNN	68.71	70.28	82.14	82.93
ABCNN+RL	69.63	71.02	83.08	83.71
CompAgg-CNN	70.24	71.47	86.48	86.90
CompAgg-CNN+RL	72.17	73.49	87.21	87.76
CompAgg-GRU	71.28	72.59	89.90	90.40
CompAgg-GRU+RL (our model)	73.39	74.45	90.39	90.95

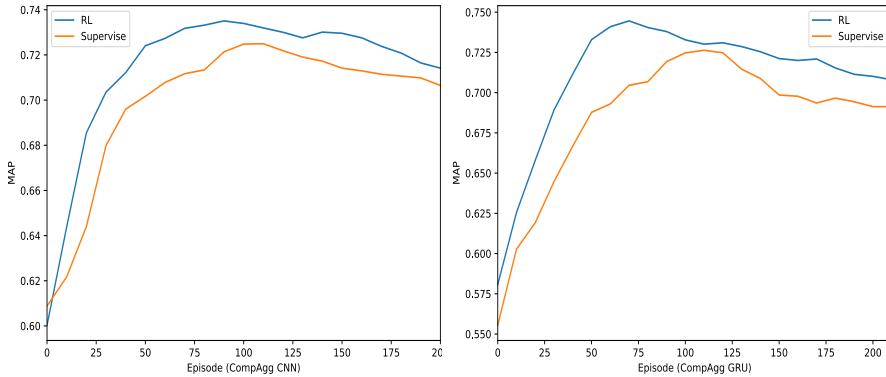


Figure 2. Comparison of training speed on development set of WikiQA in two experiment settings.

Our method not only utilizes the information of question, but also utilize the overlapping information between candidates, which fills the gap between the question and candidates. This is the main reason why our method outperforms them.

Table IV shows the results of models with and without our RL framework. We can see that the proposed RL framework enhances all the baselines both on the small-scale corpus WikiQA and the large-scale corpus SelQA. Under the same corpus, our RL framework gains stable improvement on different models, which proves its generalization.

To further confirm the validity of our model, we also visualize the training speed with and without the proposed RL framework for CompAgg-CNN and CompAgg-GRU under the same hyper-parameters set. Figure 2 shows the comparison of training speed on the development set of WikiQA. Models enhanced by the RL framework achieve better performance in less time. We believe that the global matching information, which is obtained by referring to the highest-confidence candidate ranked previously, fills the gap between the question and current candidate and makes training faster and easier.

V. CONCLUSION

We propose a reinforcement learning framework to model Answer Selection as a sequential ranking problem, which can refer to the high-confidence answer ranked previously and extract global matching information for better decisions. Experiments on several representation models show the effectiveness of our RL framework. For future work, we'd like to explore a more powerful way

to better fuse the global matching information to improve the effectiveness of our RL framework.

ACKNOWLEDGMENT

The authors would like to thank the organizers of IALP 2019 and the reviewers for their helpful suggestions. This research work is supported by the National Key Research and Development Program of China under Grant No.2017YFB1002103, the National Natural Science Foundation of China under Grant No.61751201, and the Research Foundation of Beijing Municipal Science & Technology Commission under Grant No.Z181100008918002.

REFERENCES

- [1] M. Heilman and N. A. Smith, “Tree edit models for recognizing textual entailments, paraphrases, and answers to questions,” in *NAACL*, 2010.
- [2] M. Wang, N. A. Smith, and T. Mitamura, “What is the jeopardy model? a quasi-synchronous grammar for qa,” in *EMNLP-CoNLL*, 2007.
- [3] W.-t. Yih, M.-W. Chang, C. Meek, and A. Pastusiak, “Question answering using enhanced lexical semantic models,” in *ACL*, 2013.
- [4] A. Severyn, M. Nicosia, and A. Moschitti, “Building structures from classifiers for passage reranking,” in *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, 2013.
- [5] K. Tymoshenko, A. Moschitti, and A. Severyn, “Encoding semantic resources in syntactic structures for passage reranking,” in *EACL*, 2014.

- [6] A. P. Parikh, O. Täckström, D. Das, and J. Uszkoreit, “A decomposable attention model for natural language inference,” *arXiv preprint arXiv:1606.01933*, 2016.
- [7] W. Yin, H. Schütze, B. Xiang, and B. Zhou, “Abcnn: Attention-based convolutional neural network for modeling sentence pairs,” *Transactions of the Association for Computational Linguistics*, 2016.
- [8] W. Bian, S. Li, Z. Yang, G. Chen, and Z. Lin, “A compare-aggregate model with dynamic-clip attention for answer selection,” in *CIKM*, 2017.
- [9] S. Wang and J. Jiang, “A compare-aggregate model for matching text sequences,” *CoRR*, vol. abs/1611.01747, 2017.
- [10] R. S. Sutton and A. G. Barto, “Reinforcement learning: An introduction,” *IEEE Transactions on Neural Networks*, 1988.
- [11] G. A. Miller, “Wordnet: a lexical database for english,” *Communications of the ACM*, 1995.
- [12] L. Sha, X. Zhang, F. Qian, B. Chang, and Z. Sui, “A multi-view fusion neural network for answer selection,” in *AAAI*, 2018.
- [13] B. Wang, K. Liu, and J. Zhao, “Inner attention based recurrent neural networks for answer selection,” in *ACL*, 2016.
- [14] Z. Wang, H. Mi, and A. Ittycheriah, “Sentence similarity learning by lexical decomposition and composition,” *arXiv preprint arXiv:1602.07019*, 2016.
- [15] D. Savenkov and E. Agichtein, “Evinets: Neural networks for combining evidence signals for factoid question answering,” in *ACL*, 2017.
- [16] K. Tymoshenko and A. Moschitti, “Cross-pair text representations for answer sentence selection,” in *EMNLP*, 2018.
- [17] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” *Machine learning*, 1992.
- [18] D. Mollá, “Towards the use of deep reinforcement learning with global policy for query-based extractive summarisation,” *arXiv preprint arXiv:1711.03859*, 2017.
- [19] Q. Yin, Y. Zhang, W.-N. Zhang, T. Liu, and W. Y. Wang, “Deep reinforcement learning for Chinese zero pronoun resolution,” in *ACL*, 2018.
- [20] P. Qin, W. Xu, and W. Y. Wang, “Robust distant supervision relation extraction via deep reinforcement learning,” in *ACL*, 2018.
- [21] B. Pan, Y. Yang, Z. Zhao, Y. Zhuang, D. Cai, and X. He, “Discourse marker augmented network with reinforcement learning for natural language inference,” in *ACL*, 2018.
- [22] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *CoRR*, 2014.
- [23] K. Cho, B. van Merriënboer, aglar Gülehre, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” in *EMNLP*, 2014.
- [24] Y. Yang, W.-t. Yih, and C. Meek, “Wikiqa: A challenge dataset for open-domain question answering,” in *EMNLP*, 2015.
- [25] T. Jurczyk, M. Zhai, and J. D. Choi, “Selqa: A new benchmark for selection-based question answering,” *ICTAI*, 2016.
- [26] D. Shen, M. R. Min, Y. Li, and L. Carin, “Adaptive convolutional filter generation for natural language understanding,” *arXiv preprint arXiv:1709.08294*, 2017.
- [27] Y. Tay, L. A. Tuan, and S. C. Hui, “Hyperbolic representation learning for fast and efficient neural question answering,” in *WSDM*, 2018.
- [28] M. Nicosia and A. Moschitti, “Semantic linking in convolutional neural networks for answer sentence selection,” in *EMNLP*, 2018.
- [29] T. Shao, Y. Guo, H. Chen, and Z. Hao, “Transformer-based neural network for answer selection in question answering,” *IEEE Access*, 2019.
- [30] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *EMNLP*, 2014.

Development of a Filipino Speaker Diarization in Meeting Room Conversations

¹Angelica H. De La Cruz, ²Rodolfo C. Raga Jr.

{ahdelacruz@national-u.edu.ph, rodolfo.raga@jru.edu}

¹National University, Philippines

²Jose Rizal University, Philippines

Abstract—Speaker diarization pertains to the process of determining speaker identity at a given time in an audio stream. It was first used for speech recognition and over time became useful in other applications such as video captioning and speech transcription. Recently, deep learning techniques have been applied to speaker diarization with considerable success, however, deep learning are conventionally data intensive and collecting large training samples can be difficult and expensive to collect especially for resource scarce languages. This study focused on investigating a speaker diarization approach for meeting room conversations in the Filipino language. To compensate for lack of resources, a one shot learning strategy was explored using Siamese neural network. Among the experiments conducted, the lowest diarization error rate yielded to 46%. There are, however, more parameters that can be tuned to improve the diarization results. To the best of our knowledge, no work in speaker diarization dedicated for Filipino language has yet been done.

Keywords—speaker diarization, segmentation, clustering, MFCC, one shot, siamese network, meetings

I. INTRODUCTION

Communication is an important part of the daily lives of people. It is a way to impart knowledge and information to other people. For instance, a teacher is sharing his knowledge and presenting facts and ideas to his students through spoken or written discussions. Communication is also a way for people to express thoughts and feelings and at the same time, understand others. With the advent of technology, people have better means of communication and easier access to knowledge and information. For this reason, a huge amount of data are being generated everyday from people communicating around the world, and so, organizing these data and extracting information from this vast amount of data has always been a challenge.

In organizations such as universities and industry companies, communication is an essential part to their growth and development. Discussions for making decisions are always noted or recorded for future reference. However, writing notes during a meeting sometimes lead to missed or incomplete information being recorded due to the fast-paced flow of discussion, which is why audio recording of the meetings are done to avoid these situations. The audio recordings are then transcribed but this process is very time-consuming. Also, people that will use these for future reference will only be interested to know the important details. With the advancement of technology, people are now relying to automatic means of extracting important information.

Speaker diarization is a process of determining "who spoke when?" in an audio stream by partitioning the audio, that contains an unknown number of speakers and unknown amount of speech data, into homogeneous segments according

to the speaker identity [1,2,3]. It was initially proposed as a topic related to automatic speech recognition which serves as an upstream processing step, but over the years, it become very useful for many tasks such as information retrieval, providing a more structured notes for medical records, automatic notes generation for meetings, court houses and parliaments, call center data analysis, and automatic transcription of broadcast news in television or radio [2,3].

The first Machine Learning based works of speaker diarization began around 2006 but significant improvements started only around 2012 and at the time, it was considered an extremely difficult task. Most methods used back then were GMMs or HMMs based. The first freely-distributed algorithm for speaker diarization with reasonable accuracy is LIUM¹. The algorithm in the core of LIUM uses GMM with i-vectors [4]. A number of researchers have explored using neural network embedding for diarization tasks which largely focused on feedforward deep neural networks (DNNs) [6]. Currently, IBM Watson² offers a service for speaker diarization as part of the speech-to-text module in different languages. However, the Filipino language is not included in the languages catered by IBM Watson. Also, to the best of the researcher's knowledge, local researchers both in the academe and the industry have not yet developed and released a Filipino speaker diarization system. A key factor why local researchers were not able to fully explore this field is because of lack of data. But today, there has been a trend on using just a few samples in order to train a deep neural network known as one-shot learning, few-shot learning and n-shot learning that are mostly used in computer vision. While learning models notoriously require huge amounts of training data, a recent study [7] was able to show that it is possible to learn much information about a category from just one, or a handful, of examples.

Building on this trend, this study focused on developing a speaker diarization system for meeting room conversations in Filipino language through one-shot learning [9] implemented using a distinct form of neural network known as siamese neural networks [8]. This study will serve as a baseline in developing speaker diarization systems in Filipino and will also contribute to the exploration of using few samples as input in deep learning in the local setting. This paper is organized as follows: Section II discusses related works; methodology in Section III; results and discussion in Section IV; conclusion in Section V.

II. RELATED WORKS

A. Speaker Diarization

Speaker diarization is normally a process of speech segmentation followed by clustering. Speech segmentation is the task of detecting voice change point or speaker turns for

¹ <http://www-lium.univ-lemans.fr/diarization>

² <https://www.ibm.com/watson>

partitioning an audio sample into regions each of which corresponds to only one speaker [10] while speaker clustering is a process that groups the speech segments together based on the similarity and association of the speech features in each of the segment.

1) Voice Activity Detection (VAD)

One of the most important preprocessing steps in speaker diarization is Voice Activity Detection (VAD), also known as speech activity detection. It is the process of detecting speech in an audio recording. The output of other subtasks in speaker diarization highly depends on the performance of the VAD [10]. VAD classifies audio into silence, speech or non-speech segments. The general approach for VAD is using Maximum Likelihood classification with Gaussian Mixture Models (GMMs), however, this approach requires prior knowledge in order to train the model in detecting speech and non-speech. The simplest systems only include two classes which are speech and non-speech, others also include silence and noise and more complex systems add more classes such as background noise, room noise, music, cross-talk, speech noise, and speech music. One of the approaches used for VAD is the model-based approach where labelled training data is required in order to train the VAD model. Existing studies for VAD include [11,12, 13,14,15,16].

2) Speaker Segmentation

Speaker segmentation is the task of detecting voice change point or speaker turns for partitioning an audio sample into regions each of which corresponds to only one speaker [17]. Speech features used in this task depends on the domain of analysis of the speech signal. In the frequency domain, features such as Distance between Power Spectral Density, Mel Cepstral Coefficients, Fundamental frequency, pitch, and formants can be used. Methods for speaker segmentation are divided into two, first is the blind segmentation where no prior knowledge is required and the other one is the model-based segmentation. Blind segmentation is a method that is speaker, text and language independent. However, the performance of the segmentation is low. In order to improve the segmentation process, model-based segmentation was introduced. Model-based segmentations make use of language models, voice databases and speech corpus. The output of this task is a sequence of segments where the boundaries indicate speaker turns or voice change. Existing works include [18, 19].

3) Speaker Clustering

After determining segments where speaker change point occurs, these segments are clustered or grouped together based on their similarity and association of the speech features in each of the segment. This process is called speaker clustering. The most commonly used algorithm for clustering speech segments is hierarchical clustering algorithm. Both top-down and bottom-up approaches were used for general speaker diarization systems, some have also explored using the hybrid

approach [12]. The common features used are spectral energy and MFCC. It finds the closest pair of clusters using BIC distance measure and merges these clusters then iterate the process until such time it reaches the stopping criterion. The stopping criterion is determined based on the local BIC, global BIC or a certain threshold. Agglomerative clustering is sometimes done separately for each gender and bandwidth condition.

B. One Shot Learning

Over the years, machine learning and deep learning algorithms were able to successfully achieve state-of-the-art performance in various applications given large training samples [8]. However, in most cases, it is difficult and expensive to collect large datasets.

Deep learning is considered to be somehow similar to a human brain, but humans do not need thousands of samples in order to recognize an object. For instance, if humans are to recognize a face of a person, the human will only need to see that person's face for a few times, or even only once, and the human brain will already be able to recognize the person's face. Unlike in deep learning, in order for the machine to recognize a person's face, it would need hundreds or thousands of examples. This is the idea behind one shot learning, for a machine to be able to learn an object category with just one or few samples.

One shot learning is an object categorization problem introduced in 2006 [7] that is mostly found in computer vision. Since learning models requires a huge amount of training samples, [7] showed that it is possible to learn such models with only a few samples by using Bayesian Program Learning (BPL). However, BPL has its flaws such as its learning cannot be transferred unlike deep learning. Months after, Google Deepmind [9] demonstrated the ability of a memory-augmented neural network to rapidly assimilate new data, and leverage this data to make accurate predictions after only a few samples. This shows that deep neural networks can be used in one shot learning.

C. Siamese Neural Network

The term siamese means twins. Siamese networks are neural networks containing two or more identical sub network components [20]. This concept is first introduced by Bromley and LeCun in 1994 to solve signature verification as an image matching problem [20]. A siamese network is an artificial neural network that use the same weights while working in tandem on two different input vectors to compute comparable output vectors. The neural network is not learning to classify but is actually learning a similarity function between the inputs.

1) One Shot Learning with Siamese Neural Networks

A study [8] explored using siamese neural networks for one shot learning in image classification which employ a unique structure to naturally rank similarity between inputs. To implement one shot learning with siamese neural network, the first thing one should do is to create a base network. For most cases in the literature, convolutional neural network is the deep learning network used as a base network in the siamese neural network. The base network for a siamese network do not require a very large amount of data for

training since it will not be used for learning how to classify inputs, it will only be used for determining how similar the two inputs are.

a) Convolutional Neural Network

Convolutional Neural Network (CNN) is a deep learning algorithm which takes an image as an input and then outputs a class as the label of the image. The first layer of CNN is the convolutional layer where high level features of the input are being extracted followed by the pooling layer that is used to reduce the spatial size of the convolved feature and capture the dominant features. A fully connected layer is then added to the convolutional layer and pooling layer to learn non-linear combinations of the high-level features then classifies using the softmax classification technique.

In one shot learning, only a few samples are used as input to train the neural network. As the CNN will be used as the base network, CNN is trained using only few samples. The computed weights and bias value from the CNN architecture were saved and loaded to the siamese network.

The saved weights of the CNN were transferred to the siamese network that has the same parameters and weights. Assuming that the neural network model is trained properly, two pairs of input are used, the network then computes for the contrastive loss function then compare the two identical sub networks based on the loss value.

III. METHODOLOGY

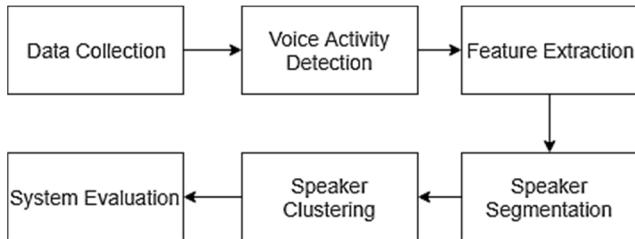


Fig. 1 Overview of the Methodology

A. Data Collection

Meeting room recordings were collected using a conference system equipment. A console (Kevler LM-500), a chairman unit microphone (Kevler LM-501) and 3 delegate unit microphones (Kevler LM-502) to tune and record the meeting. The duration of the meeting was 14 minutes and 27 seconds. There are 4 speakers in the meeting, consists of 2 male and 2 female individuals. Since the goal of the study is to explore the use of few samples for training, only 7% (52 seconds) of the data were used for training the base network and the entire recording was used for testing the model performance.

B. Voice Activity Detection

The speech data was cleaned using Audacity to remove the background noise such as the sound of the air-conditioning unit in the venue. After cleaning, the speech data underwent voice activity detection. In implementing VAD, energy features were extracted. The threshold used in detecting speech activity was set to the minimum energy seen

in the speech signal which is 12.5% of the actual maximum energy of the signal. The sample window size is set to 250 ms and a sample overlap of 100 ms. Timestamps of detected speech were stored in a JSON file.

After extracting timestamps in the audio signal that was identified as voice, the original recording was cut into segments based on the timestamps. These audio segments were concatenated again to generate a single audio file without the non-voice segments.

The training data were selected from the segments classified as audio files having voice. These segments were manually selected to have comparable lengths of audio for each speaker and also to make sure that the segments used for training is good. The segment is considered good if it has no overlapping speech and misclassified silence considered as voice. Training data were also manually labelled.

C. Feature extraction

The speech feature used in this study is Mel Frequency Cepstral Coefficient (MFCC) extracted using the *librosa* library in python. 13 coefficients were extracted in a 25 ms frame length with an overlap of 10 ms. 12-20 cepstral coefficients is typically optimal for speech analysis, depending on the sampling rate, which is why 13 number of coefficients were used and also because the first 13 coefficients stores the most important information of the speech. It is also an efficient number of coefficient because it has a smaller dimension that makes the computation faster.

D. Speaker Segmentation

In order to detect speaker change turns in the segments, one-shot learning is implemented. A base network is trained using a few samples (52 seconds or 7% of the entire audio) of data to develop the model (as shown in Figure 2). The weights of the developed model from the base network are loaded to the siamese network. The input for the siamese network are the extracted MFCC from two windows, one for each node. The siamese network determines the similarity between the two segments based on a certain threshold. Similar segments are treated as a speech from the same speaker and are not tagged as a speaker change point, while the dissimilar segments were treated as a speech from different speakers and were tagged as a speaker change point. The audio were cut to speech segments based on the speaker change point.

There are two experimental set-ups implemented for speaker segmentation and clustering. Each experimental set-up corresponds to the one shot learning model generated using CNN with different parameters as its base network.

The generated weights from the CNN model were saved as JSON file. Afterwards, it is loaded to the siamese neural network in order to create two identical sub networks. The similarity function calculates the similarity of the pair of inputs which passed through the siamese network. It will predict whether the inputs are the same or different. The similarity function is determined by the difference of the values of the computed loss for each sub network. In a siamese network, a distance-based loss function is used called contrastive loss function. It is calculated on pairs that tries to ensure that semantically similar examples are embedded close together. The experimental set-ups are shown in Table 1.

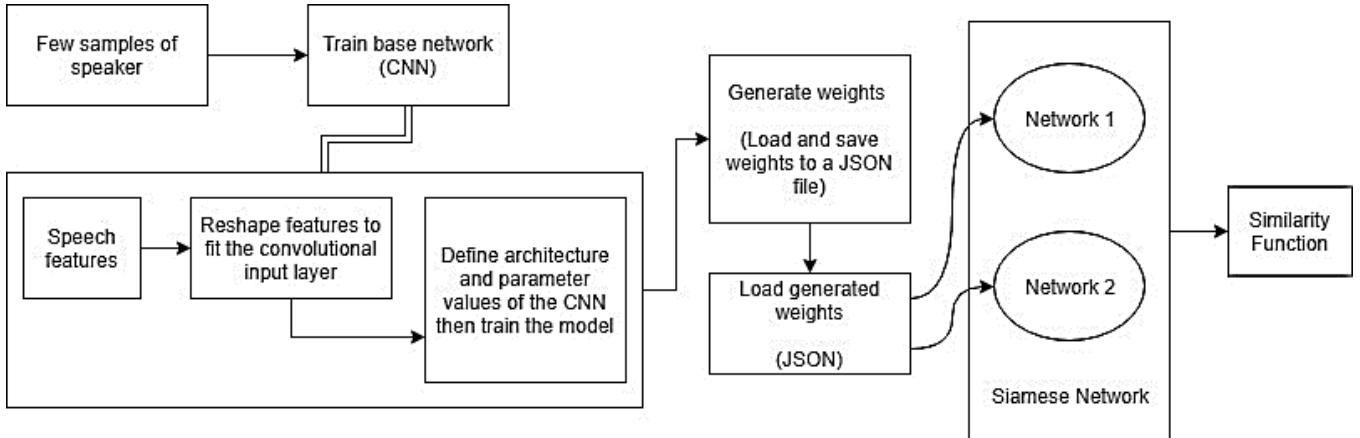


Fig. 2 Creating a base network and loading weights to the Siamese network

Table 1. Experimental set-ups for training the base network

Experiment	Base Network Architecture	Parameters
1	Convolutional layer	Dense: 32 Activation: ReLU Kernel: 1
	Pooling layer	Pool size: 1 Dropout: 0.2
	Flattened	
	3 Hidden layers	Dense: [1]128, [2] 64, [3] 32 Activation: ReLU Dropout: 0.2
	Output layer	Dense: 4 Activation: Softmax
	Model accuracy	94.72%
2	Convolutional layer	Dense: 32 Activation: ReLU Kernel: 1
	Pooling layer	Pool size: 1 Dropout: 0.4
	Flattened	
	3 Hidden layers	Dense: [1]128, [2] 64, [3] 32 Activation: ReLU Dropout: 0.2
	Output layer	Dense: 4 Activation: Softmax
	Model accuracy	84.24%

The concatenated speech segments resulted after conducting VAD were cut into segments with a length of 25 ms. MFCC were extracted for each set of two segments and

were inputted in the siamese network. The similarity score produced by the network is used by the system to determine if the previous speech segment (input 1) is spoken by the same speaker of the current speech segment (input 2). The threshold used for the similarity is 0.113, if the similarity score is less than the threshold, it is considered a speaker change, otherwise, it is considered the same speaker. The threshold is acquired by conducting a separate series of experiment, where it was found that setting a slightly higher threshold resulted to more inaccurate prediction of speaker change. This process is illustrated in Figure 3.

The contrastive loss function used in getting the similarity score is categorical cross-entropy given by the equation:

$$-(y \log(p) + (1 - y) \log(1 - p)) \quad (1)$$

where y is the binary indicator of the correct classification for an observation and p is the probability observation o is of the correct class.

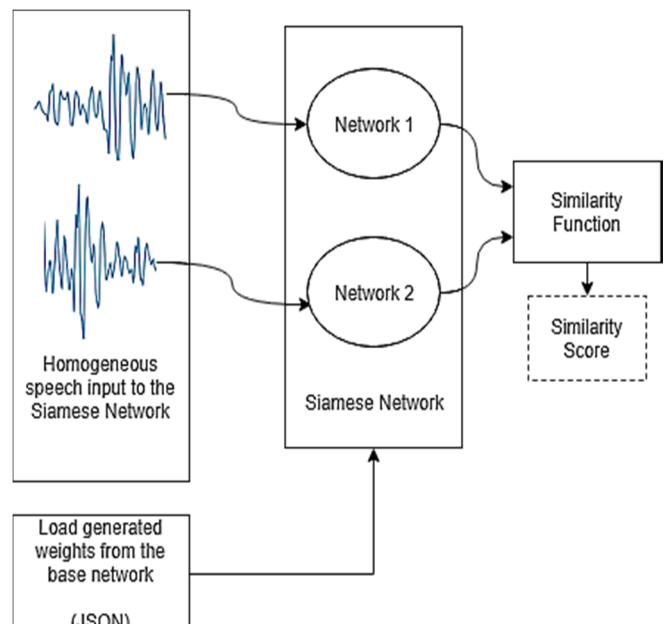


Fig. 3 Identifying similar segments

E. Speaker Clustering

Speech segments are clustered using agglomerative hierarchical clustering algorithm. In agglomerative hierarchical clustering algorithm, each vector is assigned initially as a different cluster. The Euclidean distance between all pairs of vectors are computed and stored in a distance matrix. The two closest clusters are successively merged until only one remains, obtaining the whole clustering dendrogram as output [22]. Each identified clusters of speech segments are labelled as one speaker.

F. Evaluation

In order to evaluate the speaker diarization system, Diarization Error Rate will be used. It is the main metric that is used for speaker diarization experiments as described and used by NIST in the Rich Transcription Evaluations. It is measured as the fraction of time that is not attributed correctly to a speaker or to non-speech.

$$DER = \frac{\sum_{s=1}^S dur(s) \cdot (\max(N_{ref}(s), N_{hyp}(s)) - N_{correct}(s))}{\sum_{s=1}^S dur(s) \cdot N_{ref}} \quad (2)$$

Equation 1 is used to compute for the diarization error rate where S is the total number of speaker segments where both reference and hypothesis files contain the same speaker/s pair/s. It is obtained by collapsing together the hypothesis and reference speaker turns. The terms $N_{ref}(s)$ and $N_{hyp}(s)$ indicate the number of speaker speaking in segment s , and $N_{correct}(s)$ indicates the number of speakers that speak in segment and have been correctly matched between reference and hypothesis. Segments labelled as non-speech are considered to contain 0 speakers. When all speakers/non-speech in a segment are correctly matched the error for that segment is 0.

IV. RESULTS AND DISCUSSION

The results of speaker diarization is manually evaluated by the researchers. The result of each experiment set-up is shown in Table 2.

Table 2. DER of each experiment

Experiment	DER	Average DER of various systems [23]
1	60.12 %	
2	46.46 %	42.14 %

In the first experiment set-up, the DER is 60.12%. The resulting segments in the experiment were 68 segments in total. For the second experiment set-up, the DER is 46.46%. The resulting segments were 154 segments.

The best performing set-up of the current diarization system is comparable to various diarization systems developed for a diarization challenge in John Hopkins University for Track 2 (systems that automatically estimates speech segments through VAD) [23]. The performance of the existing diarization systems range from 37.19% to 55.93% which used different features such as 24-MFCC, i-vectors, x-vectors and combined features (fusion) with and without

Variational Bayes (VB). Approaches used were BLSTM-DNN for speaker segmentation and Agglomerative Hierarchical Clustering for speaker clustering.

During the validation phase, it can be observed that the system was more able to capture and segment correctly the utterances from female speakers than utterances made by the male speakers. It can also be observed that overlap speeches were not considered by the system.

Another observation was the false detection in speaker change during the abrupt change in the amplitude in the signal. When there is a sudden spike and sudden noise, the system treats it as a speaker change.

The performance of the base network used may have also affected the diarization performance of the system. Lastly, implosive words were being segmented or was treated as a speaker change point. Utterances with low energy or low in volume were also treated as silence, which made a negative effect in the system.

V. CONCLUSION AND FUTURE WORK

In this study, a Filipino speaker diarization is developed for meeting room conversations using one shot learning implemented with siamese neural network. The speech feature used for segmentation and clustering was MFCC.

Experiments showed that it one shot learning can be an approach to develop a Filipino speaker diarization system, however, it still needs a lot of improvement.

For future work, it is recommended to work with other speech features such as pitch and LPCC.

REFERENCES

- [1] Zhang, A., Wang, Q., Zhu, Z., Paisley, J., & Wang, C. (2018). Fully supervised speaker diarization. arXiv preprint arXiv:1810.04719.
- [2] Anguera, X., Bozonnet, S., Evans, N., Fredouille, C., Friedland, G., & Vinyals, O. (2012). Speaker diarization: A review of recent research. IEEE Transactions on Audio, Speech, and Language Processing, 20(2), 356-370.
- [3] Yoav, R. (2019, February). Speaker Diarization with Kaldi. Retrieved from <https://towardsdatascience.com/speaker-diarization-with-kaldi-e30301b05cc8> by April 8, 2019. (footnote)
- [4] NIST. (2009). “The NIST Rich Transcription 2009 (RT’09) evaluation”. <http://www.itl.nist.gov/iad/mig/tests/rt/2009/docs/rt09-meeting-evalplan-v2.pdf> (footnote)
- [5] Basu, J., Khan, S., Roy, R., Pal, M., Basu, T., Bepari, M. S., & Basu, T. K. (2016, October). An overview of speaker diarization: Approaches, resources and challenges. In 2016 Conference of The Oriental Chapter of International Committee for Coordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA) (pp. 166-171). IEEE.
- [6] Wang, Q., Downey, C., Wan, L., Mansfield, P. A., & Moreno, I. L. (2018, April). Speaker diarization with lstm. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5239-5243). IEEE.
- [7] Fei-Fei, L., Fergus, R., & Perona, P. (2006). One-shot learning of object categories. IEEE transactions on pattern analysis and machine intelligence, 28(4), 594-611.
- [8] Koch, G., Zemel, R., & Salakhutdinov, R. (2015, July). Siamese neural networks for one-shot image recognition. In ICML deep learning workshop (Vol. 2).
- [9] Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., & Lillicrap, T. (2016). One-shot learning with memory-augmented neural networks. arXiv preprint arXiv:1605.06065.
- [10] Moattar, M. H., & Homayounpour, M. M. (2012). A review on speaker diarization systems and approaches. Speech Communication, 54(10), 1065-1103.
- [11] Wootters, C., Fung, J., Peskin, B., & Anguera, X. (2004, November). Towards robust speaker segmentation: The ICSI-SRI fall 2004 diarization system. In RT-04F Workshop (Vol. 23, p. 23).

- [12] Nguyen, P., Rigazio, L., Moh, Y., & Junqua, J. C. (2002, May). Rich transcription 2002 site report. Panasonic speech technology laboratory (PSTL). In Proc. Rich Transcription Workshop (RT-02).
- [13] Sun, H., Ma, B., Khine, S. Z. K., & Li, H. (2010, March). Speaker diarization system for RT07 and RT09 meeting room audio. In 2010 IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 4982-4985). IEEE.
- [14] Zhu, X., Barras, C., Meignier, S., & Gauvain, J. L. (2005). Combining speaker identification and BIC for speaker diarization. In Ninth European Conference on Speech Communication and Technology.
- [15] Kristjansson, T., Deligne, S., & Olsen, P. (2005). Voicing features for robust speech detection. In Ninth European Conference on Speech Communication and Technology.
- [16] Anguera, X., Wootters, C., Peskin, B., & Aguiló, M. (2005, July). Robust speaker segmentation for meetings: The ICSI-SRI spring 2005 diarization system. In International Workshop on Machine Learning for Multimodal Interaction (pp. 402-414). Springer, Berlin, Heidelberg.
- [17] Meignier, S., Bonastre, J. F., & Igounet, S. (2001). E-HMM approach for learning and adapting sound models for speaker indexing. In 2001: A Speaker Odyssey-The Speaker Recognition Workshop.
- [18] Hain, T., Johnson, S. E., Tuerk, A., Woodland, P. C., & Young, S. J. (1998, February). Segment generation and clustering in the HTK broadcast news transcription system. In Proc. 1998 DARPA Broadcast News Transcription and Understanding Workshop (pp. 133-137).
- [19] Gauvain, J. L., Lamel, L. F., & Adda, G. (1998). Partitioning and transcription of broadcast news data. In Fifth International Conference on Spoken Language Processing.
- [20] Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., & Shah, R. (1994). Signature verification using a "siamese" time delay neural network. In Advances in neural information processing systems (pp. 737-744).
- [21] Rao, S. J., Wang, Y., & Cottrell, G. W. (2016). A Deep Siamese Neural Network Learns the Human-Perceived Similarity Structure of Facial Expressions Without Explicit Categories. In CogSci.
- [22] Khoury, E., El Shafey, L., Ferras, M., & Marcel, S. (2014, June). Hierarchical speaker clustering methods for the nist i-vector challenge. In Odyssey: The Speaker and Language Recognition Workshop (pp. 254-259).
- [23] Sell, G., Snyder, D., McCree, A., Garcia-Romero, D., Villalba, J., Maciejewski, M., ... & Khudanpur, S. (2018). Diarization is Hard: Some Experiences and Lessons Learned for the JHU Team in the Inaugural DIHARD Challenge. In Interspeech (pp. 2808-2812).

Statistical Analysis of Syllable Duration of Uyghur Language

Askar Hamdulla, Guzalnur Dilmurat
 Institute of Information Science and Engineering
 Xinjiang University
 Urumqi 830046, China
 askar@xju.edu.cn, 1453091752@qq.com

Gulnur Arkin, Mijit Ablimit
 Institute of Information Science and Engineering
 Xinjiang University
 Urumqi 830046, China
 gulnur407@163.com, 125195364@qq.com

Abstract—Phonetics is both an ancient and a young subject. Syllables are important units of speech. Based on the data requirements of speech synthesis and speech recognition, this paper studies from the perspective of experimental phonetics. Firstly, different syllable words are counted from the large-scale "Speech Acoustic Parameters Database of Uyghur Language", including monosyllable words, two-syllable words, three-syllable words and four-syllable words. Secondly, the prosodic parameters are extracted, and statistical analysis is made. Accordingly, the duration distribution of different length words for male and female speakers are studied, and the fixed CV type syllable duration of consonant , the duration of vowel, the whole syllable duration and the pitch of syllable are extracted and analyzed. The effect of different vowels on the duration of CV syllables is further studied, and provided the reliable parameter basis for Uyghur speech synthesis and speech recognition.

Keywords-Syllable(CV); Duration; Uyghur language; Statistical analysis; Speech synthesis

I. INTRODUCTION

Uyghur language belongs to the Altai language and is attributed to adhesive language type in form. Scholars agree that there are eight unit phonetic bits a[a], ä[ɛ], e[e], i[i], o[o], ö[ø], u[u], ü[y], no compound vowels [1-3]. There are six kinds of syllable types in Uyghur language, such as V, CV, VC, CVC, VCC and CVCC. Only one vowel appears in each syllable [4]. The phonetic syllable of Uyghur is composed of vowels and consonants. Since the syllable can be formed by a unitary sound, the vocal sounds are often subjected to precise acoustic measurement in speech recognition technology. Although more Uyghur phonetics research has done a lot of work in vowels, consonants, pronunciation parts, pronunciation methods and so on, there are many basic issues still need to be studied [5].

Phonetics is one of the most important part of language, is a manifestation of language. The science of phonetics is called "phonetics". Phonetics is divided into traditional phonetics and experimental phonetics. Traditional phonetics mainly describe the phonetic features of a language by means of listening and recording. However, people's hearing ability has some limitations and subjectivity. Therefore, later more modern phonetic experiments, data, technology and other more objective research methods gradually occupy the voice of the dominant position [6]. Duration is one of the important

prosody features, and is acoustic parameters that are sensitive to the speech perception of an ordinary audience. The duration of syllables varies with interaction of syllable and rhythm features. For speech synthesis, natural speech can be synthesized only by effectively mimicking changes in the duration of natural speech [7].

Syllable is the smallest phoneme unit in which the phonemes are combined together. Syllable is also one of the most frequently debated problems in language learning [8]. The inherent words of the Uyghur language are basically monosyllabic or disyllabic. Syllables cannot consist of only a few consonants and must contain vowels. The number of vowel phonemes in a word must be equal to the number of syllables, in a word, how many vowels in the sentence is divided into how many syllables. In order to improve the accuracy of Uyghur speech recognition, it is necessary to analyze the duration of syllables of recognized speech and the duration of syllables in different contexts [9]. Although syllables can be directly divided by hearing, however, how to define the syllable exactly, how to scientifically explain the nature of syllable and how to determine the boundary between syllable are always one of the most difficult problems in phonetics [10].

II. DATA SOURCES

A. Sources of Corpus

In order to further analyze and study the duration of Uyghur syllables, considering the importance of analyzing syllables, this paper collected syllable duration from the "Speech Acoustic Parameters Database of Uyghur Language" which is built by the National Ministry of Education and the National Language Commission's National language standards construction and Informatization project on the "Speech Acoustic Parameter Database of Tibet, Uyghur, Yi", and carried out the statistics and analysis. According to statistics, we know that there are 520 (male, female) monosyllabic words, 979 (male) and 986 (female) two-syllable words, 345 (male) and 352 (female) three-syllable words and 103 (male, female) four-syllable words in the database. The corresponding proportions are shown in Figure 1 below.

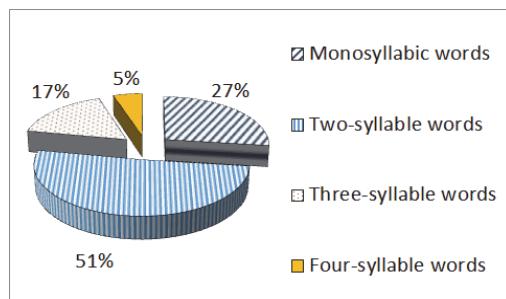


Figure1. Word Distribution in Uyghur Language Database

Also selected 1438 (male) and 1516 (female) CV type syllable. As shown in Table 1. (F - female, M - male).

TABLEI. THE DISTRIBUTION OF CV TYPE SYLLABLES IN DATABASE

	Monosyllable	prefix	Mid-fix	Suffix	Total
F	16	622	325	553	1516
M	16	599	325	543	1483

B. Recording and Data Acquisition

We carried out the recording task in the standard studio of the Ethnology and Anthropology Institute of the Chinese Academy of Social Sciences by using IBM R51 laptop and the external sound card. The pronunciation collaborators are male, female, at the age of 30 to 40, are professional announcers of China National Radio. Each word individually read twice. Acoustic parameters extracted by using the PRAAT Speech Analysis Software [11]. Such as: duration of syllable, duration of consonant, duration of vowel and overall syllable duration of CV type syllables.

III. EXPERIMENTAL DATA ANALYSIS

A. Syllable Duration Distribution Statistics of Uyghur Language

According to the distribution of the speaker's duration, draw male and female speakers monosyllabic words (520 for men and women), two-syllable words (male: 979, female: 986), three-syllable words (male: 345, female: 325) and the distribution of the duration of the four-syllable syllables (103 for men and women). Figure 2 to Figure 9 show the overall distribution of syllable duration for male and female speakers, M (male) for male speakers and F (female) for female speakers.

It can be seen from Fig. 2 and Fig. 3 that the monosyllabic effective duration of male speaker's is 661(ms), the shortest is 227(ms), the average duration of syllable is 438(ms), and the standard deviation is 75. As can be seen from the figure, there is a peak at 436(ms), and the overall distribution of duration is concentrated. of the longest duration of a female speaker's monosyllabic is 705(ms), the short is 105(ms), and the calculated average duration and standard deviation are 421(ms) and 83, respectively. As can be seen from the figure, there is a peak at 486(ms). It is as concentrated as the male speaker. Male

and female speaker duration distributions are consistent with positive distribution.

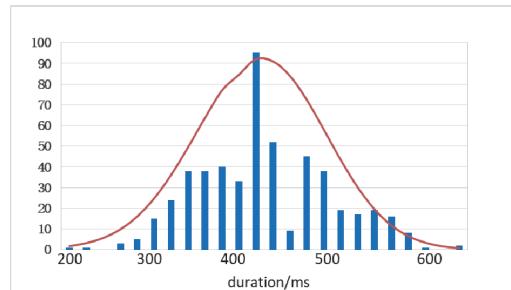


Figure2. Duration Distribution of Monosyllabic Words (M)

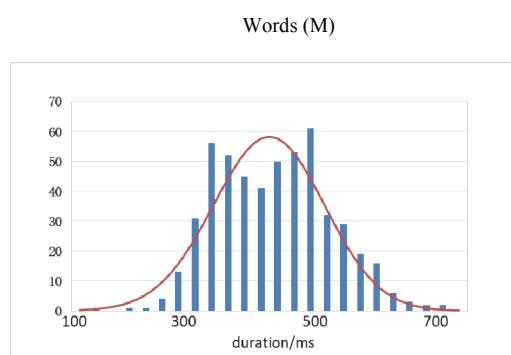


Figure3. Duration Distribution of Monosyllabic Words (F)

As shown in the Fig. 4 and Fig. 5, that the male and female speaker's two-syllable duration are roughly the same.

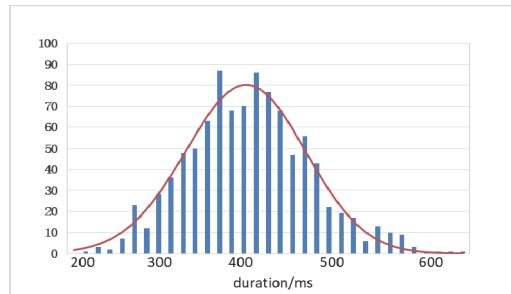


Figure4. Duration Distribution of Two-syllable Words (M)

Further analysis of the duration distribution reveals that the male speaker's two-syllable duration is 827(ms), the shortest is 293(ms), and the average duration of syllable is 530 ms. Peaks appear at 497(ms)and 548(ms). The longest and shortest duration of female speakers are 943(ms) and 312(ms), respectively, with an average duration of 572(ms). The duration value is mainly concentrated in 557(ms). As a whole, duration of two-syllable words for male and female distribution pattern is uniform. In the general trend, female are longer than male.

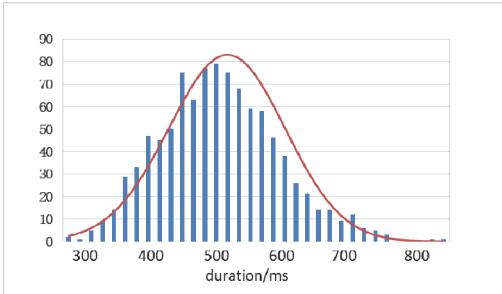


Figure5. Duration Distribution of Two-syllable Words (F)

As is shown in the above Fig. 6 and Fig. 7 that the distribution of the duration of the three-syllable words of the male and female speakers is scattered.

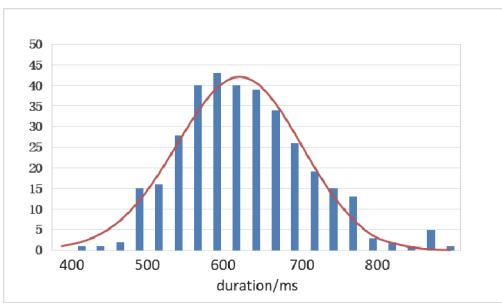


Figure6. Duration Distribution of Three-syllable Words (M)

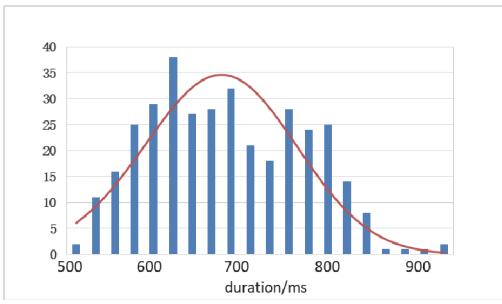


Figure7. Duration Distribution of Three-syllable Words (F)

Further analysis shows that the male speaker has a longest duration of 871(ms) and a minimum duration of 397(ms). The calculation shows that the average duration is 619 ms and there is a peak at 595(ms). The average length of the three-syllables of female speakers is 709(m), and the duration distribution is mainly concentrated at 654(m). The longest duration is 978(ms) and the shortest duration is 513(ms). The average duration of female speakers is greater than the average duration of male speakers.

From the Fig. 8 and Fig. 9 above that the average duration of the four-syllables of the male and female

speakers is 768(ms) and 850(ms), respectively, and the longest duration of the male speaker is 1058(ms), and the shortest is 488 ms. It can be seen that there is a peak at 849(ms) on the distribution map. The female speaker has a maximum duration of 1153(ms), a minimum of 564(ms), and a peak at 884(ms). Duration of four-syllable words for male and female overall decentralized distribution of female than male.

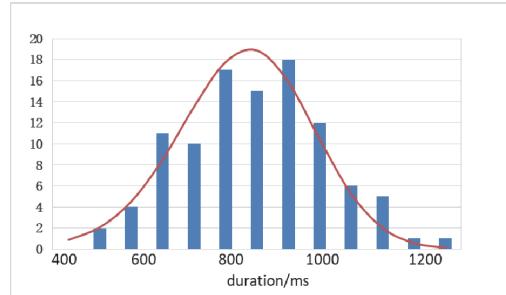


Figure8. Duration Distribution of Four-syllable Words (M)

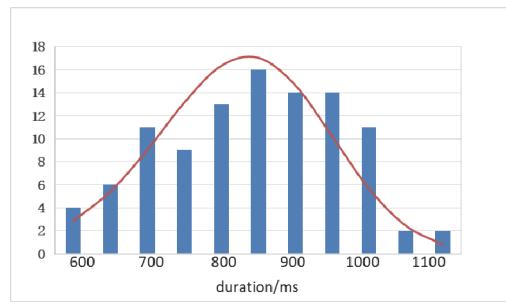


Figure9. Duration Distribution of Four-syllable Words (F)

Table 2 shows syllable duration statistics of male and Table 3 shows syllable duration statistics of female. (AVG - average, STDEV - standard deviation).

TABLEII. DURATION STATISTICS OF AVERAGE SYLLABLE FOR

	MALE				
	Male				
	Max(ms)	Min(ms)	AVG(ms)	STDEV	Sample Points
Monosyllabic words	661	227	438	75	520
Two-syllable words	827	293	530	85	979
Three-syllable words	871	397	619	82	345
Four-syllable words	1058	488	768	116	103

TABLEIII. DURATION STATISTICS OF AVERAGE SYLLABLE FOR

	FEMALE				
	Female				
	Max(ms)	Min(ms)	AVG(ms)	STDEV	Sample Points
Monosyllabic words	1153	564	850	103	979
Two-syllable words	1058	488	768	116	103
Three-syllable words	978	513	709	82	345
Four-syllable words	1153	564	884	116	103

	Max(ms)	Min(ms)	AVG(ms)	STDEV	Sample Points
Monosyllabic words	705	152	421	83	520
Two-syllable words	943	321	572	99	986
Three-syllable words	978	513	709	95	352
Four-syllable words	1153	564	850	132	103

B. Duration Analysis of CV type Syllable

There are 1438(male) and 1516(female) CV type syllable in “Uyghur speech acoustic parameters database”, C represents consonants and V represents vowels. As shown in Table 4, CV type syllable duration of the overall distribution. (CD - consonant duration, VD - vowel duration, SD - syllable duration).

TABLEIV. DURATION DISTRIBUTION OF CV TYPE SYLLABLE

Separate pronunciation		Polysyllabic words						
Monosyllable		Prefix		Mid-fix		Suffix		
M	F	M	F	M	F	M	F	
CD (ms)	120	111	98	86	68	69	78	82
VD (ms)	238	255	81	89	67	82	183	218
SD (ms)	358	366	179	180	134	149	261	298

C. Duration of Consonants , vowel and syllable

Generally speaking polysyllabic words pronunciation of consonant is inadequate, especially the consonant in the tail and pronunciation of a faster pronunciation speaker, there will be noise reduction phenomenon, a stop sounds and fricative when the tongue is not in place. With the increase of phonemes in syllables, the phenomenon of inadequate consonant pronunciation is more prominent. In the natural speech stream will be more obvious [12].

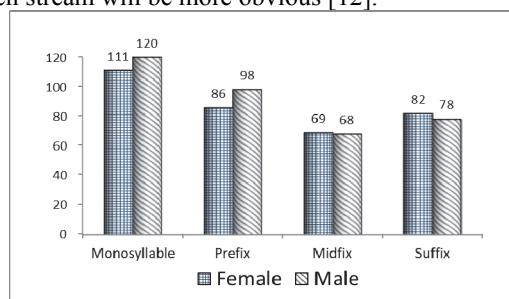


Figure10. Duration of Consonants

As shown in the Figure 10 consonant duration statistics in CV type syllables, overall, the duration of consonants when the CV type syllables are pronounced alone is longer than the prefix syllable, mid-fix syllable and suffix syllable. In a polysyllabic word the prefix syllable has the longest consonant duration, the consonants mid-fix have shortest consonant duration. Duration of the male consonant is longer than the female in the CV type syllables monosyllable and the prefix syllables of polysyllabic words.

Duration of female consonant longer than male in the mid-fix syllable and suffix syllable.

In the Uyghur language, the pitch of vowel is stronger than the consonant. Duration of vowel is lower than the aspiration consonant and fricative, but similar to semivowel, lateral and nasal duration [13]. The duration distribution of vowel pattern of CV type syllables are shown in the Figure11. The duration of vowels when the CV type syllables are pronounced alone is longer than the prefix syllable, mid-fix syllable and suffix syllable. The longest duration of vowels in polysyllabic words is the suffix syllable and the shortest is mid-fix syllable. Whether in monosyllabic words or polysyllabic words, the CV type syllable duration of vowel female higher than the male.

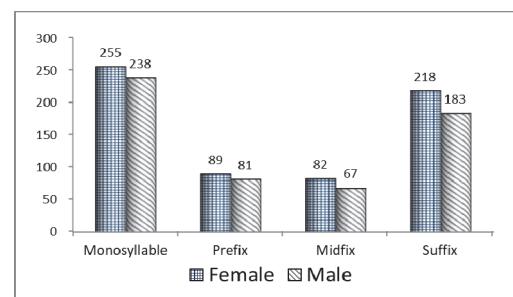


Figure11. Duration of Vowel

Figure 12 shows the duration distribution of CV type syllables contained in the word.

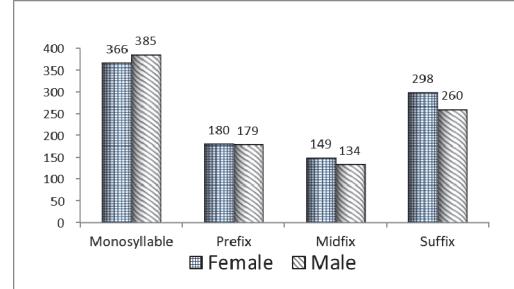


Figure12. Duration of CV type Syllable

Duration CV Type syllable pronounced alone are longer than duration of syllable for prefix syllable, mid-fix syllable and suffix syllable, duration of prefix syllable longer than the mid-fix syllable, the duration of mid-fix syllable is shortest, this means that syllables in the word position will affect duration of syllable. When the CV syllable is monosyllable, the duration of male longer than the female, in polysyllabic words female longer than the male.

D. Pitch distribution pattern of CV syllable

Pitch is determined by the frequency of vibration, if the frequency is high and the pitch is high. Figure 13 is a schematic of the pitch (average) distribution pattern of the CV syllables contained in the word. As can be seen from the figure, the starting frequency distribution pattern is: suffix > mid-fix > monosyllable words > prefix. Frequency

distribution pattern of turning point: monosyllable words > prefix > mid-fix > suffix. The mode of the end point frequency distribution is: prefix > mid-fix > suffix > monosyllable words. CV syllables appear at the end of the word with the highest starting frequency. The end point frequency is lowest. The frequency changes the most.

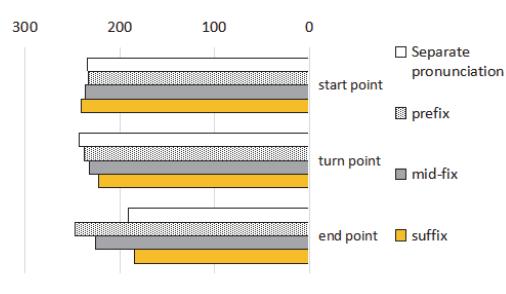


Figure13. pitch distribution pattern of the CV syllables

E. Statistical Analysis of Different Vowel in CV type Syllables

There are a total of 32 phonemes in the modern Uyghur standard language, eight of which are vowel phonemes. They are: a[ɑ]、ä[ɛ]、e[e]、i[i]、o[ɔ]、ö[ø]、u[u]、ü[y]. As shown in Table 5:

TABLEV. VOWEL PHONEME PATTERN OF UYGHUR LANGUAGE

Vowel	Lip-opened		Lip-rounded	
	Front	Back	Front	Back
High and Half-high	i		y	u
	e		ø	ɔ
Low	ɛ	a		

The patterns of the eight vowel phoneme of Uyghur in the speech system. The effect of the duration of the eight vowels on the duration of the inherent CV type syllable, the last syllable is easily prolonged, regardless of two-syllable words or three three-syllable words [14]. Variance analysis of syllable duration when the inherent CV type syllables contain different vowels gives $F(7, 1488) = 42.39 > Fa, P < 0.05$. It is clear that eight vowels have a significant effect on the duration of a syllable. Figure 13 is the duration of CV syllables in different vowels. The CV syllables containing vowel “a” and “ɛ” have longer duration and the syllables that contain “e” vowels has the shortest duration. To sum up, the syllables with a lip-rounded vowel have longer duration and longer than the syllables with the lip-opened vowels. In general, duration of syllable for female are longer than those of male.

Finally, we analyze the CV-type syllables in different vowels and get the result:

$$\epsilon > a > \emptyset > o > u > y > i > e.$$



Figure14. Duration of Different Vowel in CV type Syllables

IV. CONCLUSION

Through the statistics and analysis of the overall distribution patterns of syllable duration in Uyghur language, the distribution of monosyllabic words and two-syllable word in male and female is concentrated, and the syllable duration distributions of three-syllable words and four-syllabic words are relatively scattered. Generally speaking. The physiological difference in female vocal cords is shorter than that of male and the intonation of the female has a wide range of changes. male usually flat end, compared to the firm. Therefore, the duration of syllable for female is longer than the male. We can realize gender recognition from duration features.

In this article, we analyzed the duration of syllable when fixed CV syllables are in different positions of words, and extracted the duration of consonants, the duration of vowel and the duration of syllables. We found that the duration of consonant of CV syllable is the longest when it is pronounced alone. In a polysyllabic word, the prefix syllable is the longest, and the shortest syllables are the mid-fix syllable. In a word, duration of monosyllable > prefix > suffix > mid-fix. Duration of vowel is longer when pronounced alone, the longest as suffix syllable, and the shortest as mid-fix syllable in the polysyllabic words. In a word, monosyllable > suffix > prefix > mid-fix. Duration of CV type syllable in separate pronunciation is the longest. In the polysyllabic words the longest duration at the suffix syllable, the shortest duration at the mid-fix syllable. In a word, monosyllable > suffix > prefix > mid-fix. All this confirms that the duration of the syllables are affected by the syllables before and after. Pitch and duration are the main acoustic correlate of word stress [15]. The duration distribution pattern and pitch distribution pattern of the CV type syllables in polysyllabic words can be seen, The CV syllables are long at the suffix, and the pitch is dominant in the prefix and mid-fix. It can be proved that the traditional pronunciation of the "Uyghur word stress generally falls on the last syllable of the word". An analysis of the duration of syllables when CV type syllables contain different vowels reveals that the duration of syllables with low vowels is the longest, and the duration of syllables with high vowels is shorter. This explains that the tongue position can be a reason for the difference in duration of syllable.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (NFS; grant 61462085, 61662078, and 61633013) and National Key Research and Development Plan of China (2017YFC0820602).

REFERENCES

- [1] Aynur Nurtay, Zulpiye Aman, Dilmurat Tursun, et al. "Research on vowel patterns of disyllable words in Uyghur language," Computer Engineering & Science, 35(1), 2013, pp. 185–190.
- [2] Gulnur Arkin, Askar Hamdulla, Dilmurat Tursun. "Format Features of Vowel Harmonious Word in four-syllabic Uyghur Words." Communication Technology 11, 2013, pp. 47–50.
- [3] Askar Hamdulla. "Experimental Phonetics Based Acoustical Analysis of Nasals in Uyghur." Journal of Chinese Information Processing, 2012, 26(1), pp. 110-118.
- [4] Zilikam Kasim, Nasirjan Tursun and Wushour Silamu, "Acoustic Analysis of the Initial Syllabic Vowels in Uyghur Language", Journal of Chinese Information Processing, 2009,23(5), pp. 114-118.
- [5] Wang Kunlun, Zhang guanhong and Turgunjan Abdurrim, "Analysis of Acoustic Frequency Feature of Uyghur Vowels and Their Identification" , Journal of Chinese Information Processing, 24(2), 2010, pp. 122-125.
- [6] Zhang Qian. "Phonetic Empirical Study on the Relationship between Syllable Duration and Corresponding Information Carried in Mandarin Chinese", Anhui: Anhui University, 2016.
- [7] Feng yongqiang, Chu min, He lin, Lv shinan. , "A statistical analysis to syllable duration of Chinese". In Proc. of 5th National Conference on Modern Phonetics, Beijing. 2001.
- [8] Wu Zongji, Lin Maocan. Essential of Experimental Phonetics, Peking University Press, 2014, pp. 214.
- [9] Zhao Xiangru, Zhu Zhining. The Uyghur Jian Zhi, national press, 1985, pp. 14.
- [10] [Lin Tao, Wang Lijia. A Course in Phonetics, Peking University Press, pp. 91, 2013.
- [11] Seyyare Imam, Aynur Nurtay, Akbar Pattar, Askar Hamdulla. "The Patterns of Vowels in Monosyllabic Words of Uyghur Language", International Journal of Database Theory and Application, 9(3), 2016, pp. 113-122.
- [12] Wang Kunlun. "A Study of Uighur Syllable Speech Recognition and the Base Element of the Recognition", Computer science, 30(7), 2003, pp. 182-184.
- [13] Wang Hui, Numemel Yolwas, Wushouer Silamu, "Acoustic Feature Analysis of the Uyghur Phonemes", Journal of Chinese Information Processing, 28(1), 2014, pp. 100-106.
- [14] Gulnur Arkin, Askar Hamdulla. "Comparitive Analysis on Acoustic Parameters of Uyghur Word Vowel Harmony". Conference of the Oriental Chapter of International Committee for Coordination and Standardization of Speech Databases and Assessment Technique (Cocosda), Seoul. , 2017, pp. 353-358.
- [15] Ranagul Dagut, Askar Hamdulla, Dilmurat Tursun. "Acoustic Analysis on Prosodic Feature of CVC Type Syllable in Uyghur Language", Computer Engineering, 37(9), 2011, pp. 193-195.

A Comparative Analysis of Acoustic Characteristics between Kazak & Uyghur Mandarin Learners and Standard Mandarin Speakers

Gulnur Arkin, Gvljan Alijan

Institute of Information Science and Engineering
Xinjiang University
Urumqi 830046, China
gulnur407@163.com, 1582888526@qq.com

Askar Hamdulla, Mijit ablimit

Institute of Information Science and Engineering
Xinjiang University
Urumqi 830046, China
askar@xju.edu.cn, 125195364@qq.com

Abstract —In this paper, based on the vowel and phonological pronunciation corpora of 20 Kazakh undergraduate Mandarin learners, 10 Uyghur learners, and 10 standard pronunciations, under the framework of the phonetic learning model and comparative analysis, the method of experimental phonetics will be applied to the Kazak and Uyghur learners. The learners and standard speaker Mandarin vowels were analyzed for acoustic characteristics, such as formant frequency values, the vowel duration similarity and other prosodic parameters were compared with the standard speaker. These results are conducive to providing learners with effective teaching-related reference information, providing reliable and correct parameters and pronunciation assessments for computer-assisted language teaching systems (CALLs), as well as improving the accuracy of multinational Chinese Putonghua speech recognition and ethnic identification.

Keywords-Mandarin monophthongs , Uyghur , Kazakh learners , Acoustic features; comparative experiment , Ethnic identification

I. INTRODUCTION

With the social progress and economic development, the communication field of human beings has also expanded rapidly. In the process of communication, the ability of speaking standard and fluent Mandarin has become an important criterion for measuring conversational ability. At present, Mandarin (Putonghua) has been promoted very comprehensively in China and is now developed to a new stage [1] [2]. At the same time, in the past few years, Mandarin (Putonghua) has become a tool for information exchange between ethnic minorities in daily communication and also in the field of intelligent information processing systems.

Mandarin Chinese (MC), as a lingua franca for Chinese people, has received considerable attention in previous studies by scholars with varied academic backgrounds. Yet, comparatively few studies have been carried out on the influence of one's dialect on the production of MC, which learners acquired as a "second language", especially in the case of the MC produced by minority groups, for example, Kazakh、Uyghur speakers. Such impact is most evident in the production of target vowels of MC, which is also the most difficult to eliminate. It thus makes Chinese speech processing difficult when it comes to "accented" MC,

especially in the building of automatic speech recognition system. For that reason, this paper will carry out an experimental study on the features of MC monophthongs by Kazakh and Uyghur speakers.

As known to all, for the Mandarin learners, the pronunciation of Mandarin will be interfered by their native accent, which will cause some deviation between the Mandarin learners and the native Mandarin speakers. Therefore, the acoustic character of the Mandarin learners other than the standard Mandarin speech recognition system, which will reduce the recognition rate, is used to recognize between the Mandarin learners and the native Mandarin speakers. Many scholars have been engaged in research in this field, and have made many academic achievements. For example, a bilingual comparative experiment was conducted on the acquisition of Tibetan Mandarin learners whose mother tongue is not Mandarin [3]. So far, there have been many researches on the learning of Mandarin by Kazakh and Uyghur students in the academic circles [4][5], but there is few research on the acoustic characteristics of Mandarin learning in Kazakh and Uyghur learners.

For most students from minority regions, L1 is the native language (Kazakh and Uyghur Language for example), with Chinese as L2 and English as L3. Given the differences and similarities among the three languages in their vowel systems, such a learning process may lead to the phonetic transfer of L1 and L2 on L3 in the process of L3 vowel learning. So the present research can help strengthen phonetic teaching among ethnic minorities and provide more effective empirical data for speech synthesis and speech recognition of ethnic minorities' [6-8].

Contrastive Analysis Hypothesis (CAH) [9] and Speech Learning Model (SLM) [10] are closely related to the studies of second language (L2) acquisition. CAH claims that the principal barrier to L2 acquisition is the interference of the first language (L1) system with the L2 system. It predicts that L2 phonemes that are similar to L1 phonemes will be "easy", while L2 phonemes that are different from those in L1 will be "hard". Flege [11] obtained evidence against CAH, i.e., the "new" doesn't mean the "difficult", while the "similar" doesn't mean the "easy". "A new phonetic category can be established for an L2 sound that differs phonetically from the closest L1 sound if bilinguals discern at least some of the phonetic difference between the L1 and L2"[9]. It predicts that successful L2 production

relies on the construction of new phonetic categories for the constituents in L2 which do not have any counterparts in L1. So far, various experiments have been conducted under the framework of SLM, such as Wang & Deng [12] and Wang & Li [13]. However, comparatively few studies have been carried out on the influence of Kazakh and Uyghur speakers' mother tongue on their production of MC monophthongs. This paper is thus to examine such influence in the case of Kazakh and Uyghur speakers through an acoustic experiment.

This paper focuses on the acoustic characteristics of the pronunciation of Mandarin vowels of the Kazakh and Uyghur Mandarin learners and native Mandarin speakers. Based on the method of acoustic features, the vowel pronunciation experiment of Mandarin learners is used to compare the acoustic characteristics of the vowels of Uyghur and Kazakh Mandarin learners through quantitative experimental research methods. The purpose of this research is as follows: it is beneficial to provide learners with effective reference information related to teaching, provide reliable and correct parameters and pronunciation evaluation for computer-aided language teaching system (CALL), and also improve multi-ethnic Mandarin speech recognition accuracy.

II. EXPERIMENTAL DESIGNS

A. Subjects

This paper studies the acoustic characteristics of the pronunciation of Mandarin vowels of the Kazakh and Uyghur Mandarin ethnic learners, that is, the acoustic characteristics of the native Kazakh speakers and native Uyghur speakers in the process of learning Mandarin. Therefore, the learners' Chinese minority Chinese proficiency test (MHK) oral scores are used as the basis. In the specified range, 15 Kazakh students (7 males and 8 females) and 10 Uyghur students (5 males and 5 females) from Xinjiang University, aged 20-26 were selected as subjects. They started learning Mandarin from the third grade of elementary school and their Chinese proficiency was MHK Level 4. Their Mandarin teachers were from the same nationality (either Kazakh or Uyghur) and have passed the MHK test. And also the Mandarin level of these students' parents was poor. The everyday communication between the students and their parents is mainly carried out in their national language (either Kazakh or Uyghur) and there is no obvious hearing impairment of these students and the articulation is clear. The other group is made of 10 students in Han nationality and comes from Beijing. They are native Mandarin speakers and their average age is 24 years old.

B. Stimuli and Procedure

The voice information required for the experiment is collected in a dedicated recording studio, and the hardware devices used include a notebook, an external sound card, a microphone, and some interconnected data lines. The

external sound card can achieve functions such as adjusting the volume of sound, reducing noise, and monitoring the popping sound. The acquisition software is a recording project written in the Matlab environment. The reading data of each subject was 50 Chinese single bytes, and the acquisition frequency was 16 kHz. After collecting the voices of the participants, the collected Kazakh Mandarin learners' data and the native Mandarin speakers' voices are marked. In this study, manual audio proofreading was performed on 35 people's voices, which achieved at a high accuracy. Based on the labeled monosyllabic notes, the Praat speech software can be used to generate a three-dimensional map of the sound samples and extract the formant acoustic parameters.

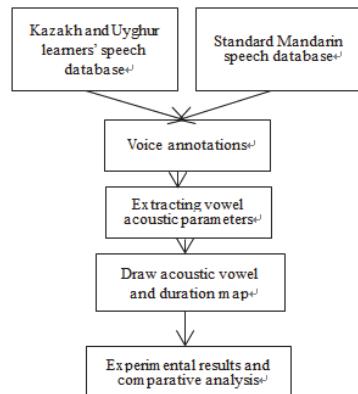


Figure 1. Experimental Flow Chart

III. EXPERIMENTAL RESULTS AND ANALYSIS

C. Formant analysis

The unit sound refers to a vowel whose tongue position, lip shape, and opening degree are always unchanged during the pronunciation. One of the important quality signs of vowels is the vowel formant, which resonates through the excitation source as a vibration. Different vowels have different acoustic cavity shapes, and thus have their respective formants [14]. From the study of acoustic characteristics, we can learn that the learner's pronunciation vowel has a certain influence on the F1 and F2 values. When the learner pronounces the vowel, the F1 frequency value is inversely proportional to the height of the tongue. It can also be said that when the learner pronounces vowel the higher the tongue position, the lower the first formant, the lower the tongue position, the higher the first resonance peak; when the learner pronounces, the tongue position is proportional to the F2 frequency value, that is, the learner's tongue is on the front. The higher the second resonance peak, the lower the tongue position, and the lower the F2 frequency value [15].

In order to objectively and accurately study the acoustic characteristics of the learner's Mandarin phonetic output, here we extract the F1 and F2 of all the tested speeches in order to compare and analyze the formant frequency values, and extract all the resonances peak data [16] in the vowel

relative segment. Since the speech has significant individual characteristics, that is, in order to eliminate the individual differences of different speakers, the formant is used to calculate the formant data. But in this situation, the data of each phonetic sample cannot be obtained, that is, the characteristics of the sound are not truly reflected. Therefore, the sound pattern theory of Shifeng is adopted in this study. F2 to F5 is normalized, and we further calculate the average value of each resonance peak of each group, and convert it into Bark value [17]. We will further normalize the Bark values, formulas (1) and (2), as follows:

$$Bark = 7 * \ln \left\{ (f / 650) + \left[(f / 650)^2 + 1 \right]^{\frac{1}{2}} \right\} \quad (1)$$

$$V = \frac{B_x - B_{\min}}{B_{\max} - B_{\min}} \times 100 \quad (2)$$

D. Acoustic analysis

The data of the first formant (F1) and the second formant (F2) of the vowel determine the sound quality of the vowel. After extracting the formant frequency of the vowel and converting it to the Barker value and normalizing it, an acoustic vowel map can be drawn. The acoustic vowel map is a visualization of the resonance characteristics of the entire acoustic cavity, which is much more accurate than the conventional tongue map. We used one-way analysis of variance to analyze the significant differences in pronunciation between Kazakh and Uyghur Mandarin learners and the native Mandarin speakers. Figures 1 shows the difference of acoustic vowels pronunciation of Kazakh female Mandarin learners and the native female Mandarin speakers, Figures 2 shows the difference of acoustic vowels pronunciation of Uyghur female Mandarin learners and the native female Mandarin speakers, Figures 3 and 4 are the difference between Kazakh male Mandarin learners and the native male Mandarin speakers in acoustic vowels pronunciation, respectively,

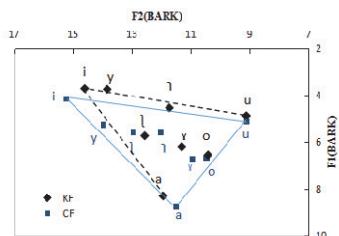


Figure 2. Kazakh learners' acoustic vowel map

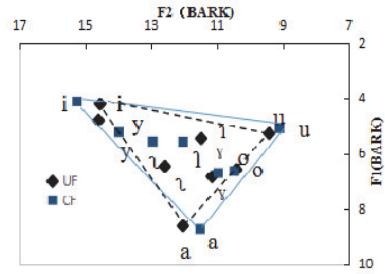


Figure 3. Uyghur learners' acoustic vowel map

Figure 2 shows that Kazakh female Mandarin learners and native female Mandarin speakers have systematic differences in unit sound output. Through the results of one-way analysis of variance, it is concluded that the tongue position of the two groups is significantly different. Except for /h/, the F1 of the Kazakh female learner's Mandarin phone is smaller than the native speakers'. In addition to /a, γ / F2 of the unit sound, others are smaller than the native speakers of Mandarin. In terms of /i, y, ɿ/, the average of F1 and F2 of Kazakh female learners is higher than that of native Mandarin speakers. The average values of F1 and F2 are slightly lower, and it can be inferred that the tongue position of the Minority Mandarin (Kazak & Uyghur) learner when pronouncing the vowels is higher. And the vowel /i/ ($pF1<0.05$, $pF2<0.05$) is the most different from that of the native speakers'. Secondly, the difference between vowels /y, ɿ/ ($pF1<0.05$, $pF2>0.05$) is significant in F1. The main difference between Kazakh Mandarin learners and the native Mandarin speakers is the high and low tongue position. The second formant data does not reflect the difference of specific statistical significance ($pF2>0.05$). In terms of vowels/a/ and /ɿ/, the average value of F1 of Kazakh female learners is lower than that of the native Mandarin speakers, and the F2 frequency value of the output is higher than that of the native Mandarin speakers. What's more, vowel /a/ (The $pFe1<0.05$, $pF2<0.01$) yielded a higher tongue position, which was more significant than that of the native speakers. However, the difference in output of vowels/ γ ($pF1<0.01$, $pF2>0.05$) is extremely significant. The output tongue of Kazakh female learners is significantly higher, while the difference of vowel/ γ ($pF2>0.05$) is not significant. The results of the analysis of variance show that for vowels/h, o, u/, the frequency values of the Kazakh learners and the native Mandarin speakers are ɿ, o, u/ ($pF1>0.05$, $pF2>0.05$). There is no significant difference on F1 or F2.

Figure 3 show that female Uyghur Mandarin-learners and native Mandarin speakers have systematic differences in unit sound output. Through the results of one-way analysis of variance, it is concluded that the tongue position of the two groups is significantly different. The F1 of the Uyghur female learners' vowel pronunciation of /a, ʌ, ɒ, i/ is larger than the native Mandarin speakers, while the F2 of the Uyghur female learners' unit sounds /a, ʌ, u, ʊ/ is

larger than the native speakers'. According to the analysis of variance, the difference between Uyghur female learners' pronunciation of vowels like /i, o/ ($pF1<0.01, pF2<0.01$) and native Mandarin speakers' is very obvious. The speaker's tongue of vowel /i/ is higher and the speaker's tongue position of vowel /o/ is lower and more backwards, the speaker's tongue position of vowel /u/ ($pF1<0.05, pF2<0.05$) is lower and more backwards, too. Secondly, the difference between vowels /a, e/ ($pF1<0.01, pF2>0.05$) and /u/ ($pF1<0.05, pF2>0.05$) is significant in F1. The main difference between Uyghur female Mandarin learners and native Mandarin speakers is at the position of front and back point of the tongue. The two formant data did not reflect the difference in specific statistical significance ($pF2>0.05$). When the learner issued /a, u/ vowel, the tongue position was in front, and when issued /e/, the tongue position was backwards, vowel /u/ ($pF1>0.05, pF2<0.05$) as long as the difference is in F2, the tongue position is basically the same, and the tongue position is posterior. The vowel /u/ analysis showed that for the vowel /y/ ($pF1>0.05, pF2>0.05$), the frequency value of the output had no significant difference on F1 and F2.

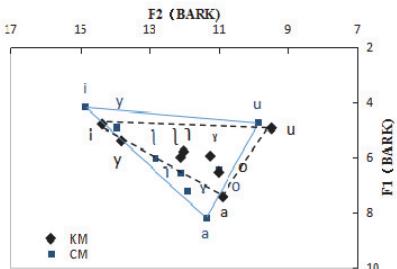


Figure 4. Kazakh learners' acoustic vowel map

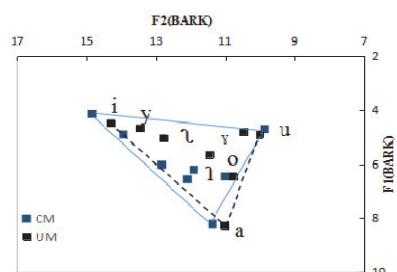


Figure 5. Uyghur learners' acoustic vowel map

As can be seen in Figure 4, when Kazakh male Mandarin learners produce unit sounds /i, y, o, u/, the F1 is higher than the native Mandarin speakers, and the F2 value of the vowel produced by Kazakh male Mandarin learners is lower than the native Mandarin speakers. As a result, the Kazakh learners' output is low. The analysis of variance showed that the output difference of vowel /i/ ($pF2<0.01$) and vowel/u/ ($pF2<0.05$) was more significant. The Kazakh learners' tongue position was significantly lower and the F2 value did not reflect the statistically significant difference ($pF2>0.05$). In addition, for vowels /y, o/ ($pF1>0.05,$

$pF2>0.05$), there is no significant difference between Kazakh learners and the native Mandarin speakers. The F1 of the Kazakh male Mandarin learners' voices of /s, γ, a, ɿ, ɿ/ are smaller than the native Mandarin speakers, except for F2 of /ɿ/, other Kazakh learners' are smaller than native speakers'. There is a discernible difference between the two groups of subjects. It can be seen that the learner's vowel output has a higher tongue position, especially in vowel/y/ ($pF1<0.05, pF2<0.05$). The difference is the most obvious. Kazakh male Mandarin learners' voices mainly distinguish vowels /a/ ($pF1<0.05, pF2>0.05$) with linguistic height, and for vowels/γ/($pF1<0.01, pF2>0.05$) the difference is mainly reflected in the height of the tongue. We can know from the analysis of variance that the difference between the F1 and F2 values of the vowel /u/ output of the learner and the native speaker is not significant ($pF1>0.05, pF2>0.05$).

We can see from Figure 5 that for Uyghur male learners, except for vowels /a, γ, i, u/, the F1 value of the Uyghur male learners' Mandarin phone is smaller than the native Mandarin speakers'. The F2 value of the unit sound of Uyghur male learner is smaller than that of the native speakers. The F1 and F2 value of the above two groups are analyzed by one-way ANOVA, and the vowels /a, γ, i, ɿ/ are known. The difference between /y/ ($pF1<0.01, pF2<0.01$) is extremely significant. The tongue position of the learner's of vowel /a, γ, i, y/ is significantly lower than the front, and /u/ tongue is higher. There was a difference in the tongue position of vowel /u/ ($pF1<0.05, pF2<0.01$), which was higher than the front. The main difference of vowel /o, u/ ($pF1>0.05, pF2<0.01$) is that the tongue position is higher or lower while the tongue position remained unchanged. The second formant data does not reflect the difference of specific statistical significance. The vowel/o/ ($pF1>0.05, pF2<0.01$) differed from the native speakers in that the tongue position was higher. In the case of /u/, the tongue is lower.

The vowel pronunciation of the minority (Kazak & Uyghur) learners and the native Mandarin speakers is analyzed by variance. In order to compare the pronunciation difference between the learners and the native speakers, the Euclidean distance is used to calculate the distance between the learners and the native speakers in the acoustic vowel. The formula is:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (3)$$

Where (x_1, y_1) , (x_2, y_2) are the two points in the acoustic vowel diagram, the Bark value of F2 is x, and the Bark value of F1 is y. The table below shows the unit distance of minority (Kazak & Uyghur) learners and the native Mandarin speakers.

TABLE I. LEARNER AND STANDARD SPEAKER VOWEL EUCLIDEAN DISTANCE

Vowel	a	o	γ	ι	τ	i	u	y
Male KM-CM	0.91	0.48	1.05	1.42	0.09	0.81	0.46	0.51
Male UM-CM	0.34	0.21	1.11	2.01	0.99	0.63	0.24	0.54
Female KF- CF	0.60	0.12	0.61	1.08	0.42	0.77	0.20	1.51
Female UF- CF	0.88	0.08 4	0.73	0.55	0.94	0.7	0.37	0.75

In Table 1, according to the Euclidean distance between the acoustic vowel diagrams, the EDKM-CM of the vowels /ɑ, o, u/ are: 0.09, 0.46, 0.48, respectively, indicating that the male learners pronounced /ɑ, o, u/ is of the best. And the EDKM-CM of vowel /y, i, a/ is 0.51, 0.81, 0.91, respectively. So it is more difficult to make these sounds for the learners, which means it is difficult for the learners to pronounce vowel /ɪ, γ/. For female learners, the EDKM-CM of vowels /o,u,γ/ are: 0.12, 0.20, 0.42, indicating female learners pronounced /o,u,γ/ at the best. While the EDKM-CM value of vowel /a,γ,i/ is 0.60, 0.61, and 0.77. So it is more difficult to make these sounds. The learners have difficulty in the pronunciation of vowels /ɪ, y/.

E. The Similarity Comparison of Vowel Duration

We need to further analyze the difference of vowel duration between the native Mandarin speakers and minority (Kazak & Uyghur) Mandarin learners. In this process, we introduce the concept of Unit Sound Duration Similarity. The Pearson correlation coefficient in equation (3.3) below will provide objective data to show the similarity of the vowel duration of minority (Kazak & Uyghur) Mandarin learners and the native Mandarin speakers. Correlation coefficient is a commonly used index to check whether the consistency of two sets of data sequences is good [18]. Therefore, it can be used to measure the acoustic consistency between minority (Kazak & Uyghur) Mandarin learners and the native Mandarin speakers. Hypothesis $X=\{x_1, x_2, \dots, x_n\}$ and $Y=\{y_1, y_2, \dots, y_n\}$ are the speech sequences of two different speakers of the same pronunciation sample sequence, respectively, n stands for the number of samples, then their correlation coefficients are calculated as follows:

$$P_{X,Y} = \frac{\sum (X - \bar{X})(Y - \bar{Y})}{(\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2})} \quad (4)$$

Where, X_i , Y_i and $i \in (1,8)$ are the corresponding eight-unit time distribution vectors, $P_{X,Y} \in (-1,1)$ means that when the linear relationship between the two variables X and Y is very high, the correlation coefficient is either 1 or -1; the correlation coefficient is less than 0; if the correlation coefficient is equal to 0, it means that the similarity between X and Y is very low and the difference is high. Table 2

below compares the similarity of vowel duration distribution between minority (Kazak & Uyghur) Mandarin learners and the native Mandarin speakers.

TABLE II. DISTRIBUTION SIMILARITY OF VOWEL DURATION

Vowel	a	o	γ	ι	τ	i	u	y
Male KM-CM	0.50	0.82	0.45	0.44	0.66	0.56	0.54	0.48
Male UM-CM	0.54	0.66	0.52	0.39	0.61	0.48	0.53	0.56
female KF- CF	0.76	0.68	0.62	0.64	0.58	0.54	0.58	0.44
Female UF- CF	0.52	0.45	0.55	0.72	0.65	0.38	0.55	0.50

In Table 2, the distribution of the duration distribution is calculated based on the distribution of the length of the native mandarin (5 male, 5 female) speakers. As can be seen from Table :

- the similarity of time length of vowel /o/ of Kazakh male Mandarin speakers is very strong, and also their correlation degree of vowel duration /γ/ is strong. The duration similarity of vowel /a, γ, ι, i, u, y/ when compared with the native male Mandarin speakers is 0.50, 0.45, 0.44, 0.56, 0.54, and 0.48, respectively. The correlation degree of vowels between Kazakh male Mandarin learners and the native male speakers is similar, indicating that the vowel pronunciation is slightly different.
- The duration similarity of Kazakh female Mandarin learners' vowel pronunciation of /a, o, γ, ι/, when compared with that of the native female Mandarin speakers, is 0.76, 0.68, 0.62, 0.64, respectively. We can see that the duration correlation with the native female speakers' is very strong. And duration similarity of vowel /i, u, y/ when compared with the native female speakers' are 0.58, 0.54, 0.58, and 0.44, respectively, indicating that the temperament correlations of the two groups of speakers are moderate.
- According to the similarity analysis between the Uyghur learners and the native learners, the duration of the vowels /o, γ/ of the Uyghur male speakers is 0.66, 0.61, respectively. The vowel /i, γ/ duration similarity of female Uyghur Mandarin-learners' is 0.72 and 0.65, respectively, which is strongly correlated with that of the native female speakers.
- While the duration of the vowel /a,o,γ,u,y/ when compared with the native female speakers', the value is 0.52, 0.45, 0.55, 0.55, and 0.50, respectively, indicating that the vowel duration of the two groups of speakers is moderate. The duration of vowel /i/ is 0.38, which is lower than the similarity of other vowel durations. It can be seen that the vowel duration of the female Uyghur Mandarin-learners is similar to that of

the native female speakers, indicating that the length of the vowel pronunciation is different.

IV. CONCLUSIONS

In this paper, 10 Uyghur Mandarin learners and 20 Kazakh Mandarin learners are taken as research object. The purpose of the contrast experiment is to provide effective reference information for the learners, and to provide a reference for the development of natural language speech synthesis and high-precision speech recognition technology for minority Mandarin learners, and to provide reliable computer-aided speech systems with a much more correct parameters and pronunciation assessment.

In this study, the experimental acoustics method was used to compare the acoustic characteristics of the Uyghur and Kazakh Mandarin learners. By comparing the experimental results:

- the Mandarin syllables of the Uyghur male learners /a, u/ pronunciation is the best, and they have problems with the pronunciation of vowels /ɿ, ɿ, i, y/.
- As for Uyghur female Mandarin learners, the vowel /u, ɿ/ pronunciation are the best. They have pronunciation difficulty in vowels /o, ɿ, i/. The best acoustic vowel performance for Kazakh male learners are /o, y/, and they have difficulty in vowels /ɿ, ɿ/, while female learners pronounce /o, u, ɿ/ the best, but also have difficulty in pronunciation of vowel /a, i/.
- In general, Uyghur learners pronounce vowels /u/ the best, while Kazakh learners pronounce vowel /o/ the best. For the duration of the vowel unit, the length of the vowels of the Uyghur learners is longer than that of the native Mandarin speakers, while the length of the Kazakh learners is always different to collect for analysis.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (NFS; grant 61462085, 61662078, and 61633013) and National Key Research and Development Plan of China (2017YFC0820602).

REFERENCES

- [1] LI Jing. "Thoughts on Putonghua Proficiency Test Work". Journal of Language and Literature Studies, 2010,PP:12-13.
- [2] KOU Fangling,YANG Jian,CHENG Yang,KUI Liping. "CV-syllable Feature Extraction for Automatic Language Identification"[J].Information and Control, 2013,42(04),PP.464-469.
- [3] Zhao L, Feng H, Wang H, et al. "Acoustic features of Mandarin monophthongs by Tibetan speakers"[C]. International Conference on Asian Language Processing (IALP). IEEE, 2014,PP. 147-150.
- [4] TIAN Jing.An "Analysis of Difficulties in Acquisition of Chinese Consonant Initials by Uyghur and Kazak Students"[J].Journal of Research on Education for ethnic Minorities,2010,21(04),PP.12-18.
- [5] ZHAO Xi. "The Analysis of Putonghua Interlanguage Monophthongs of Minority Ethnic Students in XinJiang"[C]. (NCMMSC2013) , 2013.
- [6] Gulnur.Arkin,Askar.Hamdua "Investigation of the Phonological Error Rules Of Mandarin by Uyghur Second Language Learners".[J].IPPTA: Quarterly Journal of Indian Pulp and Paper Technical Association ,2018,30(04),P.483-491.
- [7] Gulnur Arkin, Askar Hamdulla "Comparative analysis on acoustic parameters of Uyghur word vowel harmony"[C].Oriental COCOSDA ,2017.
- [8] Yultuz Rapkat, Gulnur Arkin and Askar Hamdulla "Experimental Research of Mandarin Diphthongs Produced by Uyghur Learners"[C].The 21th International Conference on Asian Language Processing (IALP 2017), 2017.
- [9] Lado, R "Linguistics across Cultures: Applied Linguistics for Language Teachers" [M]. Michigan: University of Michigan Press, 1957.
- [10] Flege, J., "Second language speech learning: Theory, findings, and problems"[J].Speech perception and linguistic experience: Issues in cross-language research, 1995, pp. 233-277.
- [11] Flege, J., "The instrumental study of L2 speech production: some methodological considerations"[J].Language Learning, 1987,(02),pp.285-296.
- [12] Wang, Y. and Deng, D., "The acquisition of the 'Unfamiliar Vowels' and 'Similar Vowels' in Chinese by Japanese Learners", Chinese Teaching in the World, 2009,(02),PP.262-279.
- [13] Wang, M. and Li, C "Phonological Perception and Processing in a Second Language" [J].Contemporary Foreign Language Studies, 2012,(12),PP.94-109.
- [14] SHI Feng. "The vowel pattern of Beijing dialect"[J].Nankai Linguistics,2002,PP,30-36.
- [15] PAN Min. "Research on Individual Characteristics of Vowels Acoustic Space"[D].University of Electronic Science and Technology of China,2014.
- [16] ZHOU Zhengjian,YAN Qin. "Analysis of Formant Distributions of four Chinese Accented Putonghuas"[J].Science paper Online,2010,5(02),PP.159-164.
- [17] LIU Fei,AN Yuxiang. "Research on Chinese vowel acquisition of Pakistani students based on experiment"[J].Science and Technology Vision,2013,(34),PP.69-165.
- [18] WANG Dongliang. "Analysis of Uyghur Students in Mandarin Tone Error Experiment"[D].Northwest Minzu University, 2014.

Construction of Quantitative Index System of Vocabulary Difficulty in Chinese Grade Reading

Huiping Wang, Lijiao Yang¹

Institute of Chinese Information Processing,
Beijing Normal University
Beijing, China
18013752306@163.com yanglijiao@bnu.edu.cn

Huimin Xiao

Chinese Language and Cultural College,
Beijing Normal University
Beijing, China
3087136946@qq.com

Abstract—Chinese grade reading for children has a broad application prospect. In this paper, Chinese textbooks for grade 1 to 6 of primary schools published by People's Education Press are taken as data sets, and the texts are divided into 12 difficulty levels successively. The effective lexical indexes to measure the readability of texts are discussed, and a regression model to effectively measure the lexical difficulty of Chinese texts is established. The study firstly collected 30 indexes at the text lexical level from the three dimensions of lexical richness, semantic transparency and contextual dependence, selected the 7 indexes with the highest relevance to the text difficulty through Person correlation coefficient, and finally constructed a Regression to predict the text difficulty based on Lasso Regression, ElasticNet, Ridge Regression and other algorithms. The regression results show that the model fits well, and the predicted value could explain 89.3% of the total variation of text difficulty, which proves that the quantitative index of vocabulary difficulty of Chinese text constructed in this paper is effective, and can be applied to Chinese grade reading and computer automatic grading of Chinese text difficulty.

Keywords-readability; grade reading; regression models

I. INTRODUCTION

“Graded reading” refers to a reading method and strategy that starts from the characteristics of children's age, thinking and socialization, selects books suitable for children of different ages and guides them to read (Xu Jianhua, Liang Haoguang, 2011). For teachers, parents and children, it is very important to choose the text with appropriate difficulty among many reading texts. The essence of “graded reading” is the grading of text difficulty.

The concept commonly used to describe the ease of writing is text readability. The readability of a text refers to the degree or nature of its readability and comprehension (Li Shaoshan, 2000; Fry, 2002). At present, the relevant researches on the readability of text are mainly divided into the researches on the readability based on the traditional formula of the readability of text features, the researches on the readability inspired by cognitive theory, the researches on the language model method based on word statistics, and the researches on the readability based on the natural language processing foundation and the machine learning model method. Most studies focus on English texts, focusing on the construction of readability formulas and the verification of their validity, such as Flesch, Spache, Powers-Sumner-Kearl and other formulas (Guo Wanghao, 2009). Although there are many kinds of formulas, and the

constants and variable coefficients in these formulas are different, the most important predictors are syntactic features (mainly manifested as sentence length) and lexical features (mainly manifested as word length and word frequency).

Vajjala & Meurers(2012) compared the influences of lexical features and syntactic features on the readability of English texts and found that lexical features have stronger influences than syntactic features, especially the word length index (the average number of syllables and characters per word) has the strongest predictive power on the readability of English texts. Different from the importance of word length in English texts, Chinese words are short and of limited types, with monosyllabic words and disyllabic words dominating. The influence of word length on the intelligibility of meaning and the readability of texts may not be as obvious as that of English words. What are the lexical indexes that affect the readability of Chinese text? Previous studies have found that the token, the type, the type-token ratio, the number of unusual words, the number of unique words, the sum of the normalized value of the whole word set, the ratio of difficult words, the number of notional words, the number of function words, the ratio of notional words to function words, and the number of fixed phrases are important lexical indicators to measure the readability of Chinese text. (Sun Hanyin, 1992; Chen Alin, Zhang Su, 1999; Zhang Ningzhi, 2000; Wang Lei, 2005, 2017; Yang Jinyu, 2008; Guo Wanghao, 2009; Chen Yujia, 2012; Zuo Hong, Zhu Yong, 2014; Sun Gang, 2015 et al.). However, these researches usually refer to the readability of English texts rather than directly focusing on the lexical features of Chinese.

The existing research results mainly have the following two problems: First, the correlation statistics between lexical indexes and text readability are carried out directly, and the specific analysis and explanation of the Chinese lexical features subordinated to the indexes are not made, as a result it is difficult for readers to know the role of lexical features in text difficulty classification directly from the statistical data. Second, there is a lack of research on the weights of different indicators in the same lexical features. For example, the number of functional words, the number of notional words and the ratio of notional to function words are all part-of-speech characteristics of vocabulary, which are proved to be important indexes affecting the readability of Chinese texts, but which indexes can best reflect the relationship between part-of-speech characteristics and text readability has not been compared and analyzed in previous

¹ Corresponding autor

studies. Even though some studies have clarified the relationship between lexical indexes and lexical features, the study of lexical features is not comprehensive and systematic, and there are also some errors in the attribution of measurement indexes. For example, Bie Xiaolei (2017) only deals with lexical features such as part of speech and frequency in the selection of lexical factors affecting the readability of Chinese texts, but not with other lexical features such as word length and lexical diversity. The number of characters and strokes should belong to the factor of Chinese characters, but Song Yaoting et al. (2013) took the number of characters and strokes with Chinese character attributes as the index of vocabulary number and vocabulary length respectively, and incorporated them into the measurement of vocabulary factors.

From the perspective of lexical features in Chinese, this paper, by referring to the lexical richness analysis framework and combining with previous research results on Chinese readability, selects possible measurement indexes from the three dimensions of lexical richness, semantic transparency and context dependence, and discusses the effective lexical parameters for measuring text difficulty. Through selecting the items, a regression model is established to measure the vocabulary difficulty of Chinese text effectively, so as to provide support for constructing Chinese readability assessing model and Chinese reading ability assessing tool from the perspective of lexicology.

II. TEXT VOCABULARY DIFFICULTY MEASUREMENT INDEX

A. Lexical Richness

“Lexical richness” mainly includes lexical diversity, lexical complexity, word frequency profile, lexical errors, and lexical density (Read, 2000). By using some of these dimensions, the researchers measured the depth and breadth of the learner’s vocabulary knowledge and assessed their vocabulary levels. Vajjala & Meurers (2012) applied lexical diversity and lexical density in the vocabulary richness analysis framework to the English text difficulty classification study, which achieved good results. In view of this, this paper develops from the perspective of lexical features of Chinese, and draws on the framework of lexical richness analysis to explore effective vocabulary parameters for measuring the difficulty of Chinese text.

1) Lexical diversity

“Lexical diversity” refers to the use of different words by learners in language expression to avoid the reuse of certain words, which can also be called “Lexical variability”. The traditional method of measuring vocabulary diversity is to calculate the ratio of the number of different words to the total number of words, namely the Type-Token Ratio (TTR). (Malvern et al., 2004) The number of times the same word appears in the text is counted as a type, and every occurrence of a word in the text is counted as a token. However, the traditional TTR is affected by the length of text, that is, if there are too many words in the text, TTR cannot effectively evaluate Lexical diversity (Arnaud, 1984; Richards, 1987).

In order to solve this problem, some researchers have proposed measures such as Guiruud index, Herdan index, Mass index, Uber index, HD-D value, average segment-like-to-shape ratio (MSTTR), and text diversity measurement (MTLD), in which the Guiruud index value is relatively stable and less affected by text size, proved to be

one of the most useful tools for studying vocabulary diversity (McCarthy & Jarvis, 2007, 2010; Torruella & Capsada, 2013). In order to eliminate the influence of text length, this paper also considers the Guiraud index on the basis of examining the type, token and TTR.

2) Lexical Complexity

Read (2000) points out that “complex word”, also known as Lexical rareness, refers to the less commonly used word (Unusualword) or advanced word (Advancedword) that appear in the text. Research suggests that the higher the proportion of complex words used, the higher the text quality and learner’s language level (Linnarud, 1986; Vermeer, 2000; Liu Donghong, 2003; McNama-raetal., 2010; Wan Lifang, 2010; Fan, 2012, etc.). In this paper, according to the “Compulsory Education Common Vocabulary”, the scope of simple vocabulary is defined as first-level and second-level words. The scope of complex vocabulary is defined as three-level and four-level words. The absolute and relative quantities of simple words and complex words in textbooks of different grades are measured respectively.

3) Lexical frequency profile

“Lexical frequency profile” refers to the use of words with different degrees of commonality in a text, which reflects the proportion of words used in each word frequency level (Laufer & Nation, 1995). In the study of English word recognition, researchers have found that word frequency is an important variable affecting word cognition. The higher the frequency of occurrence and use of a word, the easier it is to extract and activate from the psychological dictionary when reading. Conversely, the use of low-frequency words has a relatively long activation time and a relatively low level of activation. In general, if a word is often used, it becomes easier to understand because of repeated cognition. Therefore, the role of word frequency factors in distinguishing text legibility cannot be ignored. This paper refers to the frequency of use of modern Chinese words published by the National Language Committee, and divides the words into the most common words, common words, sub-common words and very words, and counts the number and proportion of different levels of words used in textbook texts. In addition, this article also considers the number and proportion of high-frequency words in the text.

4) Lexical density

Lexical density refers to the ratio of the number of words with lexical attributes (ie, words with practical meaning, including nouns, verbs, adjectives, etc.) to the total number of words (Ure, 1971). Gilliland (1972), based on a series of experimental studies, proposes that the ratio of notional words to functional words will affect the legibility of the article. Zhang Biyin also found in the book Reading Psychology (1992) that readers mainly pay attention to the notional words in the reading process, focusing on the meaning of the notional words. And people who read quickly tend to work on notional words. The more substantive words in the text, the greater the vocabulary density, the more information is passed, and the difficulty of the text increases accordingly. Vajjala & Meurers (2012) also considers the notional word ratio to be an effective indicator of the difficulty of measuring English text. Chen

Yujia (2012) used six sets of commonly used Chinese second language textbooks (496 texts) as the analysis object, and used the chi-square statistical method to select the most relevant measurement indicators. It was found that the number of notional words and the ratio of notional words were important vocabulary indicators for measuring text readability. Wang Lei (2005, 2017), Zuo Hong and Zhu Yong (2014) found that the number of function words is an important lexical factor affecting readability in the study of Chinese two-language text readability.

In view of the classification criteria for Chinese notional words and function words, there is still controversy in the academic circles. This paper adopts two ways to calculate the notional word density: one is to take the usual practice in foreign countries, dividing the notional words into nouns, substantive verbs, adjectives and adverbs, and the rest are classified as function words; the other is to draw on the word classification criteria of Zhu Dexi (1982), and to use nouns (including time words and position words, place words).), verbs (including auxiliary verbs and directional verbs), adjectives (including attribute words and status words, distinguishing words), numerals, quantifiers, pronouns (including personal pronouns, demonstrative pronouns, and interrogative pronouns) are classified as notional words, adverbs, prepositions, conjunctions, auxiliary words (including modal particles), interjections, onomatopoeia, etc. are classified as function words.

B. Semantic transparency

In 1962, S. Ullmann, a British functionalist semantician, put forward the idea of "semantic transparency/obscurity words" in his book "Semantics: An Introduction to the Science of Meaning". Semantic transparency refers to the degree to which the semantics of a compound word can be inferred from the semantics of each morpheme that constitutes a compound word. Its operation is defined as the degree of semantic correlation between the whole word and its morpheme (Wang Chunmao, Peng Yuling, 1999). Semantic transparency is usually divided into four types: total transparent, transparent, obscurity and total obscurity. The study of semantic transparency has application value in children's reading and Chinese as a foreign language. Xu Caihua (2001) studied the influence of semantic transparency on children's reading, and believed that transparent words can promote learning, while opaque words will cause certain obstacles to learning. Based on this, this paper assumes that the transparency of word meaning is also one of the characteristics that affect the readability of text.

In this paper, we use the word vector to represent the semantics of the whole word, the char vector represents the semantics of the morpheme, and calculate the semantic transparency of the compound word by calculating the semantic similarity between the word and the char. We add the morpheme char vectors and then take the mean value. Then we calculate the similarity between the morpheme char vectors and the whole word vectors. Then we can get the semantic transparency of the whole word. Our formula for calculating semantic transparency is as follows:

(1) Adding and averaging the morphological prime word vector:

$$C_m = \frac{\sum_{i=1}^n c_i}{n}$$

C_m is the average of the morpheme vectors, c_i is the morpheme vector, and n is the number of morphemes that make up the compound word.

(2) Calculate semantic similarity using cosine distance:

$$s = \frac{w \times c_m}{\sqrt{w^2 + c_m^2}}$$

In this paper, the cosine distance is chosen to calculate the semantic similarity. As shown in Equation 2, s is the similarity of the average of the morpheme vectors and the compound word vector, and w is the word vector.

(3) Normalized semantic transparency:

$$w_t = s + 0.5 \times s$$

In order to facilitate analysis, as in equation (3), we normalize the semantic similarity obtained at (2). W_t is the semantic transparency of compound words.

Based on the above method for calculating the semantic transparency of words, we use the cleaned 6.2G Wikipedia unlabeled corpus as the data set, and choose Word2vec as the word vector training tool to train the word vector. This paper calculates the semantic transparency of all the words in the textbook. We divide words with transparency higher than 0.4 into high-transparency words, such as "花园" and "吃饭", and divide words with a transparency lower than or equal to 0.4 into low-transparency words, such as "马上" and "须眉". We investigate the distribution of high transparency words and low transparency words in different difficulty levels of text.

C. Contextual Dependence

Context dependent refers to whether the understanding of lexical meaning depends on the context. For example, "刘爷爷今天早上走了". This is an ambiguous sentence, the ambiguity caused by the polysemy "走". "走" means "离开" (leave) and "死亡" (die). It depends on the context to know which one to take. In practice, there are cases where some polysemous words can have two or more interpretations in a sentence, and the specific meaning needs to be determined by context. We believe that the meaning of words is related to the difficulty of text comprehension. But is it that polysemous words will increase the difficulty of text reading? The answer is not always. The common words such as "打" and "大" have more meanings, but because they usually have specific meanings in sentences, they will not cause reading difficulties for readers. We believe that polysemy with high context dependence is difficult to read. Based on the Modern Chinese Polysyllabic Dictionary, we manually screens 2000 ambiguous words with high degree of context dependence, and investigates the distribution of these polysemous words with high context dependence in textbooks of different grades.

III. REGRESSION MODEL OF TEXT VOCABULARY DIFFICULTY

The regression model constructed in this paper can be divided into five stages: (1) pre-processing stages such as automatic word segmentation, part-of-speech tagging and punctuation removal; (2) obtain each characteristic value; (3) feature selection through Pearson correlation coefficient; (4) Lasso and Ridge models are used to carry out regression analysis of text vocabulary difficulty; (5) verify the model.

A. Corpus processing

1) Selection of text

The readability of the text depends on the difficulty of the text. It is very important to select a series of Chinese texts with distinct difficulty levels, because the main indexes affecting readability and their weights should be analyzed and refined on the basis of comparing the linguistic features of Chinese texts. Under the present condition, the primary school Chinese textbooks compiled and approved by many Chinese experts have a systematic gradient from easy to difficult, which is an ideal language material for the study of the readability of Chinese texts. This paper takes the Chinese textbooks for grade one to grade six published by People's Education Press as the corpus, selects the modern narrative, expository, argumentative and applied texts, and excludes the poems, lyrics, ballads, fu and other rhymes to build the text library. There are 576 articles in total, and the grade level of the text is its difficulty value, which is divided into 12 levels. The distribution at all levels is as follows: 36 at level 1, 41 at level 2, 42 at level 3, 38 at level 4, 42 at level 5, 52 at level 6, 51 at level 7, 55 at level 8, 54 at level 9, 56 at level 10, 54 at level 11, and 55 at level 12.

Considering that the number of texts and words at the lower level are smaller than those at other levels, in order to avoid data skew, we added 50 articles and invited 10

educational experts to grade them. The final total of each level is: 47 for level 1, 45 for level 2, 48 for level 3, 49 for level 4, 51 for level 5, 52 for level 6, 51 for level 7, 55 for level 8, 54 for level 9, 60 for level 10, 58 for level 10, and 55 for level 12.

2) Preprocessing

Word is the basic unit of sentence composition. The first step of Chinese processing is word segmentation and part-of-speech tagging. In this paper, the NPIR system for modern Chinese is used for word segmentation and part-of-speech tagging. Because there are some errors in the system, we need to proofread the results manually. The proofreading work is mainly based on the Modern Chinese Dictionary (7th edition), and exclusion of proper words such as names and place names, Western letters and alphabetic words (such as "father") in statistics. After word segmentation and part-of-speech tagging are completed, punctuation is removed.

B. Feature acquisition

On the basis of examining the correlation between the difficulty of Chinese textbooks for primary school students and various factors of lexical richness framework, semantic transparency and context dependence, this paper puts forward 30 lexical parameters to measure the difficulty of texts. The results are shown in table I.

TABLE I. QUANTITATIVE INDEX SYSTEM OF TEXT VOCABULARY DIFFICULTY

b	c	d	e	f	g	h	i	j	k
Number of type	Number of token	TTR	IOG	Number of most common words	Number of common words	Number of sub-common words	Number of non-common words	Ratio of most common words	Ratio of common words
m	n	o	p	q	r	s	t	u	v
Ratio of sub-common words	Number of non-common words	Number of complex words	Number of simple words	Number of outline words	Ratio of complex words	Ratio of simple words	Ratio of outline words	Number of highly transparent words	Number of lowly transparent words
w	x	y	z	aa	ab	ac	ad	ae	af
Ratio of highly transparent words	Ratio of lowly transparent words	Number of notional words 1	Number of notional words 2	Substantive density 1	Substantive density 2	Number of polysemous words	Ratio of polysemous words	Number of textbook high-frequency words	Ratio of textbook high-frequency words

C. Feature selection

The common feature selection methods can be roughly divided into three categories: filtering, enveloping and embedded. In this paper, the filtering method is selected. By evaluating the degree of correlation between individual features and difficulty level, the ranking leaves the features with high degree of correlation. We draw the correlation thermal diagram of each index, and measure the correlation of index based on Pearson correlation coefficient. The correlation thermal diagram is shown in figure 1.

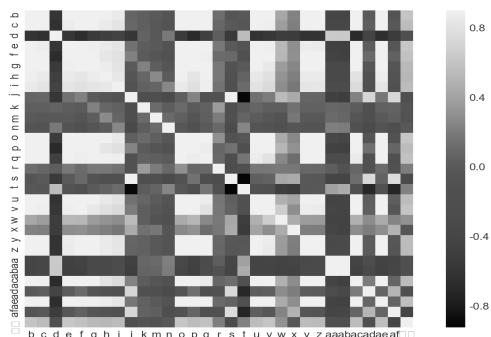


Figure 1. Thermal Diagram of Index Correlation

Usually, the correlation intensity of variables is judged as follows: correlation coefficient 0.8-1.0 is extremely

strong correlation; correlation coefficient 0.6-0.8 is strong correlation; correlation coefficient 0.4-0.6 is medium; correlation coefficient 0.2-0.4 is weak correlation; correlation coefficient 0.0-0.2 is weak correlation or no correlation. As shown in the figure, the lighter the color, the higher the correlation, we can draw the following conclusions:

- (1) IOG, number of complex words, number of most common words, number of notional words, number of lowly transparent words, and number of polysemous words are all significantly positively correlated with the text difficulty.
- (2) the proportion of non-common words, common words and highly transparent words is not related to the text difficulty.

(3) the correlation coefficient of variables such as the number of most common words, the number of common words, the number of frequently used words, and the number of non-common words is close to 1, and there may be collinearity. These variables can be used as features to predict sentence difficulty, but because of collinear problems, only one can be selected as features.

Finally, we selected the following features based on the two principles of removing indexes with high collinear rate and preferential selection of features with high contribution to the result value:

- (1) vocabulary diversity -- IOG
- (2) vocabulary complexity -- number of complex words
- (3) word frequency overview -- number of unusual words and number of textbook high-frequency word
- (4) notional word density -- notional word density 1
- (5) semantic transparency -- number of lowly transparent words
- (6) context dependence -- number of polysemous words

D. Experimental results

Taking 500 texts as training samples, this paper constructs a regression algorithm based on Lasso Regression, Elastic Net, Ridge Regression, Gradient Boosting Regression and XGBoost. In order to ensure the objectivity and accuracy of model evaluation and avoid over-fitting caused by uneven data distribution, we adopt the 10 fold cross-validation method to record the mean square error as the evaluation index, and finally take the average of 10 tests as the experimental results. Finally, it is found that the effects of the five algorithms are similar. Lasso model has the best effect, the mean square error is 0.1040 and the standard deviation is 0.0069.

Different models have their own advantages and differences, and model fusion can give full play to the advantages of each model. We choose the above five models as the basic model, and use the average value of the predicted values of the five models as the final predicted value. The final experimental results are shown in Table II.

TABLE II. EXPERIMENT RESULT OF TEXT DIFFICULTY REGRESSION

Model	Neg-mean-squared-error	
	Mean	Std
Lasso	0.1040	0.0069
ElasticNet	0.1109	0.0074
Ridge	0.1137	0.0075

Model	Neg-mean-squared-error	
	Mean	Std
Gradient Boosting	0.1068	0.0080
XGBoost	0.1151	0.0069
Averaged-Model	0.1022	0.0052

Model evaluation

To evaluate the model, we selected 100 texts for validation. The regression model trained above was used for prediction. The input values were IOG, number of complex word, number of high-frequency word, number of most common word , number of notional word, number of low transparent word and number of polysemous word in each text, and the output value was predicted difficulty value of each text. Regression analysis was conducted with predicted value as independent variable and actual difficulty value as dependent variable, and the results were shown in table 3. The regression results showed that the model fitted well and the predicted value could explain 89.3% of the total variation of text difficulty.

TABLE III. REGRESSION ANALYSIS OF MODEL PREDICTION VALUE ON TEXT DIFFICULTY

	R2	Adjusted R2	F Variation	Standardization Beta	Value of t
Predictive Value	0.895	0.893	376.818***	0.946	19.412**

Note :* means $p < 0.05$, ** means $p < 0.01$, *** means $p < 0.001$

IV. RESULTS ANALYSIS

Through experiments, we found that IOG index, number of most common words, number of high frequency words in textbooks, number of complex words, notional word density, polysemous words and lowly transparent words are the most effective indexes affecting the text difficulty. We can find that:

(1) lexical diversity affects the readability of texts. With the increase of text difficulty level, the more diverse the lexical types are, the more diverse the use of word meanings is.

It is proved that IOG index and number of polysemy are effective indexes to predict lexical diversity of text readability. According to the IOG index, the higher the difficulty level of the text, the richer the words in the text and the more diverse the expressions. From the use of different meanings of polysemous words, with the increase of text difficulty level, the use of meaning is more. For example, the locational noun "上" has two main meanings in the Modern Chinese Dictionary. One is to express the specific position, and the other is to extend to time, scope and rank, which is more obscure than the meaning of the first one. In the primary difficulty level, it generally refers to "1 position word, used after a noun, means on the surface of an object", while in the higher difficulty level, it is more inclined to use "2 position word, used after a noun, means within the scope of something", which is a more abstract meaning, and the use of sense 2 obviously increases.

(2) word frequency profile affects the readability of text, and the number of most common words increases with the increase of text difficulty level.

The topic of low difficulty text is limited, the vocabulary in the text is also limited, the coverage of most common words is low. In the middle and high stage, the text content is more abundant, including narrative, exposition, argumentation and lyric prose and other genres. These contents make the vocabulary to organize the text structure and express the feelings of the text more diverse, the total text vocabulary has increased significantly, and the coverage of the most common words also increases.

(3) lexical density affects the readability of a text. As the difficulty level of a text increases, the number of notional words increases, while the content word ratio decreases.

By comparing the distribution of parts of speech in different levels of corpus, we find that nouns, verbs, adjectives, auxiliary words and adverbs are the most commonly used five parts of speech in all levels of texts, among which nouns, verbs, adjectives and other notional words have the largest number and the largest proportion. Compared with low-difficulty texts, the proportion of notional words in high-difficulty texts is decreasing. This may be related to the theme setting of the text. The content of primary difficulty text is mainly fairy tales, fables and stories. The language of these subjects is popular, simple and vivid. Words are often presented by notional words with rich meanings. Therefore, notional words expressing real meanings such as things, actions, behaviors, places and time occupy an absolute advantage in primary difficulty text. There are fewer functional words with legal meaning or function, mainly "的", "了", "着", "都" and other commonly used functional words. Highly difficult texts are richer in content, including narrative, expository, argumentative and Lyric prose. These texts increase the proportion of functional words which bear the functions of organizing text structure, expressing words and logical relations between words and sentences, while the proportion of notional words decreases accordingly.

(4) the transparency of words affects the readability of text. With the increase of text difficulty level, the frequency of words with low transparency becomes more prominent.

Semantic transparency plays an important role in compound word processing. Previous studies have found the semantic transparency effect, that is, transparent words are processed faster than opaque words. In the single-word priming condition, compared with opaque words, words with high transparency have shorter vocabulary judgment time. Morpheme information of words with high transparency promotes the processing of whole words, while morpheme information with low transparency hinders the processing of whole words(Wang Chunmao et al. ,1999). Semantic transparency is an important factor in the process of Chinese vocabulary cognition. By examining the transparency of compound words in textbooks, we find that, compared with low-difficulty texts, the proportion of words with low transparency in high-difficulty texts increases gradually and the comprehension difficulty increases.

V. CONCLUSION

In this paper, Chinese textbooks for grade 1 to 6 of primary schools published by People's Education Press are taken as data sets, and the texts are divided into 12 difficulty

levels successively. The effective lexical indexes to measure the readability of texts are discussed, and a regression model to effectively measure the lexical difficulty of Chinese texts is established. The study firstly collected 30 indexes at the text lexical level from the two dimensions of lexical richness and semantic transparency, selected the 7 indexes with the highest relevance to the text difficulty through Person correlation coefficient, and finally constructed a Regression to predict the text difficulty based on Lasso Regression, ElasticNet, Ridge Regression and other algorithms. The regression results show that the model fits well, and the predicted value could explain 89.3% of the total variation of text difficulty, which proves that the quantitative index of vocabulary difficulty of Chinese text constructed in this paper is effective, and can be applied to Chinese grade reading and computer automatic grading of Chinese text difficulty.

Based on the analysis and summary of the experimental results, we believe that the text vocabulary difficulty regression in this paper still needs to be improved in the following aspects :(1) the number of samples is small, so it is necessary to increase the number of samples to alleviate the problem of data sparsity and improve the prediction effect of text vocabulary difficulty; (2) in this paper, the lexical level features that affect the text difficulty are selected, which are completely based on statistical methods. The lexical difficulty of the text is the result of the interaction of multiple factors, and some interventions will be carried out in combination with rules in later work.

REFERENCES

- [1] Chall J S , Dale E.Readability revisited:the new Dale-Chall readability formula[J].Brookline Books, 1995:149.
- [2] Colins-Thompson K, Calan J P.A language modeling approach to predicting reading difficulty[C]//Hu- man Language Technologies:the 2004 Conference of the North American Chapter of the Asociation for Computational Linguistics. Asociation for Computational Linguistics, 2004: 193-200.
- [3] East, Martin. The impact of bilingual dictionaries on lexical sophistication and lexical accuracy in tests of L2 writing proficiency: A quantitative analysis. Assessing Writing, 2006, 11(3) .
- [4] Engber, Cheryl A. The relationship of lexical proficiency to the quality of ESL compositions. Journal of Second Language Writing, 1995, 4 (2) .
- [5] Fan, Fengxiang. A quantitative study on the lexical change of American English. Journal of Quantitative Linguistics, 2012, 19(3) .
- [6] Fiona.J Tweed & Baayen, R Harald. How variables may a constant be Measures in lexical richness in perspective. Computer and the Humanities, 1998, 32(4) .
- [7] Flor M, Klebanov B B, Shehan K M.Lexical tight- ness and text complexity[C]//The Workshop on Nat- ural Language Procesing for Improving Textual Accesibility, 2013:29-38.
- [8] Gilliland,J. Readability[M]. University of London Press for the United Kingdom Reading Association. 1972.
- [9] Heilman M, Colins-Thompson K, Eskenazi M.An analysis of statistical models and features for reading difficulty prediction[C]//Proceedings of the Work- shop on Innovative Use of Nlp for Building Educational Applications, 2018:71-79.
- [10] Petersen S E, Ostendorf M.A machine learning approach to reading level asesment[J].Computer Spech & Language, 2009, 23(1):89-106.

- [11] Pierre, J. A. Rare words, complex lexical units and the advanced learner[A]. In Coady and Huckin, ed.,Second Language Acquisition[C].Shanghai,2001.
- [12] Vogel M, Washburne C. An objective method of determining grade placement of children's reading material[J].Elementary School Journal, 1928, 28(5):373- 381.
- [13] Chen alin, Zhangsu. On the Difficulty Model of Chinese Text Reading and Readability Formula[J]. Computer Science,1999(11):42-44+27.
- [14] Guo Wanghao. Rearch on Readability Formula of Chinese Text for Foreign Students[D]. Shanghai Jiao Tong University, 2010.
- [15] Jing Xiyu. Study on Readability of Chinese Textbooks: Estimation of Grade Value[J]. Educational Research Information.1995, (05): pp113-127.
- [16] Li Yongkang. Researches in Definition of Words-Difficulty in Second Language[J]. Journal of Anhui University of Technology(Social Sciences) . 2003,20(05):pp122.
- [17] Liu Xiao. A Review of Text Readability Studies[J]. Journal of Hubei University(Philosophy and Social Science) ,2015,42(03):141-146.
- [18] Liang Maocheng. What is Corpus Linguistics[M]. Shanghai: Shanghai Foreign Language Education Press, 2016.
- [19] Sun Hanyin. Chinese intelligibility formula[D]. Beijing: Beijing Normal University.1992.
- [20] Sun Gang. Research on Prediction Method of Chinese Text Readability Based on Linear Regression[D]. Nanjing University, 2015.
- [21] Wang Yixuan. The Correlation between Lexical Richness and Writing Score of CSL Learner -- the Multivariable Linear Regression Model and Equation of Writing Quality[J]. Applied Linguistics, 2017(02): 93-101.
- [22] Wang Lei. Some Concepts of Readability Formula and Relevant Research Paradigm as well as the Research Tasks of Formula in TCFL[J]. Language Teaching and Linguistic Studies.2008,(06):pp50-57.
- [23] Wang Chunmao, Peng Danling, The Roles of Surface Frequencies, Cumulative Morpheme Frequencies, and Semantic Transparencies in the Processing of Compound Words[J]. Acta Psychologica Sinica, 1999(03):266-273.
- [24] Wu Siyuan, Cai Jianyong, Yu Dong, Jiang Xin. A Survey on the Automatic Text Readability Measures[J]. Journal of Chinese Information Processing, 2018,32(12):1-10.
- [25] Xu Caihua, Litang. The Role of Semantic Transparency on Word Recognition and Reading Comprehension:An Experimental Study on Children[J]. Applied Linguistics,2001(01):53-59.
- [26] Xu Jianhua, Liang Haoguang. Study on The Current Conditions and Countermeasures of Grade Reading Guide in Children's Library[J]. Library Tribune,2011, 31(06):247-252.
- [27] Zhang Biyin. Reading Psychology[M]. Beijing: Beijing Normal University Press.1992.
- [28] Zhang Yan, Chen Jiliang. Quantitative measuring approach of lexical richness in speech production[J]. Foreign Language Testing and Teaching, 2012(03): 34-40.
- [29] Zhao Shaohui. A Multi-dimension Perspective of Assessment of Lexical Competence[J]. TCSOL Studies, 2008(2).

An Measurement Method of Ancient Poetry Difficulty

for Adaptive Testing

Huiping Wang

Institute of Chinese Information Processing,
Beijing Normal University
Beijing, China
18013752306@163.com

Bihua Wang

Institute of Chinese Information Processing,
Beijing Normal University
Beijing, China
282843696@qq.com

Abstract—As traditional Chinese culture education grows fast, adaptive testing for ancient poetry seems promising. The prerequisite of adaptive learning is question bank, while the quality of question bank depends on the rationality of question difficulty. The paper proposes a method that measures ancient poetry difficulty for objective questions and builds a measurement system. The method includes five steps: (1) Identify the verses corresponding to the question. (2) Get four indexes of the corresponding verses: search volume, correctly answered frequency, sentence length and grade of the textbook which includes the verses. (3) Use analytic hierarchy process to index system for weight assignment. (4) Compute the weighted sum of the four indexes as the measurement of difficulty. (5) Classify the question bank according to the calculated difficulty. Experimental results showed the effectiveness of this measurement method, which thereby can be used in various adaptive testing of ancient poetry.

Keywords-Adaptive Testing; Ancient Poetry ; Difficulty

I. INTRODUCTION

Ancient poetry is a treasure of Chinese literature. As the most important cultural memory of the Chinese people, ancient poetry has always attracted the attention of researchers. With the rise of the "fever of traditional Chinese learning", the education of ancient poetry has been paid more and more attention.

Previous studies on ancient poetry learning focused on learning content [1-2], learning methods [3-6], and learning strategies [5-6]. There was relatively little research on ability evaluation. Meng Qi [7] analyzed the examination questions of ancient poetry in middle school entrance examination from both quantitative and qualitative dimensions. It was found that the main reason for the low validity of the examination papers was that there was no effective difficulty arrangement, which made the examination questions simpler. Ren Xueyun [8] analyzed the characteristics of poetry reading test questions in Shanghai College Entrance Examination, summarized the problems existing in the test questions, and proposed that the test questions should reflect the prospects of individual cultural accomplishment. These studies focus on the assessment of ancient poetry competence in primary and secondary schools, ignoring the test needs of higher stage learners. Adaptive testing for ancient poetry may be a good way to meet the needs of different level poetry learners.

Adaptive testing is a personalized testing method based on Item Reflection Theory(IRT), which can provide subjects with test questions that are suitable for their ability level, and truly achieve the goal of "testing according to people". The current application of adaptive testing is mostly embodied in foreign language proficiency testing. In

the adaptive test of foreign language proficiency, the subjects will complete the test questions of phonetic notation, word filling, etc. The final answer data can be used to update the difficulty value of the question bank so as to better serve the next adaptive test, thus forming a virtuous circle. Looking back at the self-adaptive test of ancient poetry, the predecessors have not attempted to build the ancient poetry question bank and quantify the difficulty of the test questions, let alone have constantly updated the answer data to predict the difficulty of the test questions.

The most important part of the IRT-based adaptive test system for ancient poetry ability is the construction of question bank. Whether the question bank system is good or not depends to a great extent on whether the quantification of the difficulty of a poem is reasonable. In the process of adaptive testing, the computer adaptive test (CAT) system acts as the "decision maker". Whatever strategy it adopts (such as maximum information method, hierarchical method, random method, etc.), its ultimate purpose is to select the exam questions with specific difficulty value matching the ability of the subjects from the question bank system. These questions are valuable for the subjects. Apart from being decisive in the selection of questions, the difficulty value of a poem also plays a crucial role in the subsequent estimation of the poem ability of the subjects. It directly affects the final ability level of the subjects and the reliability of the test. Therefore, how to quantify the ancient poetry difficulty for objective questions and grade the difficulty of the test questions to meet the practical needs of the Adaptive Testing of Ancient Poetry is a challenge.

The difficulty measurement of objective test questions of ancient poetry can be divided into two questions: the choice of difficulty quantification indexes and the assignment of weight. At present, domestic and overseas scholars seldom involve in quantitative research on the difficulty of ancient poetry. Professor Wang Zhaopeng of Wuhan University [9] launched the Tang Poetry Ranking List and Song Ci Ranking List. By means of quantitative scientific research, four kinds of statistical indicators were collected to measure the attention and influence of the poems: selected editions of past dynasties, commentaries of past dynasties, contemporary research and selection of literary history works. After reviewing Mr. Wang Zhaopeng's research, we think that there are some problems in his research, such as too strong subjectivity of indicators, unreasonable assignment of weights and incomplete measurement indicators. Specifically, it is unavoidable and impersonal for the subjectivity of selected editions, commentaries, contemporary studies and literary history works. Secondly, the rationality of assigning equal weights to the four indicators remains to be discussed. Thirdly, these

four indicators should be strictly classified as one indicator, and the indicators are not complete.

Defects in difficulty calculation of ancient poetry test questions will inevitably lead to inaccuracy of the adaptive test system for ancient poetry in topic selection and ability evaluation, and its practical value will be greatly reduced. This paper designs a difficulty quantification system and proposes a method for calculating the difficulty of ancient poetry test questions. In the process of calculating difficulty, we choose objective and multi-dimensional quantitative indicators, and allocate the weight of indicators reasonably, so that the calculation results are more scientific and practical.

II. QUANTITATIVE SYSTEM OF DIFFICULTY IN ANCIENT POETRY

The paper proposes a method that measures ancient poetry difficulty for objective questions and builds a measurement system. The structure of the system is shown in Figure 1, which includes question bank module, quantitative indicators module, information acquisition module, difficulty calculation module and difficulty classification module. Among them, (1)the question bank module includes many kinds of objective questions of ancient poetry; (2)the quantitative indicators module is used to select the quantitative indicators of a poem difficulty (3)the information acquisition module is used to identify the verses corresponding to the question, and obtain the information of search volume, correctly answered frequency, sentence length and grade of the textbook which includes the verses; (4)the difficulty calculation module is used to calculate the difficulty of the test questions; (5)the difficulty classification module can classify the question bank according to the calculated difficulty.

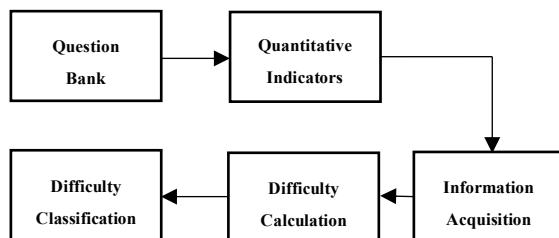


Figure 1. Quantitative System of Difficulty in Ancient Poetry.

A. Question Bank Module

In this paper, the ancient poetry question bank is derived from the previous research results of intelligent proposition of ancient poetry. The question bank contains selected labels, selected sentences, sorted sentences, selected style and crossword questions, as shown in the following examples:

- Selected Labels:

Select the following poem labels: “看朱成碧思纷纷，憔悴支离为忆君。不信比来长下泪，开箱验取石榴裙。”

- A.思念 B.惜时 C.爱国 D.离别

- Selection Style:

Select the following poetic style: “兰叶春葳蕤，桂华秋皎洁。欣欣此生意，自尔为佳节。谁知林栖者，闻风坐相悦。草木有本心，何求美人折?”

- | | |
|--------|--------|
| A.五言古诗 | B.五言律诗 |
| C.五言绝句 | D.五言乐府 |

- Selected Sentence:

Choose the next sentence of the following poems: “感时花溅泪，_____。”

- | | |
|---------|---------|
| A.恨别鸟惊心 | B.昔有鵩鵼心 |
| C.昔有鵩鵼心 | D.冥冥鸟去迟 |

- Sorted Sentences:

Choose the correct order of the following poems: “岱宗夫如何 齐鲁青未了 造化钟神秀 阴阳割昏晓。”

- A.1234 B.2314 C.1324 D.2413

- Crossword questions:

Choose the words to fill in the blank: “兰__春葳蕤，桂华秋皎洁。”

- A.叶 B.花 C.树 D.枝

B. Quantitative Indicators Module

The difficulty of a poem depends on many factors, such as: language, rhythm, genre, rhetoric, allusion, rhythm and ideological value. These factors can be summarized as internal factors; such as: attention, awareness, influence, transmission, etc. These factors can be summarized as external factors. This paper focuses on the difficulty of the poem in the CAT question bank. We believe that this difficulty can be understood as the tester's familiarity with poetry. In the face of familiar ancient poetry test questions, the probability of the tester to answer it is far greater than that of unfamiliar poem. For example, Li Bai's <蜀道难> and the Book of Songs <摽有梅>, from the internal factors such as metrics, genre, allusions, language, etc., the former is more difficult than the latter. But because the former is more familiar to the public, its attention, awareness and influence are far greater than the latter. In the adaptive test, the difficulty of <蜀道难> is less than that of <摽有梅>.

How can we effectively and scientifically obtain the public's familiarity with a poem through surveys and statistics? We then transform the abstract concept of familiarity into the influence of poem based on the degree of attention and dissemination. The higher the degree of attention and dissemination of the poem, the greater its influence, and the higher the public's familiarity with the poem. Based on the big data, we define the following indexes to measure the difficulty of poem:

- 1) Internet Search Volume

Nowadays, the Internet is an important medium for the dissemination of ancient poetry. It is also an important way for ordinary readers to understand, read and comment on ancient poetry. The more transcribed by the web page, the more popular the poem is and the more attention it receives. Therefore, we can use the link rate of the network to measure the attention of different works in the general readership.

- 2) Frequency of Occurrence

Frequency of occurrence can be regarded as a statistical feature of the difficulty of the poem, that is, the number of repeated occurrences of the poem in a certain number of real corpus. It determines the degree of common use of the poem

and the degree of familiarity of poetry learners, so it is an important factor to quantify the difficulty of a poem. Generally speaking, the higher the frequency of the use of a poem, the higher the public's daily perception, the less difficult it will be in memory.

3) Single Sentence Length

The difficulty of memorizing often affects the judgment of difficulty. The easy-to-remember sentences are easy to pass from mouth to ear. Therefore, the length of sentences is regarded as one of the difficulty indicators of quantitative topics in this paper.

4) Textbook Appearance Level

Using the above three indicators, we can roughly get the ideal difficulty of ancient poetry, but as mentioned above, this study focuses on the difficulty of the poem in the adaptive test bank. Most of the people's poetry reserve comes from textbooks of primary and secondary schools. In view of this situation, this paper regards the distribution level of ancient poetry in textbooks as a penalty factor for difficulty discrimination. The lower the grade in textbooks, the greater the difficulty value.

C. Information Acquisition Module

1) Internet Search Volume (S)

Internet Search Volume is obtained by using the precise search and fuzzy search mode of Baidu search engine. There are two steps to obtain it:

Step 1, the question is searched as a keyword, and the result is a fuzzy search quantity.

Step 2, search<"question ">as a key word, and the result is precise search quantity.

Step 3, the sum of the results of the two modes is the internet search volume of the poem.

2) Frequency of Occurrence (F)

The frequency of inscription stems comes from the frequency of visits in the knowledge map of ancient poetry constructed in the earlier period. The higher the frequency, the more familiar the public is with it, the simpler the title is.

3) Single Sentence Length(L)

The length of a single sentence is to calculate the number of Chinese characters in a given sentence. For example, the single sentence of the poem "两只黄鹂鸣翠柳，一行白鹭上青天" is 7.

4) Grade of Textbook Distribution(R*)

We compiled the poems included in the different versions of elementary, middle, and high school textbooks, and used the distribution level of the questions in the textbook as the penalty factor for difficulty discrimination. The primary school textbook is marked as 12-7 by grade, 6-4 in junior high school, 3-1 in high school, and 0 in non-existing.

D. Difficulty Calculation Module

In this study, the indexes that affect the difficulty of a poem are Internet search volume, frequency, sentence length and textbook grade. The four kinds of indexes have different influence on the determination of a poem difficulty. Therefore, we use linear weighted summation method to preliminarily define the formula for calculating the difficulty of ancient poetry titles (1):

$$* b = S \times W_1 + F \times W_2 + L \times W_3 + R * \quad (1)$$

Among them, the internet search volume (S) and the frequency of occurrence(F) are negatively correlated with the difficulty value; the single sentence length(L) and the grade of textbook distribution are positively correlated with the difficulty value.

W_1 、 W_2 、 W_3 are the weights of S、F、and L respectively. How to determine the weight of data, there are no precedents in literary measurement research, but in bibliometrics, there are two ways of subjective weighting and objective weighting. Among them, the subjective weighting method is mainly from the perspective of qualitative analysis, according to the size of each indicator to determine the impact of the object to determine the weight of the corresponding indicators, but also can be divided into expert evaluation method and analytic hierarchy. Objective weighting mainly calculates the weight of each index from the perspective of quantitative analysis, and there are two methods of coefficient of variation and correlation coefficient. Considering the specific situation of literary research, we choose the analytic hierarchy process(AHP) in subjective weighting [8] to determine. The basic steps of the analytic hierarchy process are as follows:

Step 1, we invited 10 experts in this field to grade the importance of the indexes and construct the judgment matrix as follows:

$$A = \begin{bmatrix} 1 & 3 & 5 \\ 1 & 1 & 4 \\ 3 & 1 & 1 \\ 1 & 4 & 1 \end{bmatrix}$$

Step 2, according to the matrix, W_1 , W_2 and W_3 are computed as 0.62, 0.28 and 0.10 respectively.

Step 3, we tested the consistency of the three weights and obtained the consistency ratio $0.075 < 0.1$, which shows that the weights we got are effective and reliable.

On this basis, considering the numerical differences among internet search volume, frequency of occurrence and single sentence length, we get the final formula for determining the difficulty of ancient poetry (2):

$$* b = -\log_{100}(S + 0.001) * 0.62 - \log_{100}(F + 0.001) * 0.28 + L * 0.1 - R * \quad (2)$$

It is worth pointing out that this difficulty is aimed at self-adapting test of the difficulty of a poem. It considers the attention and influence of the relevant poems, and cannot be used to evaluate the artistic and ideological value of each poem. Poems with low difficulty value can only be said to have greater influence in the long process of communication and acceptance, which does not mean that they have little artistic value and little ideological significance.

E. Difficulty Classification Module

In order to improve the applicability of the question bank to various test methods, we calculate the difficulty of all the questions according to the difficulty calculation formula defined above, and classify the question bank. The graded question bank can also be used for poetry test of other modes (such as breaking through customs), and can also be decomposed into Test Bank of different levels of ancient poetry level test.

1) Ancient Poetry Pass-through Game

The whole ancient poetry question bank is divided into 60 levels. According to the grade of textbook distribution(R^*), the test questions with non-zero textbook grades (corresponding to the ancient poems in the textbook) are arranged in ascending order of difficulty and then data is divided into 15 levels, representing low difficulty; the test questions marked with a textbook grade of 0 (corresponding to the textbook's external poems) are arranged in ascending order of difficulty and then data is divided into 45 levels, representing moderate difficulty.

The flow chart of the game is shown in Figure 2.

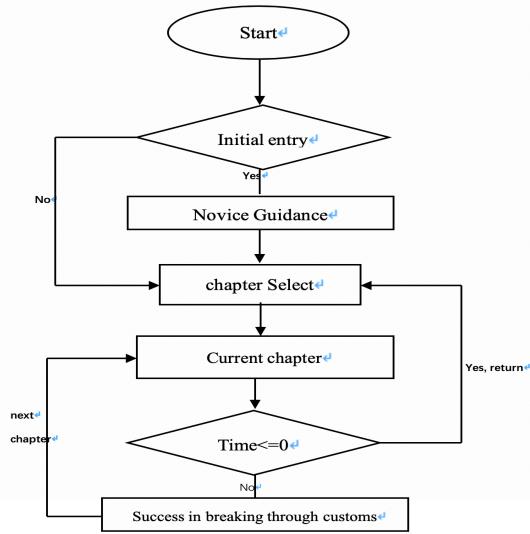


Figure 2. Ancient Poetry Pass-through Game.

2) Ancient Poetry Ability Test

The entire ancient poetry question bank is divided into 25 grades. The overall difficulty level of the first 15 grades is low. Within the scope of textbooks in primary and middle schools, the number of questions and the number of ancient poems involved in each level are small, and the applicable population is wide. The last 10 levels are medium to high difficulty, and the number of questions and the number of ancient poems per level are large, aiming at refining the ancient poetry of middle and high level people. The flow chart of the ancient poetry ability test is shown in Figure 3.

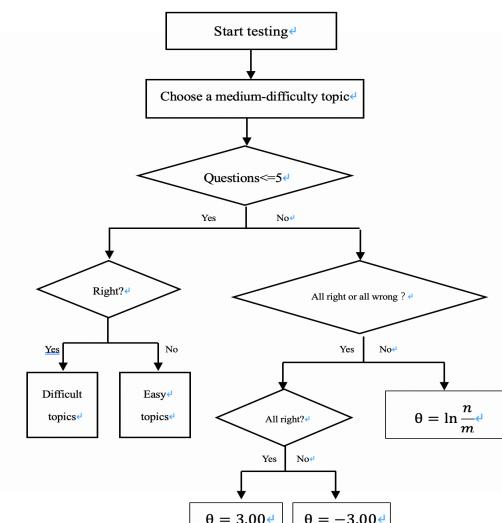


Figure 3. Ancient Poetry Ability Test.

III. VALIDATION OF DIFFICULTY EFFECTIVENESS

We use the following methods to evaluate the effectiveness of the difficulty calculation of the question stem:

1) 100 questions were randomly extracted from the question bank, and the difficulty of these 100 questions was calculated as the sample X to be tested.

2) Using min-max standardization to normalize the difficulty of questions to 1-10, the calculation formula of min-max standardization is as follows:

$$f(b)_{\text{scaled}} = \frac{b - b_{\min}}{b_{\max} - b_{\min}} \cdot (\max - \min) + \min \quad (3)$$

Among them, b_{\max} and b_{\min} correspond to the maximum and minimum difficulty values calculated in the whole question bank respectively, \max and \min correspond to the maximum and minimum values after normalization respectively, where 10 and 1 are taken respectively.

3) Twenty postgraduates majoring in Chinese were asked to grade the difficulty of 100 questions, with the score limited to 1-10, and the average of 20 scoring tables was taken as the final target sample Y.

4) The Pearson correlation coefficient is used to measure the correlation between the tested sample X and the target sample Y. The Pearson formula selected in the present invention is as follows:

$$\rho_{X,Y} = \frac{\sum(X-\bar{X})(Y-\bar{Y})}{\sqrt{\sum(X-\bar{X})^2 \sum(Y-\bar{Y})^2}} \quad (4)$$

Generally, the correlation strength of the judgment variable can be based on Table 1.

TABLE I. PEARSON CORRELATION COEFFICIENT MEANING TABLE

Coefficient	Correlations
0.8-1.0	Extremely strong
0.6-0.8	Strong
0.4-0.6	Intermediate
0.2-0.4	Weak
0.0-0.2	Extremely weak

Finally, the correlation coefficient between the test sample X and the target sample Y is 0.862 (three decimal places are reserved), and the difficulty calculated by the difficulty quantization method is highly correlated with the difficulty judged by the professional. It can be proved that the method for quantifying the difficulty of the ancient poetry proposed by the present invention is reasonable.

IV. CONCLUSION

This paper proposes a method to quantify the difficulty of a poem, which gives a score of difficulty for each ancient poetry in the adaptive test bank. We chose Internet search volume, frequency of occurrence and single sentence length as the quantitative criteria of difficulty, and assign the weight for each criteria with analytic hierarchy process (AHP). At the same time, we used the distribution of textbooks as a penalty to improve the effectiveness of system. Furthermore, we divided the difficulty-calculated question bank into several levels so that it can be adapted to more test forms. In order to validate method, we selected subjects with different educational levels to conduct

experiments. The results show that the difficulty of ancient poetry calculated by our method is reasonable and feasible. Of course, there are still many shortcomings in our research, such as more quantitative indicators are not considered, the level of subjects needs to be refined and so on. We expect that these shortcomings will be better remedied in the future work.

REFERENCES

- [1] Z. Ting, "A Study on the Joining Points of Ancient Poetry in Textbooks of Grade Four to Seven of Shanghai Educational Press", Shanghai Normal University, 2018.
- [2] X. Kexiong, "What Should Students Learn in Ancient Poetry Teaching", Inner Mongolia Education, no. 10, pp. 273-297, 2010.
- [3] C. Ximmin, "Exploration of Learning Methods for Ancient Poetry Literature", China Educational Technology & Equipment, no. 27, pp. 63-63, 2010.
- [4] W. Xianwen, "Teaching Practice of Five-step Learning Method of Ancient Poetry", Bulletin of Chinese Language Teaching, no. 6, pp. 11-11, 2001.
- [5] C. Jianwei, "Research on Teaching Strategies of Chinese Ancient Poetry in Primary Schools under the Background of New Curriculum Reform", Education for Chinese After-school(Theory), no. 1, 2014.
- [6] C. Luyi, "The High School Language Ancient Poetry Teaching Innovation Method Research", Hunan Normal University, 2013.
- [7] M. Qi, "Research on the Testing Objectives of the Poem Test Question in Shanghai Entrance Examination for Senior Middle School", Shanghai Normal University, 2015.
- [8] W. Rufei, "An Analysis of Strategies in Teaching Chinese Ancient Poetry and Ci in Junior Middle School", China Educational Technology & Equipment, no. 27, pp. 63-63, 2010.
- [9] W. Zhaopeng, *Ranking List of Tang Poetry*. Shanghai,CN: Shanghai Classics Publishing House, 2011.
- [10] L. Peng, "Research on Several Key Technologies of Computer Adaptive Testing", Dongbei Normal University, 2012.
- [11] L. Liping, W. Wenjie and G. Shining, "Design and Realization of the Item Pool System for CAT", Computer Systems & Applications, vol. 22, no. 9, pp. 10-12, 2006.
- [12] L. Zhaosheng, *Theoretical Basis of Item Response*. Beijing,CN: Beijing Normal University Publishing Group, 2012.
- [13] H. Yueh-Min, L. Yen-Ting and C. Shu-Chen, "An Adaptive Testing System for Supporting Versatile Educational Assessment", Computers & Education, vol. 52, no. 1, pp. 53-67, 2008.
- [14] M. Barla, M. Bieliková, A. Bou Ezzeddinne, "On the impact of adaptive test question selection for learning efficiency", Computers & Education, vol. 52, no. 1, pp. 53-67, 2008.

Prosodic Realization of Focus in Changchun Mandarin and Nanjing Mandarin

Ying Chen*, Jiajing Zhang, Bingying Ye and Chenfang Zhou

School of Foreign Studies, Nanjing University of Science and Technology
Nanjing, China

*E-mail: ychen@njust.edu.cn

E-mail: iris_zhang@njust.edu.cn

E-mail: yby2006lrx@126.com

E-mail: zhouchenfang_19@163.com

Abstract— This study was designed to explore the prosodic patterns of focus in two dialects of Mandarin. One is Changchun Mandarin and the other is Nanjing Mandarin. The current paper compares the acoustics of their prosodic realization of focus in a production experiment. Similar to standard Mandarin, which uses in-focus expansion and concomitantly post-focus compression (PFC) to code focus, results in the current study indicate that both Changchun and Nanjing speakers produced significant in-focus expansion of pitch, intensity and duration and PFC of pitch and intensity in their Mandarin dialects. Meanwhile, the results show no significant difference of prosodic changes between Changchun and Nanjing Mandarin productions. These results reveal that PFC not only exists in standard Mandarin but also in Mandarin dialects.

Keywords-prosodic focus; post-focus compression; Changchun Mandarin; Nanjing Mandarin

I. INTRODUCTION

Prosody refers to the suprasegmental features in an utterance. Its acoustic parameters include duration, pitch (F0), intensity, etc. Focus is used to highlight a certain part of an utterance in a certain context, which in many cases, in addition to morphosyntactic means, can be realized by means of prosody [24]. Therefore, prosodic change plays an essential role in focus realization. Reference [21] proposed main features of prosodic change: an increase in F0, intensity, and duration in focused components, a decrease of F0 and intensity in post-focus components (post-focus compression), and no consistent prosodic change in pre-focus components. Post-focus compression (PFC) was found in many languages, including standard Mandarin [21] and English [20], whereas PFC does not exist in some languages, such as Southern Min [11, 24] and Cantonese [14].

References [16, 21, 22] found that standard Mandarin has PFC. Both Changchun Mandarin and Nanjing Mandarin are subdialects of Mandarin Chinese. The former belongs to Northeast Mandarin and the latter belongs to Jianghuai Mandarin. The two dialects are both tonal but with different tonal inventories. Standard Mandarin has four tones: T1 (55), T2 (35), T3 (214) and T4 (51). The numbers in parentheses represent tone values based on Chao's five-scale notation system [19]. The corresponding tones in Changchun Mandarin are T1 (44), T2 (24), T3 (214) and T4 (53) [25], and that in Nanjing Mandarin are T1 (31), T2 (13), T3 (212) and T4 (44) [26].

Based on previous findings in the research of prosodic focus, this study investigates whether PFC exists in these two Mandarin dialects as it exists in standard Mandarin and

how native speakers of these Mandarin dialects realize prosodic focus. Three research questions are addressed:

(1) Do Changchun and Nanjing Mandarin speakers produce significant in-focus expansion of duration, F0 and intensity in their dialects?

(2) Do they produce significant post-focus compression (PFC) of F0 and intensity in their dialects?

(3) Do the prosodic focus patterns in Changchun and Nanjing Mandarin production differ from one another?

II. METHODS

A. Participants

Ten Changchun (CC) Mandarin speakers and ten Nanjing (NJ) Mandarin speakers (five males and five females in each group) were recruited from Nanjing University of Science and Technology in China. They are all undergraduate students, age from 18 to 24, born and raised in either Changchun or Nanjing. They speak CC or NJ Mandarin as their first language.

B. Stimuli

The stimuli are listed in Table I and Table II. Participants were instructed to answer the same target sentences in their dialect, guided by different prompt questions. Previous studies found that PFC can be clearly observed in a sentence containing all sonorant segments given that all sonorants are tone-bearing unit [21, 24]. Therefore, each target sentence in this study uses three words and five syllables/characters with sonorant onsets which bear the same tone in standard Mandarin while corresponding to a different tone in CC and NJ Mandarins. Each syllable in the same target sentence in the same dialect is pronounced in the same tone. For instance, the Mandarin tone in the first sentence is Tone 1 but pronounced with tones in different dialects as shown in Table II. There are four target sentences in four tones corresponding to the T1 (55), T2 (35), T3 (214) and T4 (51) in standard Mandarin respectively. To obtain prosodic change of focus, different prompt questions were raised to elicit focus in different locations: initial focus (the subject), medial focus (the verb) and final focus (the object).

TABLE I. PROMPT QUESTIONS AND ANSWERS IN THE STIMULI

		Sentence 1	Sentence 2	Sentence 3	Sentence 4
Neutral focus	Q	ni shuo shenme? 'What did you say?'	ni shuo shenme? 'What did you say?'	ni shuo shenme? 'What did you say?'	ni shuo shenme? 'What did you say?'
	A	wuma mo maomi. <i>Mom Wu patted the kitty.</i>	laomo na maoni. <i>The model worker took the woolen.</i>	limei lou laoma. <i>Li Mei hugged Lao Ma.</i>	luli ma laile. <i>Lu Li scolded Lai Le.</i>
Initial focus	Q	shui mo maomi? 'Who patted the kitty?'	shui na maomi? 'Who took the woolen?'	shui lou laoma? 'Who hugged Lao Ma?'	shui ma laile? 'Who scolded Lai Le?'
	A	wuma mo maomi.	laomo na maoni.	limei lou laoma.	luli ma laile.
Medial focus	Q	wuma dui maomi zuo shenme? 'What did Mom Wu do to the kitty?'	laomo dui maoni zuo shenme? 'What did the model worker do to the woolen?'	limei dui laoma zuo shenme? 'What did Li Mei do to Lao Ma?'	luli dui laile zuo shenme? ?
	A	wuma mo maomi.	laomo na maoni.	limei lou laoma.	luli ma laile.
Final focus	Q	wuma mo shenme? 'What did Mom Wu pat?'	laomo na shenme? 'What did the model worker take?'	limei lou shui? 'Who did Li Mei hug?'	luli ma shui ? 'Who did Lu Li scold?'
	A	wuma mo maomi.	laomo na maoni.	limei lou laoma.	luli ma laile.

TABLE II. TARGET SENTENCES IN STANDARD, NANJING, CHANGCHUN MANDARINS

	Word 1	Word 2	Word 3
Character	邬妈	摸	猫咪
Gloss	'mom Wu'	'pat'	'kitty'
Standard	wu55 ma55	mo55	mao55 mi55
Nanjing	wu31 ma31	mo31	mao31 mi31
Changchun	wu44 ma44	mo44	mao44 mi44
Character	劳模	拿	毛呢
Gloss	'model worker'	'take'	'woolen'
Standard	lao35 mo35	na35	mao35 ni35
Nanjing	lao13 mo13	na13	mao13 ni13
Changchun	lao24 mo24	na24	mao24 ni24
Character	李美	搂	老马
Gloss	'Li Mei'	'hug'	'Lao Ma'
Standard	li214 mei214	lou214	lao214 ma214
Nanjing	li212 mei212	lou212	lao212 ma212
Changchun	li214 mei214	lou214	lao214 ma214
Character	陆丽	骂	赖乐
Gloss	'Lu Li'	'scold'	'Lai Le'
Standard	lu51 li51	ma51	lai51 le51
Nanjing	lu44 li44	ma44	lai44 le44
Changchun	lu53 li53	ma53	lai53 le53

C. Procesures

Recording took place in a soundproof booth at NJUST. The two experimenters were conducted by a native speaker of CC and NJ Mandarin respectively. Prompt questions were asked in a random order with five repetitions. The written answers were provided in Chinese characters. Participants were requested to respond to the questions with appropriate foci. A total of 80 target sentences were produced by each participant (4 tones \times 4 focus locations \times 5 repetitions). A Marantz professional solid state recorder PMD661 and a Shure professional unidirectional head-worn dynamic microphone SM10A-CN were used for recording. The stimuli were recorded with a sampling rate of 44,100HZ and saved in an SD card. The production experiment was self-paced and lasted approximately 30 minutes for each participant.

D. Analysis

Data were analyzed by Praat version 5.3.65 [8] and a Praat script—ProsodyPro version 5.5.2 [23]. The acoustic parameters in the current paper include mean duration, mean F0, mean intensity and time-normalized F0 at ten even-interval points in each syllable. The ten even-interval F0 values were extracted to track the time-normalized F0 trajectory for a direct observation of pitch change as a function of focus.

To examine the prosodic change of focus, the differential of duration, F0 and intensity values between the initial, medial, final foci and the neutral focus was calculated by subtracting the mean value of each component in the neutral-focus sentence from the corresponding mean value in the focused sentences (initial, medial, and final respectively). Note that the mean F0 differential was converted from Hertz to semitones [$st = 12 \log_2(F0)$, where reference level is 1 Hz] because pitch in speech operates on a logarithmic scale just as in music [5, 6].

Therefore, pre-focus change was calculated on, for instance, the mean value of the two syllables of "wuma" in the medial-focus sentences and the three syllables of "wuma mo" in the final-focus sentences minus their counterparts in the neutral-focus sentences. The in-focus change was computed on the prosodic values of the syllables of the focused words minus its neutral-focus counterparts. Post-focus change was calculated on the mean value of the three syllables of "mo maomi" in the initial-focus sentences and the two syllables of "maomi" in the medial-focus sentences minus that of the neutral-focus sentences.

III. RESULTS

A. Time-normalized F0 Contours

Figures 1-8 display the time-normalized F0 contours with three focus locations (initial, medial, final) vs. their neutral-focus counterparts by tone and dialect. Each curve represents an average of each target sentence of five repetitions produced by the ten speakers of each group. The focused sentences are represented by dash curves, while the neutral-focus sentences by solid curves. Syllable boundaries are signified by vertical lines.

Figures 1-4 display the time-normalized F0 contours in four tones in CC Mandarin produced by CC speakers. It reveals that they produced clear in-focus expansion in F0 in the three focus locations for four tones and post-focus

compression in initial and medial focus locations. The pre-focus changes are not as clear as in-focus and post-focus changes.

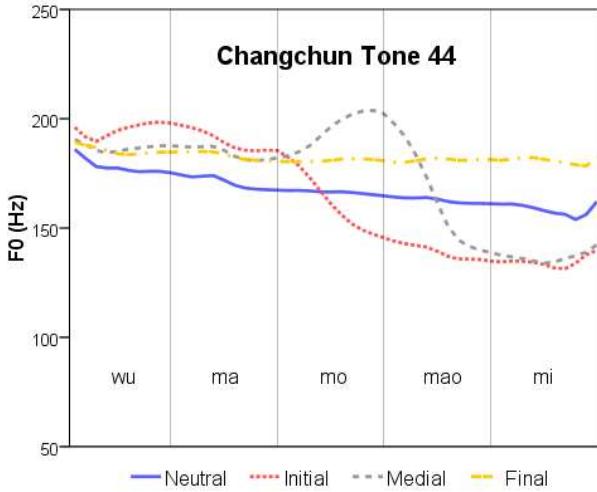


Figure 1. Time-normalized F0 contours (Hz) by Changchun speakers in Tone 44

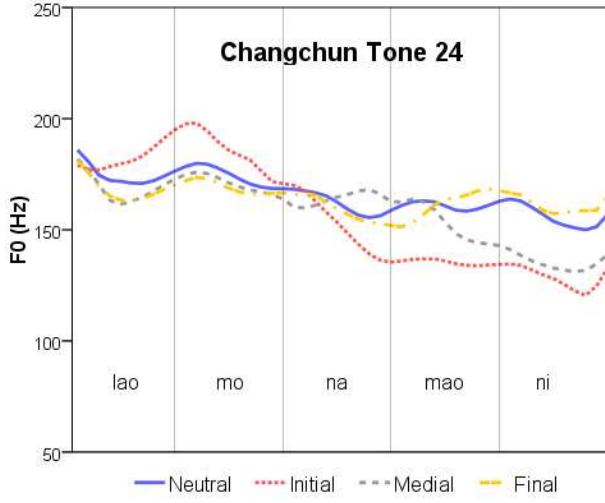


Figure 2. Time-normalized F0 contours (Hz) by Changchun speakers in Tone 24

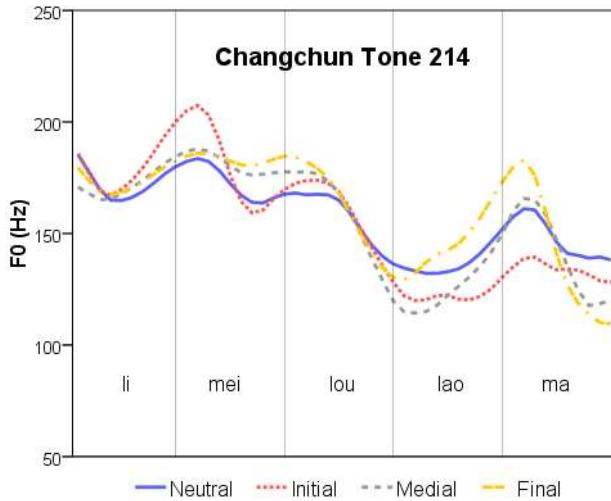


Figure 3. Time-normalized F0 contours (Hz) by Changchun speakers in Tone 214

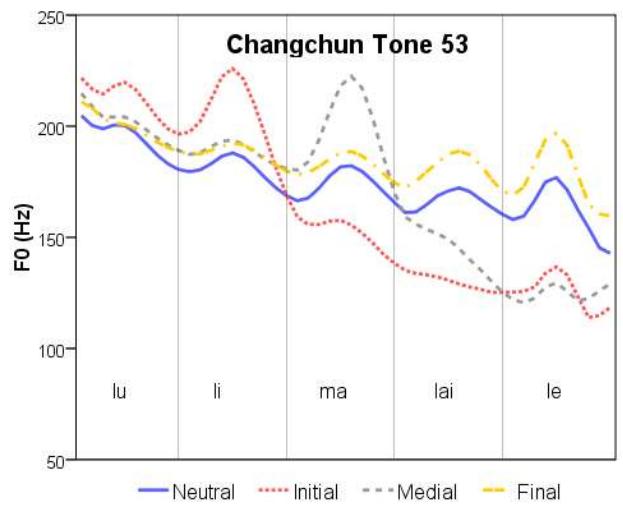


Figure 4. Time-normalized F0 contours (Hz) by Changchun speakers in Tone 53

Likewise, the in-focus expansion was quite noticeable in NJ Mandarin as shown in Figures 5-8. The F0 contours also show noticeable post-focus compression in the initial and medial focus locations with the four tones whereas unnoticeable change in pre-focus constituents.

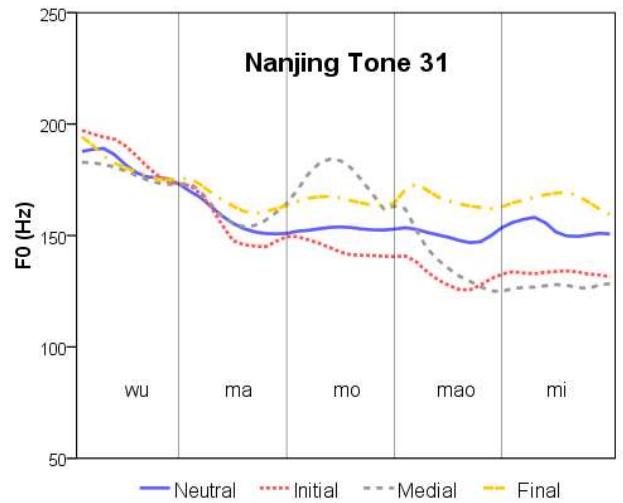


Figure 5. Time-normalized F0 contours (Hz) by Nanjing speakers in Tone 31

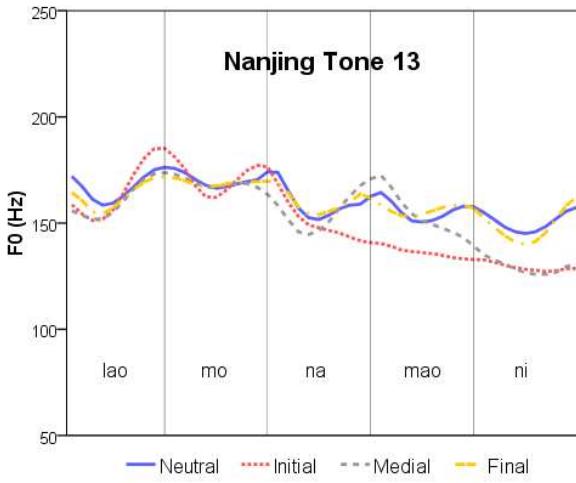


Figure 6. Time-normalized F0 contours (Hz) by Nanjing speakers in Tone 13

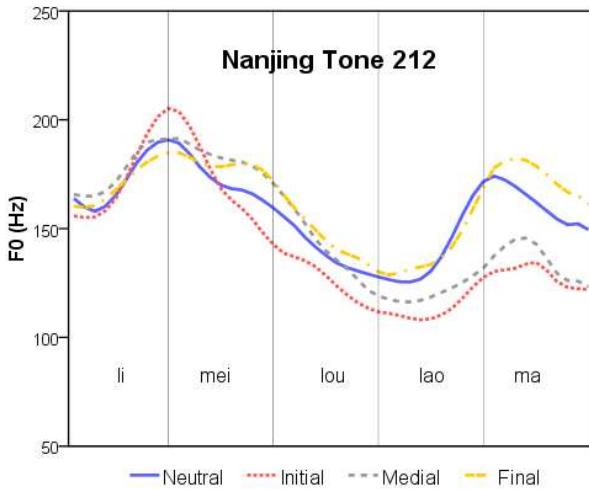


Figure 7. Time-normalized F0 contours (Hz) by Nanjing speakers in Tone 212

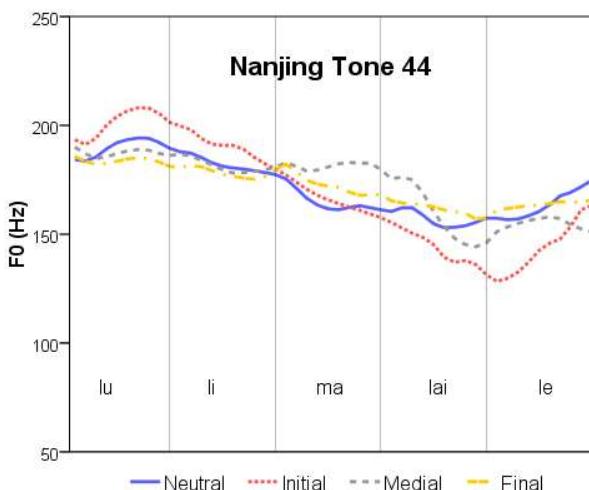


Figure 8. Time-normalized F0 contours (Hz) by Nanjing speakers in Tone 44

B. Focus Change in Mean F0, Intensity, and Duration

The time-normalized F0 contours illustrate consistent patterns of prosodic change as a function of focus regardless of tone. Therefore, focus changes were calculated collapsing tone types in the current paper. We applied repeated measures ANOVAs to analyze the mean F0, intensity, and duration change with a within-subjects factor—focus condition (three levels: pre-focus, in-focus, and post-focus) and a between-subject factor—speaker group (two levels: Changchun and Nanjing).

The ANOVA results of mean F0 change indicate no interaction between focus condition and speaker group and the main effect of the speaker group show no significance. The main effect of focus condition ($F(2, 36) = 84.181, p < 0.001$) was highly significant. The results demonstrate that there was no significant difference between CC and NJ groups. Further comparisons in paired-samples t -tests show significant differences between pre-focus change and post-focus change ($t(9) = 5.097, p = 0.001$) and between in-focus change and post-focus change ($t(9) = 7.238, p < 0.001$) in NJ Mandarin. Similar comparisons also show significant differences between pre-focus change and in-focus change ($t(9) = -4.628, p = 0.001$), between pre-focus change and post-focus change ($t(9) = 7.660, p < 0.001$), and between in-focus change and post-focus change ($t(9) = 8.635, p < 0.001$) in CC Mandarin.

To compare the degree of using F0 change to code focus at sentential level between CC and NJ speakers, the magnitude on F0 differentials was computed. It respectively subtracted pre-focus change and post-focus change from in-focus change and was examined by independent-samples t -tests. Results show no significant difference between CC and NJ speakers in F0 change magnitude between in-focus and pre-focus ($t(18) = -0.387, p = 0.703$) and between in-focus and post-focus ($t(18) = -0.612, p = 0.548$).

Figure 9 displays mean F0 change with standard error bars of the two speaker groups. CC and NJ speakers expanded F0 in focused components and compressed F0 in post-focus components (thus produced PFC) whereas no consistent change in pre-focus condition.

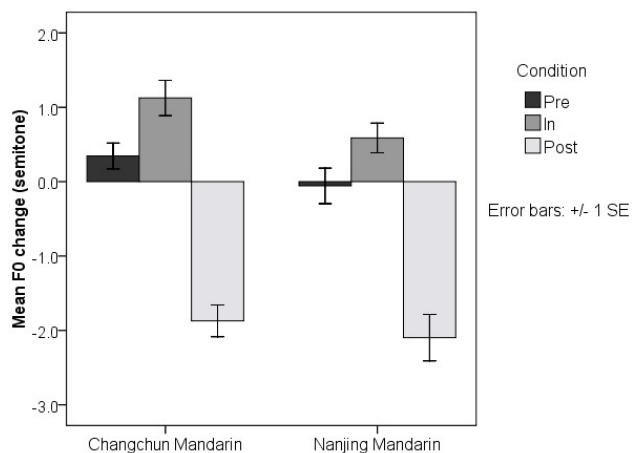


Figure 9. F0 change (semitone) by focus condition and speaker group

The results of repeated measures ANOVA on mean intensity change indicate no interaction between focus condition and speaker group, no main effect of speaker group and only main effect of focus condition ($F(2, 36) = 119.151, p < 0.001$). Further comparisons in paired-samples *t*-tests show significant differences between pre-focus change and in-focus change ($t(9) = -3.785, p = 0.004$), between pre-focus change and post-focus change ($t(9) = 7.259, p < 0.001$) and between in-focus change and post-focus change ($t(9) = 11.134, p < 0.001$) in NJ Mandarin. And significant differences between pre-focus change and in-focus change ($t(9) = -4.476, p = 0.002$), between pre-focus change and post-focus change ($t(9) = 5.582, p < 0.001$) and between in-focus change and post-focus change ($t(9) = 9.452, p < 0.001$) were found in CC Mandarin.

Independent-samples *t*-tests show no significant difference between CC and NJ speakers in intensity change magnitude between in-focus and pre-focus ($t(18) = -0.419, p = 0.680$) and between in-focus and post-focus ($t(18) = 1.174, p = 0.256$). Figure 10 reconfirms these results that CC and NJ speakers produced clear in-focus expansion and post-compression of intensity and there is no difference between CC and NJ.

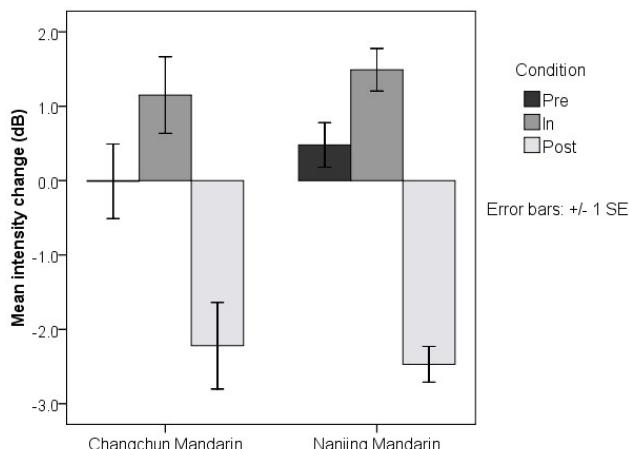


Figure 10. Intensity change (dB) by focus condition and speaker group

The results of repeated measures ANOVA on mean duration change demonstrate again only main effect of focus condition was significant ($F(1.40, 25.27) = 31.488, p < 0.001$) but no interaction between focus condition and speaker group and no main effect of group. Paired-samples *t*-tests show significance between pre-focus change and in-focus change ($t(9) = -3.298, p = 0.009$) and between in-focus change and post-focus change ($t(9) = 2.868, p = 0.019$) in NJ Mandarin, and significance between pre-focus change and in-focus change ($t(9) = -6.072, p < 0.001$) and between in-focus change and post-focus change ($t(9) = 6.203, p < 0.001$) in CC Mandarin.

Independent-samples *t*-tests also showed no significant difference between CC and NJ speakers in duration change magnitude between in-focus and pre-focus ($t(18) = -1.269, p = 0.221$) and between in-focus and post-focus ($t(18) = -1.037, p = 0.314$). As shown in Figure 11, CC and NJ speakers produced clear in-focus expansion in duration while there was no clear patterns of duration change in pre-focus and post-focus components.

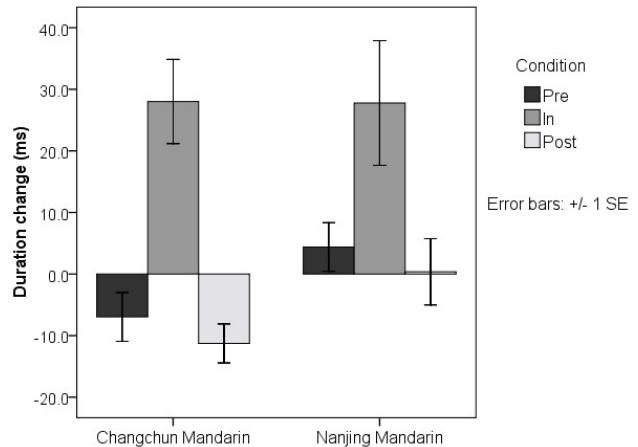


Figure 11. Duration change (ms) by focus condition and speaker group

IV. DISCUSSION

The results of the speech production experiment have answered the three research questions. For the first and second questions, both CC and NJ Mandarin speakers produced in-focus expansion of F0, intensity and duration and post-focus compression of F0 and intensity. However, they did not produce substantial pre-focus change in either CC Mandarin or NJ Mandarin. This finding is in line with that of [21] that the prosodic change in pre-focus components normally remains intact.

According to the results of ANOVAs and *t*-tests, there was no significant difference in the prosodic change of mean F0, intensity and duration as a function of focus between CC Mandarin and NJ Mandarin, which provides negative answers to the third research question. In addition, no difference of the magnitude of F0, intensity and duration between in-focus and pre-focus changes and between in-focus and post-focus changes in CC and NJ Mandarin productions suggests that both speaker groups use similar degree of acoustic variations to code focus in their Mandarin dialects though they differ in tonal inventory. This further suggests that prosodic variation is a means of coding focus at sentential level regardless of tones at lexical level.

V. CONCLUSIONS

This paper examines the prosodic realization of focus in Mandarin dialects by Changchun and Nanjing Mandarin speakers. Two main findings are discovered. First, both CC and NJ speakers produced in-focus expansion and post-focus compression (PFC) in the two Mandarin dialects. Second, CC and NJ speakers use similar prosodic pattern to code focus in these two dialects. We conclude that Changchun Mandarin and Nanjing Mandarin are PFC language.

PFC has been found not easy to transfer from one language to another [15, 16, 17], even highly proficient Chinese learners of English were not able to achieve native-like PFC in English and vice versa, future work may investigate the English production of bidialectal speakers of standard Mandarin and Changchun/Nanjing Mandarin for insight into whether one more PFC dialect in learners' L1 facilitates acquiring PFC in their L2.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation of China (61573187) and the NJUST Key Bachelor's Thesis Grant.

REFERENCES

- [1] B. Wang and Y. Xu, "Differential prosodic encoding of topic and focus in sentence-initial position in Mandarin Chinese," *Journal of Phonetics*, vol. 39, pp. 595-611, 2011.
- [2] C. Féry and F. Kügler, "Pitch accent scaling on given, new, and focused constituents in German," *Journal of Phonetics*, vol. 36, pp. 0-703, 2008.
- [3] C. T. Best and M. D. Tyler, "Nonnative and second-language speech perception: Commonalities and complementarities," Ed. *Language experience in second language speech learning: In honor of James Emil Flege*. Amsterdam: John Benjamins Publishing Company, pp. 13-34, 2007.
- [4] C. T. Best, "A Direct Realist View of Cross-Language Speech Perception," Ed. *Speech perception and linguistic experience: Issues in cross-language research*. Timonium, MD: York Press, pp. 171-204, 1995.
- [5] F. Nolan, "Intonational equivalence: an experimental evaluation of pitch scales," In *Proceedings of the 15th international congress of phonetic sciences*, Barcelona, Spain, vol. 771, pp. 774, 2003.
- [6] H. Fujisaki, "Prosody, information, and modeling-with emphasis on tonal features of speech," *Workshop on Spoken Language Processing*, pp. 5-14, 2003.
- [7] J. E. Flege, "Second-language speech learning: Theory, findings, and problems," Ed., *Speech perception and linguistic experience: Issues in cross-language research*. Timonium, MD: York Press, pp. 233-277, 1995.
- [8] P. Boersma and D. Weenink, "Praat: doing phonetics by computer. Version 5.3.65. Online: <http://www.fon.hum.uva.nl/praat/>," 2014.
- [9] R. Lado, "Linguistics across cultures: applied linguistics for Language teachers," University of Michigan Press, 1957.
- [10] R. Wardhaugh, "The contrastive analysis hypothesis," *TESOL quarterly*, pp. 123-130, 1970.
- [11] S. W. Chen, B. Wang and Y. Xu, "Closely related languages, different ways of realizing focus," In *Proceedings of Interspeech*, Brighton, UK, pp. 1007-1010, 2009.
- [12] S. Zerbian, E. O. Aboh, K. Hartmann and M. Zimmermann, "Investigating prosodic focus marking in Northern Sotho," *Focus strategies in African languages: The interaction of focus and grammar in Niger-Congo and Afro-Asiatic*, pp. 55-79, 2007.
- [13] T. V. Perneger, "What's wrong with Bonferroni adjustments," *Bmj*, 316(7139), pp. 1236-1238, 1998.
- [14] W. L. Wu and Y. Xu, "Prosodic focus in Hong Kong Cantonese without post-focus compression," *Speech Prosody 2010-Fifth International Conference*, Chicago, USA, pp.148-151, 2010.
- [15] Y. Chen, "Post-focus compression in English by Mandarin learners," *Proceedings of the 18th International Congress of Phonetic Sciences*, pp. 1-5, 2015.
- [16] Y. Chen, "Prosodic comparisons of two types of realization of focus in Mandarin," *Speech Prosody*, pp. 917-921, 2018.
- [17] Y. Chen, S. Guion-Anderson and Y. Xu, "Post-focus compression in second language Mandarin," *Speech Prosody*, pp. 410-413, 2012.
- [18] Y. Chen, Y. Xu and S. Guion-Anderson, "Prosodic realization of focus in a bilingual production of Southern Min and Mandarin," *Phonetica*, vol. 71, pp. 249-270, 2014.
- [19] Y. R. Chao, "A system of tone letters." *Le maître phonétique*, pp. 24-27, 1930.
- [20] Y. Xu and C. X. Xu, "Phonetic realization of focus in English declarative intonation," *Journal of Phonetics*, vol. 33, pp. 159-197, 2005.
- [21] Y. Xu, "Effects of tone and focus on the formation and alignment of F0 contours," *Journal of Phonetics*, vol. 27, pp. 55-105, 1999.
- [22] Y. Xu, "Post-focus compression: Cross-linguistic distribution and historical origin," *Proc. 17th ICPHS*, Hong Kong, pp. 152-155, 2011.
- [23] Y. Xu, "ProsodyPro.praat. Version 5.5.2. Online: <https://www.phon.ucl.ac.uk/home/yi/ProsodyPro/>," 2014.
- [24] Y. Xu, S. W. Chen and B. Wang, "Prosodic focus with and without post-focus compression (PFC): A typological divide within the same language family?" *The Linguistic Review*, pp. 131-147, 2012.
- [25] F. L. Wang and S. Q. Li, "A Survey of the Changchun Dialect." Jilin Literature and History Publishing house, 1995.
- [26] Nanjing Local Chronicle Compilation Committe and Dialect Chronicle Compilation Committee, "A Survey of the Nanjing Dialect." Nanjing Press, 1993.

Research on New Event Detection Methods for Mongolian News

Shijie Wang, Feilong Bao[✉], Guanglai Gao

College of Computer Science, Inner Mongolia University

Inner Mongolian Key Laboratory of Mongolian Information Processing Technology

National & local Joint Engineering Research Center of Mongolian Intelligent Information Processing Technology

Hohhot, China, 010021

997193900@qq.com; {csfeilong, csggl}@imu.edu.cn

Abstract—New event detection (NED) aims at detecting the first news from one or multiple streams of news stories. This paper is aimed at the field of journalism and studies the related methods of Mongolian new event detection. The paper proposes a method that combines the similarity of news content with the similarity of news elements to detect the new event. For the news content representation, according to the characteristics of the news and the different vocabulary expressions in different news categories, improve the traditional TF-IDF method. In addition, extract the main elements of the news, including time, place, subject, object, denoter, and calculate the similarity of news elements between the two news documents. Finally, the similarity between the news content and the news elements is combined to calculate the final similarity for new event detection. The experimental results show that the improved method is obvious, and the performance is significantly improved compared with the traditional new event detection system.

Keywords-NED; Mongolian; News Elements; Similarity combination

I. INTRODUCTION

New event detection (NED) is defined as detecting the earliest report of a new topic in a news stream. A topic consists of a seminal event and subsequent directly related events or activities. An event is something that happens at a certain time and at a certain place. An example of a topic might be an explosion of an airplane. The first story on this topic is the story that first carries the report on the explosion of an airplane's occurrence. The other stories that make up the topic are those discussing the death toll, the rescue efforts, and the commercial impact and so on. A good NED system would be one that correctly identifies the article that reports the explosion of an airplane's occurrence as the first story. NED system has important research value. Identifying the first report of the topic through the NED system helps the tracking and detection of topic (TDT) system to mine the seminal event of the topic and establish an initial topic model. In addition, NED has important application value. With the increasing expansion of Internet resources, new events are often submerged in a large amount of daily information, which greatly limits people's timely grasp of important news. In particular, the business tracking of financial and stock markets, as well as areas such as national security and political analysis, there is a need for a mechanism to accurately and effectively capture important news events.

After the development in recent years, the new event detection technology of resource-rich languages such as

English and Chinese is becoming more and more mature. However, the research on the detection of new Mongolian events is still in its infancy. There are few resources available in Mongolian, and there is a lack of publicly labeled corpus. In addition, Mongolian is an agglutinative language. The characteristics of word formation are different from those of Chinese and English. There are also problems such as inconsistent coding. This paper is aimed at the field of journalism and studies related methods for detecting new events in Mongolian. In recent years, with the increasing network resources related to Mongolian, the contents of news web pages on the Internet have expanded rapidly. It is difficult for users to obtain information of interest from complex topics quickly and accurately. The demand for detection and tracking of Mongolian is constantly increasing. Therefore, it is of great necessity to conduct research on the new event detection task in Mongolian.

The main method of NED is to calculate the similarity between each news story on hand and all the previous received stories. If all the similarities between them do not exceed a threshold, then the story triggers a new event. In addition, there are also systems that organize the previous news into news clusters, each news cluster corresponds to a topic, and compares the new news with the previous news clusters. The experiments in the literature [3,5] prove that the former method can get better results. The core problem of the former methods is well-modeled representation of news context. General method separately used TF-IDF model [1], the context vector [2], some key words which were extracted by named entity recognition (NER) system [10, 11, 12, 13] to represent the context. Papka [7] excludes NEs that frequently appear in the corpus from the text description, and gives the location class NE four times the weight of other features. Giridhar [13] describes the report as three vector spaces, which are vectors containing all features, vectors containing only NE features, and vectors excluding NE features, and compares the effects of three vector spaces on NED, but some reports work better without NE involvement. In terms of text representation, Yang [5] only selects the best relevant reports in the class to describe the topic based on the classification of the previous report categories. Brants [19] improves the incremental TF-IDF weight calculation method and uses the vector space model for text representation. Based on this, the Hellinger distance is used to match the text relevance. Xu et al. [20] applied a time window strategy to improve the single-pass clustering in NED task. Hong et al. [21] propose a new event detection model based on division comparison of subtopic. Cang et al. [22] proposed a temporal topic model (TTM) to divide the

topics and news into some smaller events according to different time expressions describing the particular time of the event occurrence.

For the news, the five elements of the news are very important information. It summarizes the objective connection of the news events themselves and plays an important role in describing the characteristics of the news. While there is few efficient work about extraction of news factor including subject, denoter, object, time and location which can represent the news more precisely. Based on this idea, this paper extracts the news element time, place, subject, object, denoter, calculates the similarity of news elements and fuses the content similarity to detect new events.

The paper proposes a new event detection method combining content similarity and news element similarity. In the news content, two methods are used to optimize representation of news content. 1) According to a large number of corpora, the title and the first paragraph of a news can often express the main meaning of a news, giving special weight to the feature words appearing in the title or the first paragraph. 2) The literature [6] verified that different types of entities and part of speech have different sensitivity levels in different categories of news, using statistical results to optimize the weight of different categories of news for different feature words. In addition, news elements are also very important information for news. In addition to calculating the similarity of news content, the paper also extracts the main elements of news and calculates the similarity of news elements between the two news documents. To extract the factors efficiently, we applied the BiLSTM+CRF method separately on headlines and the first paragraph to extract subject, denoter and object, and on the content to extract time and location.

The remainder of the paper is organized as follows. Section 2 formally formulates the basic event detection model. Section 3 introduces our improved model. Section 4 introduces the experimental corpus, experimental process and experimental evaluation and results. Finally, Sect 5 summarizes the full text and look forward to the future work.

II. BASIC MODEL

This section presents the basic NED model that is similar to what most current systems use. The paper builds on this model. A basic NED model consists of three parts, story representation, the similarity calculation, the NED process.

A. Story Representation

Firstly, we extract term to represent the text as a collection of items, and then construct the weight vector according to the weight of items. In the traditional vector space model, we consider a single word as the term extraction. We adopt incremental TF-IDF model to calculate the weight of words in the document, the incremental TF-IDF model updates the model every time period. At time period t, the model is updated as follows:

$$df_t(w) = df_{t-1}(w) + df_{D_t}(w) \quad (1)$$

Where D_t represents the news set in the t period, $df_{D_t}(w)$ represents the number of documents containing the feature w in D_t , and $df_t(w)$ is the number of documents containing

the feature w up to time t. The paper sets D_t to 50, which means that each time period t contains 50 documents.

After the above processing, the weight of the feature w in a certain time period t is calculated as follows:

$$\text{Weight}(d, w) = \frac{\log(tf(d, w) + 1) * \log(\frac{N_{t+1}}{df_t(w) + 0.5})}{\sum_{w' \in d} \log(tf(d, w') + 1) * \log(\frac{N_{t+1}}{df_t(w') + 0.5})} \quad (2)$$

Where N_t represents the total number of reports before the time period t, and $tf(d, w)$ is the number of times the feature w appears in the news d.

Each news d at time t can be described as:

$$d \rightarrow \{\text{weight}(d, w_1), \text{weight}(d, w_2), \dots, \text{weight}(d, w_n)\}$$

Where n is the number of characteristic words of news d, and $\text{weight}(d, w)$ represents the weight of feature word w in news d at time t.

B. Similarity Calculation

The paper uses the Hellinger distance to calculate the similarity between news content. For the two news d and q, the similarity between them is expressed as:

$$\text{Sim}(d, q) = \sum_{w \in d, q} \sqrt{\text{weight}(d, w) * \text{weight}(q, w)} \quad (3)$$

C. New event detection

The new document q at the time t will be compared to all the previous documents d. The maximum similarity is compared with the specified threshold. If it is larger than the threshold, the described event is considered to be the old event, and conversely, the described event is considered to be a new event.

III. IMPROVED MODEL

The basic model uses the vector to represent the news and then calculates the content similarity to determine whether the news is new or old. Then for the reasonable representation of the news content, optimizing the weight of the feature words is a way to improve the NED effect. Literature [6] verified that different types of news have different sensitivity to different types of entities and part of speech. The literature [6] uses statistical results to optimize feature word weights. This paper uses this method to the Mongolian news corpus for verification and optimization of feature word weights. Also, the first paragraph and the title of a news can often indicate the main meaning of a news. We will give special weights to the words that appear in the title or the first paragraph, and use the above two methods to better represent the news content. In addition, analyzing the characteristics of news, the five elements of news, who, when, where, why, and what are important information for the news, and contribute to the judgment of whether the two news belong to the same topic. We extract the news elements, calculate the similarity, and combine the content similarity as the final similarity to detect the new events.

A. Improve TF-IDF model according to news characteristics

The title and the first paragraph of a news can usually express the main meaning of the news report. Set the

parameter ω to adjust the weight of the words appearing in the title or the first paragraph.

$$weight_A(d, w) = \omega * \frac{\log(tf(d, w) + 1) * \log(\frac{Nt+1}{df_t(w) + 0.5})}{\sum_{w' \in d} \log(tf(d, w') + 1) * \log(\frac{Nt+1}{df_t(w') + 0.5})} \quad (4)$$

TABLE I. EXPERIMENTAL RESULTS WITH DIFFERENT VALUES

Experiment	Miss/%	FA/%	Norm(C_{det})
$\omega = 1.2$	51.25	4.28	0.6654
$\omega = 1.4$	50.19	4.15	0.6362
$\omega = 1.6$	50.89	4.29	0.6628
$\omega = 1.8$	51.43	4.34	0.6633
$\omega = 2.0$	51.45	4.37	0.6635
$\omega = 2.2$	51.56	4.48	0.6679
$\omega = 2.4$	51.89	4.69	0.6703
$\omega = 2.6$	52.33	4.72	0.6721
$\omega = 2.8$	52.24	4.51	0.6744
$\omega = 3.0$	54.89	4.56	0.6758
$\omega = 4.0$	55.26	4.59	0.6779

Through a lot of experiments, it is concluded that when the feature word appears in the title or the first paragraph, $\omega=1.4$ has a better effect.

B. News description based on special weighting of different types of entities

The basic NED model judges whether it is a new event based on comparing the similarity of news content, then optimizing the assignment of feature word weights becomes an effective way to improve the effect of NED. According to the statistical results, different types of news have different preferences for different types of named entities. It is proposed to use statistical results to optimize the weight assignment of feature words to improve the effect of NED.

This article manually names the entity categories. Named entities include people, place name, institution name, date, time, currency, and percentage. This article focuses on the sensitivity of different named entities in different categories of news. Extracting named entities using the NER system [15]. Use the N^2 statistical method to count the correlation between feature words and topics. For a feature word w and a topic t , first, get a dependency table:

TABLE II. DEPENDENCY TABLE OF FEATURE WORD W AND TOPIC T

News number	Belong to topic T	Not belong to topic T
Include w	A	B
Not include w	C	D

The statistical method of N^2 is as follows:

$$N^2(w, t) = \frac{(A+B+C+D)*(A*D - C*B)^2}{(A+C)*(B+D)*(A+B)*(C+D)} \quad (5)$$

We use the nine news topics in the collected Mongolian news corpus to average the statistical results of the same category of named entities and the same category of topics: $N^2_{avg}(P_k, R_m) = \frac{1}{|R_m|} \{ \sum_{T \in R_m} P(w, t) * N^2(w, t) \}, k = 1 \dots K, m = 1 \dots M \quad (6)$

Where K is the number of named entities categories (7 in this paper), M is the number of news categories (9 in this paper), P_k represents the set of feature words of the k^{th} entity, R_m represents the set of topics of the m^{th} topic category, $p(w, t)$ is the probability that the feature word w appears in the topic T.

Improve the feature word weights as follows:

$$weight_B(d, w) = \frac{weight(d, w) * \alpha^{\text{type}(w)}}{\sum_{w' \in d} weight(d, w') * \alpha^{\text{type}(w')}} \quad (7)$$

Where $\text{type}(w)$ is the named entity type of the feature word w , $\text{class}(d)$ is the category to which news d belongs, and α_k^c is the weighting parameter corresponding to news category C and NER category K, setting $\alpha_k^c = N^2(k, C)$.

In the process of new event detection, this paper uses Boostexter [18] to pre-classify news according to nine topic categories. Boostexter is a Boosting-based machine learning algorithm, which learns a series of simple rules for building classifiers from training data. The feature word weight generated by the initial TF-IDF model is used as the classification feature. 1000 manual annotation data is used as the training set, and all the data are classified. The classification result is used to calculate the feature weight adjustment in Equation 7.

C. News element similarity calculation

1) News element extraction

This paper cites the method of [14], using Bi-LSTM+CRF model to extract the time, place, subject, object, denoter in Mongolian news. We extract the subject, object, and denoter that appear in the headline and the first paragraph, because not all subject, objects and denoter appearing in the entire news are beneficial to us, but only in the headline or the first paragraph is more effective and more close to the characteristics of the news.

a) Host-object similarity calculation

After the subject, object and denoter are represented by vectors, the cosine formula is used to calculate the similarity. Where a_i, b_i represents the word vector in the document, w_{ik} is the weight of the feature word in the document.

$$\text{Sim}(F1, F2) = \frac{\sum_i (a_i * b_i)}{\sqrt{\sum_i a_i^2} \sqrt{\sum_i b_i^2}} = \frac{\sum_{k=1}^n (w_{ik} * w_{jk})}{\sqrt{\sum_{k=1}^n w_{ik}^2} \sqrt{\sum_{k=1}^n w_{jk}^2}} \quad (8)$$

b) Time similarity calculation

For Mongolian news reports, the expression of time is standardized as "XX XX XX Day", so the time similarity of Mongolian news corpus is relatively simple.

$$\text{Sim}(Ti, Tj) = \begin{cases} \frac{|T_i \cap T_j|}{|T_i \cup T_j|} & T_i \cap T_j \neq 0 \\ 0 & \text{Other} \end{cases} \quad (9)$$

c) Location similarity calculation

Since there is no complete gazetteer for the Mongolian place name level mark, we use the method of cosine similarity to match the similarity of the place by formula 8, and record it as $\text{Sim}(L_i, L_j)$. For example: L_i indicates the location of the Mongolian news report d_i , L_j indicates the location of the Mongolian news report d_j . $L_i = (\text{Inner Mongolia}, \text{Hohhot}, \text{Sai Han District}, \text{Zhaowuda Road})$ (means: Inner Mongolia, Hohhot, Sai Han District, Zhaowuda Road), $L_j = (\text{Inner Mongolia}, \text{Hohhot}, \text{Sai Han District}, \text{Zhaowuda Road})$ (means: Hohhot, Sai Han District, Zhaowuda Road), $\text{Loc} = (\text{Inner Mongolia}, \text{Hohhot}, \text{Sai Han District}, \text{Zhaowuda Road}, \text{use 0, 1 to determine if it is in Loc})$ (means: Inner Mongolia, Hohhot, Sai Han District, Zhaowuda Road), $L_i = (1110)$, $L_j = (0111)$, Calculate the similarity according to the cosine formula to 0.667.

The calculation process that makes the place name similarity error smaller is as follows: First, we need to establish a Mongolian location database, which is established according to the location level; secondly, when the location of the corpus is extracted, if the two names have

the same word, use the formula (10) to calculate Similarity. If there is no identical word in the two place names, check the place name database to confirm whether the two places are the same place. If it refers to the same place, use the formula (10) to calculate the similarity of the place names. If it is not the same place, the two similarities are 0.

$$\text{Sim}(L_i, L_j) = \begin{cases} \frac{|L_i \cap L_j|}{L_i \cup L_j}, & \text{if } L_i \text{ and } L_j \text{ have the same place name and alias} \\ 0, & \text{other} \end{cases} \quad (10)$$

2) Combination of similarity

After calculating the content similarity, time similarity, location similarity, subject, object and denoter similarity between news stories, we put them together to form the final similarity of the two news reports. The final similarity calculation formula is as follows:

$$\begin{aligned} \text{Sim}(d_i, d_j) &= \alpha * \text{Sim}(d, q) + \beta * \text{Sim}(T_i, T_j) + \gamma * \\ \text{Sim}(L_i, L_j) + \delta * \text{Sim}(F1, F2) \end{aligned} \quad (11)$$

Here, $\alpha, \beta, \gamma, \delta$ are parameters. In the experiment, $\alpha = 0.7, \beta = 0.1, \gamma = 0.1, \delta = 0.1$.

After calculating the event similarity, we need to determine whether the event is a new event. This paper judges whether it is a new event by comparing with the set threshold. If the final similarity is smaller than the threshold, it is a new event. This article sets the threshold to 0.2.

IV. EXPERIMENT

A. Data preparation

This article obtains corpus from the Mongolian news website, and uses Lucene to collect and sort out nine types of news corpus, namely earthquake, explosion, fire, tourism, sports, election, finance, crime, science discovery. These nine categories are used as experimental corpora. There are 1200 news reports, of which 800 are training data and 400 are test data. In the study of this paper, Mongolian news was selected as the experimental corpus. These news corpora were derived from Mongolian web-sites such as District Love Network and Hohhot. These corpora downloaded from the website cannot be used directly in the experiment and need to be processed. In the study of this paper, the preprocessing of Mongolian news corpus includes code conversion, text proofreading, removal of stop words, and removal of affixes.

For the experiment, the work of corpus annotation is as follows:

A news corpus mainly includes the title and content, and we mark the title and content. Expand the corpus by segment and expand the segment by sentence. Mark the subject, object, time, place, denoter, people, place name, institution name, date, time, currency, and percentage in each sentence. At the same time, we also mark the specific relationship between the sentence and the sentence, including Causal, Accompany, Follow relationship and so on.

B. Experimental evaluation

The Miss Rate (Miss Rate, Miss) and the False Rate (False Rate, FA) are used as the basis for the evaluation. The cost function C_{Det} is used for evaluation. C_{Det} comprehensively considers Miss and FA, and the calculation formula is as follows:

$$C_{Det} = C_{Miss} * P_{Miss} * P_{target} + C_{FA} * P_{FA} * P_{non-target} \quad (12)$$

C_{Miss} and C_{FA} respectively represent the cost function of missed detection and false detection. The values of both are generally set in advance. According to the empirical value,

$C_{Miss}=1$ and $C_{FA}=0.1$. P_{target} and $P_{non-target}$ respectively indicate the probability of occurrence of a new event and the probability of occurrence of a non-new event, $P_{target}=1-P_{non-target}$. P_{miss} indicates the conditional probability of the missed detection rate, and P_{FA} indicates the conditional probability of the false positive rate.

Use a standardized cost function as the final evaluation criteria:

$$\text{Norm}(C_{Det}) = \frac{C_{Det}}{\min(C_{Miss} * P_{target}, C_{FA} * P_{non-target})} \quad (13)$$

The standardized cost function Norm (C_{Det}) was used as an indicator for experimental evaluation. For a system that is fully judged correctly, Norm (C_{Det}) is 0, and all news is judged as new event news.

C. Experimental design

In order to test the effect of the improved model proposed in this paper, the following five systems were implemented and tested:

SYSTEM-1, this system is the baseline system, using the basic model introduced in the second section, that is, using the incremental TF-IDF model to generate feature word weights, using weight (d, w) as the weight of the t-time feature w in the news. The weights are then calculated for content similarity for event detection.

SYSTEM-2 uses the news description method proposed in Section III A, that is, assigning special weights to the words appearing in the title and the first paragraph, using weight A (d, w) as the weight of the feature w in the news at time t And then calculate the content similarity for new event detection.

SYSTEM-3 adopts the method of Section III B, which gives different weights of different entities to different types of news, and uses weight B (d, w) as the weight of feature w in the news at t time, and calculates the content similarity for new event detection.

SYSTEM-4 adopts the methods in Sections III A and III B, that is, giving special weights to the words appearing in the title and the first paragraph, giving different weights of different entities to different types of news, and using the comprehensive adjusted weights to represent the news content. Calculate content similarity for new event detection.

SYSTEM-5 comprehensively adopts the methods of Sections III A, III B, and III C, that is, assigns special weights to the words appearing in the title and the first paragraph, and gives different weights of different entities to different types of news, and uses the comprehensive adjusted feature weights to represent the news content and calculate the content similarity. Then extract the news elements, calculate the location similarity, time similarity, subject similarity, object similarity, denoter similarity, and finally merge with the content similarity for new event detection.

The above five systems were trained on 800 Mongolian news corpora that we collected and manually labeled, and tested in another 400 articles. In order to test the performance of the proposed method, this paper establishes four systems for comparison on the above data:

SYSTEM-6 when comparing two news documents, calculate three similarities, corresponding to named entities, unnamed entities, and all feature words, and use this similarity as a feature to use the support vector machine classifier to judge the news "new" or "old".

SYSTEM-7 uses division comparison of subtopics. It utilizes the structure of topic and proposes the idea of subtopic.

SYSTEM-8 Depending on the news category, choose to use a named entity or a non-named entity to calculate the similarity and remove frequent words within the category.

SYSTEM-9 uses the method of re-evaluation of the word element, which gives special weights to different named entities and part of speech in different news categories based on statistical results.

D. Experimental results and analysis

TABLE III. COMPARISON OF TEST RESULTS

Experiment	Miss/%	FA/%	Norm(C_{Det})
SYSTEM1	50.23	4.25	0.6454
SYSTEM2	50.19	4.15	0.6362
SYSTEM3	48.27	4.01	0.5978
SYSTEM4	45.77	3.82	0.5834
SYSTEM5	42.82	3.73	0.5415
SYSTEM6	49.16	4.10	0.6324
SYSTEM7	44.86	3.79	0.5803
SYSTEM8	47.15	3.95	0.5928
SYSTEM9	45.33	3.80	0.5812

For the new event detection method applied in the Mongolian news field, the following conclusions can be obtained from Tables III:

SYSTEM2, the special weighting of the feature words appearing in the title and the first paragraph is slightly better than the general TF-IDF method. SYSTEM3, for the special weighting of entities that appear in different news categories, it is better than the general TF-IDF method. SYSTEM4 combines the above two methods to improve the TF-IDF model, the effect is better, and the missed detection rate and false detection rate are further reduced. SYSTEM5 combines the first two improvements, and combines the main elements of the news, calculates the final similarity to detect new events, this method can achieve the best results, the minimum standardization cost is reduced by 0.1039 compared with the baseline system SYSTEM1. Compared with System7, which has the best result of its kind, system5 is reduced by 0.0388. Explain that the system improvement effect is significant.

V. CONCLUSIONS

This paper proposes an improved method for the detection of new events in the Mongolian news field. This method combines the similarity of news content with the similarity of news elements, and compares the final similarity to detect new events. In the presentation of news content, according to the characteristics of the news, special weights are given to the words appearing in the title and the first paragraph, and different weights are given to the characteristic words of different entities in different news categories. Improve the TF-IDF model with the above two methods to better represent the news content. In addition, using the BiLSTM+CRF method to extract the main elements of news, including time, place, subject, object and denoter, calculate the similarity of news elements between two news documents, and combine the similarity of news content with the similarity of news elements. Calculate the similarity after fusion to detect new events. At the minimum standardization cost, the proposed method is reduced by 0.1039 compared to the baseline system, and is reduced by

0.0388 compared to similar systems. The experimental results show that the improved method proposed in this paper has a significant improvement on the results of new event detection in the Mongolian journalism field.

In this paper, the Mongolian experimental corpus is relatively small, which will affect the results of the experiment to a certain extent. In the future, it will continue to collect more corpus, do more comprehensive news classification, and continue to enhance the experimental results.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Nos. 61563040, 61773224); Natural Science Foundation of Inner Mongolia (Nos. 2018MS06006, 2016ZD06).

REFERENCES

- [1] Allan J, Papka R, Lavrenko V. On-line new event detection and tracking[C]. Proceed-ings of the 21st Annual International ACM SIGIR Conference on Research and De-velopment in Information Retrieval, New York: ACM Press, 1998: 37-45.
- [2] Linmei Hu, Bin Zhang, Lei Hou, etc. Adaptive online event detection in news streams[J]. Knowledge-Based Systems, 2017: 105-112.
- [3] Allan J, Lavrenko V, Malin D, Swan R.: Detections, bounds, and timelines: Umass and tdt-3. In: Proc. of the Topic Detections and Tracking Workship(TDT-3), Vienna, 2000, 167-174,<http://cirr.cs.umass.edu/pubfiles/ir-201.pdf>.
- [4] Juha M, Helena A M, Marko S. Simple semantics in topic detection and track-ing.Information Retrieval, 2004, 7(3-4) :347-368.
- [5] Yang Y, Pierce T, Carbonell J.:A study on retrospective and on-line event detection. In: Croft WB, Moffat A, Van Rijsbergen CJ, Wilkinson R, Zobel J, eds. Proc. of the SIGIR'98. Melbourne, 1998.28-36
- [6] Kuo Z . A New Event Detection Model Based on Term Reweighting[J]. Journal of Soft-ware, 2008, 19(4).
- [7] Papka R, Allan J. On-line new event detection using single pass clustering TITL E2: Tech-nical Report UM-CS-1998-021,1998.
- [8] Lam W, Meng H, Wong K et al. Using contextual analysis for news event detection. Interna-tional Journal on Intelligent Systems,2001,16(4):525-546.
- [9] Juha M, Helena A M,Marko S. Applying Semantic classes in event detection and track-ing//Proceedings of the International Conference on Natural Language Processing(ICON 2002). Turko,Finland,2002:175-183.
- [10] Yang, J. Zhang, J. Carbonell, and C. Jin. Topic conditioned Novelty Detection. In Proceed-ings of the 8th ACM SIGKDD International Conference, ACM Press. 2002.
- [11] M. Juha, A.M. Helena, and S. Marko. Applying Semantic Classes in Event Detection and Tracking. In Proceedings of International Conference on Natural Language Processing.
- [12] M. Juha, A.M. Helena, and S. Marko. Simple Semantics in Topic Detection and Tracking. Information Retrieval,
- [13] K. Giridhar and J. Allan. Text Classification and Named Entities for New Event Detection. In Proceedings of the 27th Annual International ACM SIGIR Conference, New York,,USA:ACM Press,2004:297-304.
- [14] Gao Yao-Wen: New event detection for Mongolian news corpus [D]. Inner Mongolia Uni-versity, 2018.
- [15] Wang Wei-Hua. Mongolian named entity recognition research [D]
- [16] Thorsten B, Francine C, Ayman F.A system for new event detection//Proceedings of the 26th Annual International ACM SIGIR Conference. Toronto ,Canda:ACM Press,2003:330-337.
- [17] Croft W B, Townsend S C,Lavrenko V. Relevance feedback and personalization :A lan-guage modeling perspective//Proceedings of

- the DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries. Dublin, Ireland,2001:49-54.
- [18] Schapire RE, Singer Y. Boostexter: A boosting-based system for text categorization. *Machine Learning*,2000,39(2/3):35-168.
- [19] Brants T, Chen F, Farahat A. A system for new event detection [C]. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. [S. l.]: ACM,2003: 330-337.
- [20] Xu RF, Peng WH, Xu J, et al. On-line new event detection using time window strategy [C]. In: 2011 International Conference on Machine Learning and Cybernetics (ICMLC). IEEE, 2011: 1932-1937
- [21] Hong Y, Zhang Y, Fan JL, et al. New event detection based on division comparison of subtopic [J]. *Chinese Journal of Computers*, 2008, 31(4): 1-9.
- [22] Cang Y, Hong Y, Yao JM, et al. New event detection based on temporal topic model [J]. *Intelligent Computer and Applications*, 2011, (1): 74-78.

Improving Mandarin Prosody Boundary Detection by Using Phonetic Information and Deep LSTM Model

Ju Lin^{1,2}

¹ Beijing Language and Culture University,
Beijing, China.

² Clemson University,
Atlanta, United States of America.
e-mail: jul@clemson.edu

Yanlu Xie

Beijing Language and Culture University, Beijing
Advanced Innovation Center for Language Resources,
Beijing, China.
e-mail: xieyanlu@blcu.edu.cn

Abstract—Automatic prosodic boundary detection is useful for automatic speech processing, such as automatic speech recognition (ASR) and speech synthesis. In this paper, we propose two techniques to improve the boundary detection performance. First, in addition to prosody features (e.g., pitch, duration and energy), phonetic information (word/articulatory information) is integrated into the framework of prosodic boundary detection. We compared two forms of phonetic information: word form and articulatory form. Moreover, boundary detection can be regarded as a sequence labeling task. A deep Long Short-Term Memory (LSTM) is adopted for this task, which replaces the traditional Deep Neural Networks (DNN) model. The experimental results showed that the boundary detection performance can be improved by the additional phonetic information, with relative 5.9% (word form) and 9.8% (articulatory form) improvements respectively in contrast with the system that only used prosody features modeled. The articulatory information and prosody features with deep LSTM achieved the best result, with further performance enhancement from 76.35% to 77.85% (relative 6.3%) compared with that modeled by DNN.

Keywords—Prosodic boundary detection, articulatory information, sequence labeling, LSTM

I. INTRODUCTION (HEADING I)

Prosody plays an important role in speech production and comprehension. An effectively-organized prosody structure enables the speakers to more clearly convey their intended meanings. Meanwhile, it also helps the listeners understand the intention of the speakers more accurately. It is known that prosodic boundary indicates the degree of disjunction between adjacent syllables in Mandarin, which divides continuous speech into several prosodic units of various sizes, such as prosodic word, intermediate phrase and intonational phrase. Such prosodic effects are highly informative for listeners, with similar results found across languages. As one of the key issues concerning prosody, prosodic phrasing affects the comprehension of the sentence in speech. For example, “打死老虎” (hit the dead tiger) and “打死/老虎” (kill the tiger) have different meanings since they have different prosodic boundaries. It is thus

Zhuanzhuan Ji, Wenwei Dong

Beijing Language and Culture University, Beijing
Advanced Innovation Center for Language Resources,
Beijing, China.

e-mail: Jizhuanzhuan2017@126.com,
dongwenwei_blcu@163.com

Jinsong Zhang*

Beijing Language and Culture University, Beijing
Advanced Innovation Center for Language Resources,
Beijing, China.
e-mail: jinsong.zhang@blcu.edu.cn

greatly beneficial to take prosodic boundaries into consideration in automatic speech processing, such as automatic speech recognition (ASR) and speech synthesis, etc.

Previous studies have presented various features and approaches on the automatic prosodic boundary detection. Ostendorf et al. [1] proposed to apply decision trees and a Markov sequence model to predict the prosody boundary of the text in English and achieved 77% accuracy. An automatic phrase boundary labeling method for speech synthesis database annotation was presented by Chen et al. [2], which used context-dependent hidden Markov models (CD-HMM) and n-gram prior distributions, and improved the F-score of phrase boundary labeling from 72.2% to 79.6% on Boston University Radio Speech Corpus (BURNC). Ni et al. [3] proposed a hierarchical prosodic break classification method, which utilized the acoustic, lexical and syntactical features, and achieved 78.25% correct rate for the testing set. Yang et al. [4] used an unsupervised method based on CD-HMMs for labeling the phrase boundary positions of a Mandarin speech synthesis database and obtained an F-score of 77.64%. Lin et al. [5] proposed to utilize tone nucleus based prosodic features and DNN model to improve the detection performance.

As mentioned above, the performance of boundary detection still needs to improve. Prosody in speech is manifested by variations of loudness, exaggeration of the pitch so that low pitches are lower and high pitches are higher, and exaggeration of consonant and vowel properties, such as vowel height and aspiration [6]. Among different prosodic boundaries, this degree of exaggeration would be the difference. Thus, the speech attribute, also known as articulatory information, can be described this degree of exaggeration. It would be useful for improving the performance of boundary detection.

Previous studies have shown that several models are used for prosodic boundary detection, such as decision trees, SVM, HMM, and DNN, etc. DNN has a strong ability in feature learning, which can map the input features into a better feature representation via non-linear transformation of several hidden layers. Deep Recurrent Neural Network (RNN) not only has the abilities of DNN's feature learning

but also can model long context and sequence information. Moreover, deep RNN with LSTM architecture [7] can address the vanishing problem that exists in standard RNN. Therefore, we propose to a prosodic boundary detection approach based on Deep LSTM, which combine the prosodic and articulatory information.

The paper is organized as follows: In Section 2, a description of the LSTM model is presented. Section 3 presents the features we used in this paper. It is followed by the experiments and results in Section 4. The paper is concluded with our directions for future work in Section 5.

II. LSTM MODEL

Prosodic boundary detection is a sequence to sequence labeling task, which is mean that each syllable of the sentence needs to design a boundary index. As we knew, LSTM can model a longer context than DNN and standard RNN. This is the reason why LSTM is suitable for this task. We first give an example of the same boundary index with same tone pattern but at different positions in the sentence. As shown in Figure 1, the upper one is at the beginning of the sentence and the nether one is at the ending of the sentence. The pitch at the beginning of the syllable in the upper figure is higher than that in the nether figure. This is due to the tone level will adjust itself to be consistent with the sentential intonation structure. For instance, the pitch contour of a declarative utterance generally declines gradually [8]. We expect that the deep LTSM architecture could model this useful information for prosodic boundary detection.

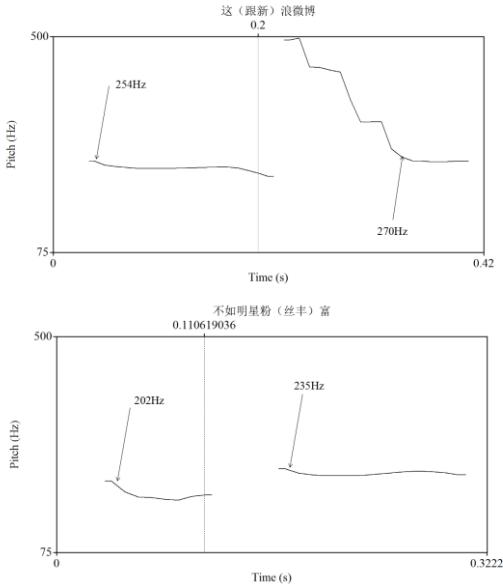


Figure 1. An example of the same boundary index at different positions in the sentence, both of them has the same tone pattern (tonel and tone 1).

The structure of LSTM we used in this paper is as shown in Figure 2. Here is an example of the unfolded architecture of unidirectional LSTM with three consecutive steps. To deal with a variable length of input sequences, zero-padding was performed to pad a shorter sentence based on the maximum length for training LSTM. If the padding values are set as zero, or some other pre-defined values, the training and testing results will be highly biased. Thus, a masking mechanism was adopted to overcome the potential

padding values problem. The Time Distributed function adds an independent layer for each time step in the recurrent model. This is used for gaining the output at each time step. The LSTM transition equations are defined as follows:

$$i_t = \text{relu}(W_i x_t + U_i h_{t-1} + b_i) \quad (1)$$

$$f_t = \text{relu}(W_f x_t + U_f h_{t-1} + b_f) \quad (2)$$

$$o_t = \text{relu}(W_o x_t + U_o h_{t-1} + b_o) \quad (3)$$

$$g_t = \tanh(W_g x_t + U_g h_{t-1} + b_g) \quad (4)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad (5)$$

$$h_t = o_t \odot \tanh(c_t) \quad (6)$$

Where the operation \odot denotes the element-wise vector product. At time t , the input gate, the forget gate, and the output gate, denoted as i_t, f_t , and o_t respectively. c_t is the memory cell and h_t is the hidden layer representation. W_* and b_* denote the weight matrix and bias vector of corresponding gate functions.

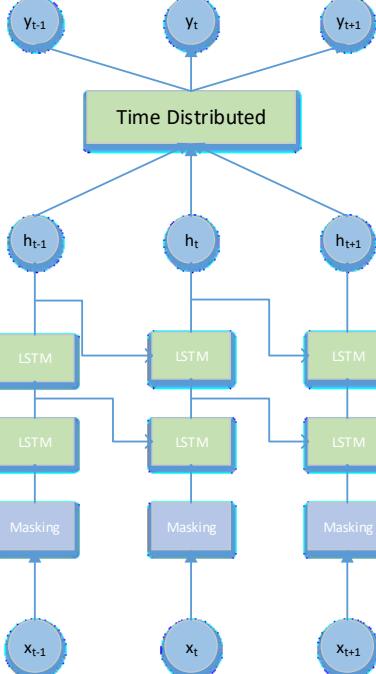


Figure 2. LSTM structure for prosodic boundary detection.

III. FEATURES

A. Phonetic information

As mentioned-above, phonetic information can be used for prosodic boundary detection. We compare two ways of feature representations to add phonetic information. One is 1 of V form, which is mean that uses a fixed-size vocabulary of symbols with V members in total, each input symbol can be coded as a vector of size V with all zeros except for the element corresponding to the symbol's order in the vocabulary, which gets a 1. The V in this paper indicates Chinese Pinyin with no tone symbol. Another way is that uses a series of detailed categories based on articulatory movements such as manner of articulation, place of articulation, etc. [8-10]. We adopted articulatory categories consistent with [8]. There are 19 categories in total, including 4 for Initials and 15 for Finals. First, the Pinyin

transcripts are mapped to the Initial and Final sequence. Then, the Initial and Final sequence is encoded in a 19-dimension vector, which has two values are 1 and others are 0.

B. Duration features

Previous studies demonstrated that duration-related features are related to boundary index [3, 11]. It is shown that pauses are more likely to appear at prosody boundaries and the duration of syllables preceding prosody boundaries is longer than that of other positions. The boundary information of Initials, Finals, and silence of the data was generated from forced alignment with a recognizer based on DNN. The durational features we used in this paper is as follows:

- The duration of silence after the current syllable
- The duration of the current syllable.

C. Pitch Features

Pitch is the most commonly used feature for prosodic boundary detection [11]. Prosody boundaries usually give rise to a variation of pitch reset, and the degree of pitch reset is greatly dependent on the level of prosody boundaries [11]. The higher boundary index is, the larger pitch reset there is. In this paper, the pitch reset is not computed explicitly but uses the context information. The pitch information of preceding, current and following syllable are spliced together. We expect the neural network could learn pitch reset information and other information via syllable based context features. Straight toolkit [12] is used for extracting pitch value. For each syllable, the following pitch-related features were calculated:

- Fitting the pitch contour of a current syllable with $f(t) = a + bt$ and $\{a, b\}$ was used to represent the pitch contour feature.
- The maximum F0 value of the current syllable.
- The minimum F0 value of the current syllable.
- The F0 range of the current syllable.
- The mean F0 value of the current tone nucleus.
- The F0 value of the first point in the current syllable.
- The F0 value of the last point in the current syllable

D. Energy features

Energy-related features are also incorporated for prosodic boundary detection. We use PRAAT toolkit [13] to extract energy features and the features we used are as follows:

- The maximum energy value in the current syllable.
- The minimum energy value in the current syllable.
- The energy range of the current syllable.
- The mean energy value of the current syllable.

IV. EXPERIMENTS AND RESULTS

A. Data set

A large mandarin speech corpus was used in this study, which was designed for TTS and labeled with prosodic ties. All of the text was read by one speaker. The speech signals were recorded in one channel, sampled at 16K Hz and at 16-bit precision. Prosodic boundaries defined in this corpus are similar to C-ToBI [14, 15]. The prosodic boundary was

labeled by 1, 2, 3, 4, which represent prosody word boundary, minor prosody phrase boundary, major prosody phrase boundary and intonation group boundary respectively. Syllable boundary not in pre-defined prosody boundaries is set as 0. In the stage of manual annotation of prosodic boundaries, annotators were asked to annotate prosodic boundary according to perceptual listening and acoustic manifestation. Each sentence was annotated by three annotators, the final label of the prosodic boundary was determined by the voting mechanism. Figure 3 is an example of a prosodic boundary annotation. Table 1 shows the break distribution in our corpus in details. In this paper, we selected the 10% of total corpus as a testing set, 10% of total corpus as a validation set, the rest as the training set.



Figure 3. An example of prosodic boundary annotation

TABLE I. THE DETAIL INFORMATION OF BOUNDARIES IN THE CORPUS

Total	0	1	2	3	4
96660	52320	19482	12048	4837	7973
100%	54.12%	20.16%	12.46%	5%	8.24%

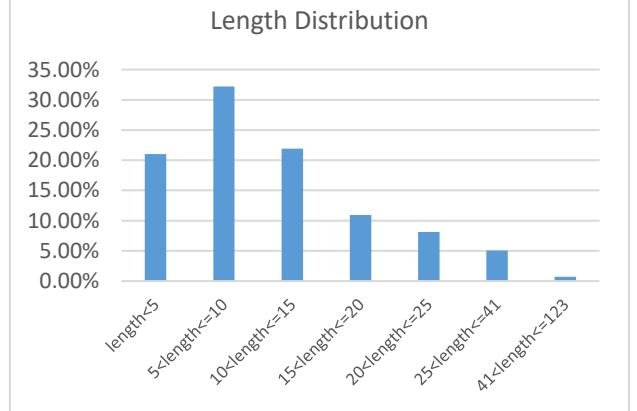


Figure 4. The distribution of sentence length in the corpus.

B. Experimental Setup

The baseline system only used prosodic features (duration, pitch, and energy features) modeled by DNN. We first compared two kinds of phonetic information under DNN framework. DNN_PY indicates that additional phonetic information is 1 of V form and DNN_SA indicates that additional phonetic information is speech attribute or articulatory form, similarly hereinafter. Before training LSTM, parameters, data pre-processing step has been executed to deal with a variable length of input sequences. Zero-padding was performed to pad short sentence based on

the maximum length (123 syllables) in the corpus. As mentioned above, we used a masking layer to address the potential padding values problem in training stage. Figure 4 is the length distribution in the corpus. From Figure 4, we can see that the length of less than 41 account for more than 99%. Thus, the time steps we selected are 41 for training LSTM. The neural networks (both DNN and LSTM) were trained for 100 epochs using SGD with a mini-batch size of 32, 20% dropout [16] in the hidden layers, a cross-entropy objective. We compared different network topologies for the best classification accuracy. Finally, the DNN has three hidden layers each with 128 nodes and the LSTM has three hidden layers each with 512 memory cells. We used KERAS [17] to realize these work.

C. Experimental Results

TABLE II. THE COMPARISON OF DIFFERENT SYSTEM'S PERFORMANCE OF BREAK DETECTION.

System	0	1	2	3	4	Accuracy
DNN	86.7	39.29	56.77	86	95.76	73.8
DNN_PY	85.67	50.02	53.12	88	98.46	75.32
DNN_SA	89.7	41.43	58.14	88	98.17	76.35
LSTM_SA	89.17	51.47	55.63	88	97.69	77.85

From Table 2 we can see that DNN_PY system, which uses conventional prosodic features and additional 1 of V feature vector, improves accuracy from 73.8% to 75.32, compared with DNN baseline, where only uses conventional prosodic features. DNN_SA system improves the boundary detection performance significantly with an absolute increase of 2.55% in contrast with the baseline system. This observation confirms that the additional phonetic information characterized by DNN is helpful for prosodic boundary detection. Moreover, DNN_SA system outperforms DNN_PY, which demonstrates that the form of speech attribute is more effective for boundary detection than the form of 1 of V. The form of 1 of V may result in data sparseness since the appended phonetic vector is a very high-dimensional vector with one-hot encoding. This also could increase model complexity. The form of speech attribute not only reduces the dimension of the appended feature vector but also includes the relationship among words through articulatory movements such as manner of articulation, place of articulation, which is mean that these words with similar articulatory movements are easier to cluster together and have closer boundary index. These may be the reason for the increase by appending speech attribute information.

LSTM with speech attribute information system further improves the performance from 76.35% to 77.85%, compared with the DNN_SA system. This indicates that the context information modeled by LSTM is helpful for prosodic boundary detection. Table 2 also gives the detailed detection performance for each boundary index. We can see that the main improvement by LSTM_SA is for prosody word boundary (boundary index is 1), which is an almost absolute enhancement of 10% compared with DNN_SA. As the input of LSTM is the whole sentence, the

tiny difference among closer boundaries with a sentence can be captured by LSTM.

V. CONCLUSIONS

In this paper, we compare two ways to incorporate phonetic information for prosodic boundary detection under DNN framework. Experimental results show that both of them can improve the detection performance compared with only utilizing conventional prosodic features (Pitch, energy and duration features). Using the form of speech attribute to represent phonetic information achieves a better result than using the form of 1 of V. The boundary detection performance can be further enhanced by using LSTM, which demonstrates long context information is helpful for this task. For future work, previous studies indicate that spectrum features are related to F0 contour [18-20]. Thus, spectrum features would incorporate into the LSTM based system.

ACKNOWLEDGMENT

This study was supported by Advanced Innovation Center for Language Resource and Intelligence (KYR17005), the Special Program for Key Basic Research fund of Beijing Language and Culture University (the Fundamental Research Funds for the Central Universities) (16ZDJ03), the Fundamental Research Funds for the Central Universities (18YJ030006).

REFERENCES

- [1] C. W. Wightman, M. Ostendorf, "Automatic labeling of prosodic patterns" [J]. Speech and Audio Processing, 1994, vol. 2, no. 4: 469-481.
- [2] Q. Chen, Z. H. Ling, C. Y. Yang, L. R. Dai, "Automatic phrase boundary labeling of speech synthesis database using context-dependent HMMs and N-Gram Prior Distributions" [C]. INTERSPEECH 2015.
- [3] C. J. Ni, A. Y. Zhang, W. J. Liu, and B. Xu, "Classification of mandarin prosodic break based on hierarchical structure of prosodic break" [J]. Application Research of Computers, 2011, vol. 28, no. 7.
- [4] C. Y. Yang, L. X. Zhu, Z. H. Ling, and L. R. Dai, "Automatic phrase boundary labeling for a Mandarin TTS corpus using the Viterbi decoding algorithm" [J]. Tsinghua Science and Technology, 2011, vol. 51, no. 9, pp: 1276-1281.
- [5] J. Lin, Y. Xie, W. Zhang, et al. "Automatic Mandarin prosody boundary detecting based on tone nucleus features and DNN model" [C]. International Symposium on Chinese Spoken Language Processing. IEEE, 2017.
- [6] P. A. Ladefoged, "Course in Phonetics-Second Edition" [M]. Heinle, 2013.
- [7] S. Hochreiter, J. Schmidhuber, "Long Short-Term Memory" [J]. Neural Computation, 1997, 9(8):1735-1780.
- [8] J. Lin, W. wang, Y. Gao, et al. "Improving Mandarin Tone Recognition Based on DNN by Combining Acoustic and Articulatory Features Using Extended Recognition Networks" [J]. Journal of Signal Processing Systems, 2018(7):1-11.
- [9] Z. J. Wu, M. C. Lin, "Experimental phonetics summary" [M], Beijing: Higher Education Press, pp. 153-191, 1989.
- [10] J. L. Zhang, "Fundamentals of Chinese Man-Machine communication". Shanghai: Shanghai Scientific & Technical Publishers, 2010.
- [11] Z. Y. Xiong, and M. C. Lin, "prosody expression in the position of speech break" [C]. National Conference on Man-machine Speech Communication (NCMMSC 2006).
- [12] H. Kawahara, M. Morise, T. Takahashi, et al. "Tandem-STRAIGHT: A temporally stable power spectral representation for periodic signals and applications to interference-free spectrum, F0, and

- aperiodicity estimation”[C]. IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2008:3933-3936.
- [13] Boersma, Paul & Weenink, David (2018). Praat: doing phonetics by computer [Computer program]. <http://www.praat.org/>
- [14] X. X. Chen, A. J. Li, S. G. Hua, “An application of SAMPA-C for standard Chinese” [C]. Sixth International Conference on Spoken Language Processing, 2000.
- [15] A. J. Li, “Chinese prosody and prosodic labeling of spontaneous speech” [C]. Speech Prosody, 2002.
- [16] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” arXiv preprint arXiv: 1207. 0580, 2012
- [17] Chollet, F., (2015). Keras. GitHub repository, <https://github.com/fchollet/keras>.
- [18] J. Zhang, “The intrinsic fundamental frequency of vowels and the effect of speech modes on formants” [J]. Acta Acustica, 1987, pp.390-393.
- [19] I. Lehiste, G. E. Peterson, “Some basic considerations in the analysis of intonation” [J]. The Journal of the Acoustical Society of America, 1961,33(4), 419–425.
- [20] N. Ryant, M. Slaney, M. Liberman, E. Shriberg, J. Yuan, “Highly accurate mandarin tone classification in the absence of pitch information” [C]. In SPEECHPROSODY 7 – 7th International Conference on Speech Prosody, May 20-23, Dublin, Ireland, Proceedings, 2014, pp. 673-677.