

LNAI 13028

Lu Wang
Yansong Feng
Yu Hong
Ruifang He (Eds.)

Natural Language Processing and Chinese Computing

10th CCF International Conference, NLPCC 2021
Qingdao, China, October 13–17, 2021
Proceedings, Part I



Lecture Notes in Artificial Intelligence 13028

Subseries of Lecture Notes in Computer Science

Series Editors

Randy Goebel

University of Alberta, Edmonton, Canada

Yuzuru Tanaka

Hokkaido University, Sapporo, Japan

Wolfgang Wahlster

DFKI and Saarland University, Saarbrücken, Germany

Founding Editor

Jörg Siekmann

DFKI and Saarland University, Saarbrücken, Germany

More information about this subseries at <http://www.springer.com/series/1244>

Lu Wang · Yansong Feng ·
Yu Hong · Ruifang He (Eds.)

Natural Language Processing and Chinese Computing

10th CCF International Conference, NLPCC 2021
Qingdao, China, October 13–17, 2021
Proceedings, Part I



Springer

Editors

Lu Wang
University of Michigan
Ann Arbor, MI, USA

Yansong Feng
Peking University
Beijing, China

Yu Hong
Soochow University
Suzhou, China

Ruifang He
Tianjin University
Tianjin, China

ISSN 0302-9743

ISSN 1611-3349 (electronic)

Lecture Notes in Artificial Intelligence

ISBN 978-3-030-88479-6 ISBN 978-3-030-88480-2 (eBook)

<https://doi.org/10.1007/978-3-030-88480-2>

LNCS Sublibrary: SL7 – Artificial Intelligence

© Springer Nature Switzerland AG 2021

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Welcome to NLPCC 2021, the tenth CCF International Conference on Natural Language Processing and Chinese Computing. Following the success of previous conferences held in Beijing (2012), Chongqing (2013), Shenzhen (2014), Nanchang (2015), Kunming (2016), Dalian (2017), Hohhot (2018), Dunhuang (2019), and Zhengzhou (2020), this year's NLPCC was held in Qingdao, a beautiful coastal city in East China. As a premier international conference on natural language processing and Chinese computing, organized by the CCF-NLP (Technical Committee of Natural Language Processing, China Computer Federation, formerly known as Technical Committee of Chinese Information, China Computer Federation), NLPCC 2021 serves as an important forum for researchers and practitioners from academia, industry, and government to share their ideas, research results, and experiences, and to promote their research and technical innovations in the fields.

The fields of natural language processing (NLP) and Chinese computing (CC) have boomed in recent years. Following NLPCC's tradition, we welcomed submissions in ten areas for the main conference: Fundamentals of NLP; Machine Translation and Multilinguality; Machine Learning for NLP; Information Extraction and Knowledge Graph; Summarization and Generation; Question Answering; Dialogue Systems; Social Media and Sentiment Analysis; NLP Applications and Text Mining; Multimodality and Explainability. On the submission deadline, we were thrilled to have received a record number of 446 valid submissions to the main conference.

After a rigid review process, out of 446 submissions (some of which were withdrawn or rejected without review due to format issues or policy violations), 104 papers were finally accepted to appear in the main conference, where 89 were written in English and 15 in Chinese, resulting in an acceptance rate of 23.3%. Among them, 72 submissions were accepted as oral papers and 32 as poster papers. Specifically, ten papers were nominated by our area chairs for the best paper award. An independent best paper award committee was formed to select the best papers from the shortlist. This proceedings includes only the accepted English papers; the Chinese papers will appear in the ACTA Scientiarum Naturalium Universitatis Pekinensis. In addition to the main proceedings, three papers were accepted to the Student workshop, 22 papers were accepted to the Evaluation workshop, and two papers were accepted to the Explainable AI (XAI) workshop.

We were honored to have four internationally renowned keynote speakers—Rada Mihalcea (University of Michigan), Yanchao Bi (Beijing Normal University), Sebastian Riedel (University College London and Facebook AI Research), and Graham Neubig (Carnegie Mellon University)—share their findings on recent research progress and achievements in natural language processing.

We would like to thank all the people who have contributed to NLPCC 2021. First of all, we would like to thank our 20 area chairs for their hard work recruiting reviewers, monitoring the review and discussion processes, and carefully rating and

recommending submissions. We would like to thank all 432 reviewers for their time and efforts to review the submissions. We are very grateful to Tim Baldwin, Chin-Yew Lin, Kang Liu, Deyi Xiong, and Yue Zhang for their participation in the best paper committee. We are also grateful for the help and support from the general chairs, Tim Baldwin and Jie Tang, and from the organization committee chairs, Zhumin Chen, Pengjie Ren, and Xiaojun Wan. Special thanks go to Yu Hong and Ruifang He, the publication chairs, for their great help. We greatly appreciate all your help!

Finally, we would like to thank all the authors who submitted their work to NLPCC 2021, and thank our sponsors for their contributions to the conference. Without your support, we could not have such a strong conference program.

We were happy to see you at NLPCC 2021 in Qingdao and hope you enjoyed the conference!

October 2021

Lu Wang
Yansong Feng

Organization

NLPCC 2021 was organized by the China Computer Federation (CCF), and hosted by Shandong University. The proceedings were published in Lecture Notes on Artificial Intelligence (LNAI), Springer-Verlag, and ACTA Scientiarum Naturalium Universitatis Pekinensis.

Organization Committee

General Chairs

Tim Baldwin	University of Melbourne, Australia
Jie Tang	Tsinghua University, China

Program Committee Chairs

Lu Wang	University of Michigan, USA
Yansong Feng	Peking University, China

Student Workshop Chairs

Fang Kong	Soochow University, China
Meishan Zhang	Tianjin University, China

Evaluation Workshop Chairs

Jiajun Zhang	Institute of Automation, Chinese Academy of Sciences, China
Yunbo Cao	Tencent, China

Tutorial Chairs

Minlie Huang	Tsinghua University, China
Zhongyu Wei	Fudan University, China

Publication Chairs

Yu Hong	Soochow University, China
Ruifang He	Tianjin University, China

Journal Coordinator

Yunfang Wu Peking University, China

Conference Handbook Chair

Xiaomeng Song Shandong University, China

Sponsorship Chairs

Dongyan Zhao Peking University, China
Zhaochun Ren Shandong University, China

Publicity Chairs

Wei Jia Beijing Academy of Artificial Intelligence, China
Jing Li Hong Kong Polytechnic University, China
Ruifeng Xu Harbin Institute of Technology, China

Organization Committee Chairs

Zhumin Chen Shandong University, China
Pengjie Ren Shandong University, China
Xiaojun Wan Peking University, China

Area Chairs

Fundamentals of NLP

Liang Huang Oregon State University, USA
Kewei Tu ShanghaiTech University, China

Machine Translation and Multilinguality

Derek Wong University of Macau, Macau
Boxing Chen Alibaba Group, China

Machine Learning for NLP

Yangfeng Ji University of Virginia, USA
Lingpeng Kong The University of Hong Kong, Hong Kong

Information Extraction and Knowledge Graph

Lifu Huang Virginia Tech, USA
Shizhu He Chinese Academy of Sciences, China

Summarization and Generation

Rui Zhang Pennsylvania State University, USA
Jin-ge Yao Microsoft Research, China

Question Answering

Huan Sun Ohio State University, USA
Yiming Cui Harbin Institute of Technology, China

Dialogue Systems

Social Media and Sentiment Analysis

Fei Liu University of Central Florida, USA
Duyu Tang Tencent AI Lab, China

NLP Applications and Text Mining

Wenpeng Yin Salesforce Research, USA
Zhunchen Luo PLA Academy of Military Science, China

Multimodality and Explainability

Xin Wang University of California, Santa Cruz, USA
Zhongyu Wei Fudan University, China

Treasurer

Webmaster

Hui Liu Peking University, China

Program Committee

Ayana	Inner Mongolia University of Finance and Economics, China
Bo An	Institute of Software, Chinese Academy of Sciences, China
Xiang Ao	Institute of Computing Technology, Chinese Academy of Sciences, China
Jiaqi Bai	Beihang University, China
Guangsheng Bao	Westlake University, China
Junwei Bao	JD AI Research, China
Oimeng Bao	University of Auckland, New Zealand

Paul Buitelaar	Data Science Institute, National University of Ireland Galway, Ireland
Xuanting Cai	Facebook, USA
Xiangrui Cai	Nankai University, China
Yi Cai	South China University of Technology, China
Yixin Cao	Nanyang Technological University, Singapore
Yuan Cao	Google Brain, USA
Shuyang Cao	University of Michigan, USA
Pengfei Cao	Institute of Automation, Chinese Academy of Sciences, China
Ziqiang Cao	Soochow University, China
Hailong Cao	Harbin Institute of Technology, China
Shuaichen Chang	Ohio State University, USA
Zhangming Chan	Peking University, China
Zhumin Chen	Shandong University, China
Sihao Chen	University of Pennsylvania, USA
Xiuying Chen	Peking University, China
Jiajun Chen	Nanjing University, China
Junkun Chen	Oregon State University, USA
Sanxing Chen	University of Virginia, USA
Hongshen Chen	JD.com, China
Zhipeng Chen	iFLYTEK Research, China
Jie Chen	Ahu University, China
Qingcai Chen	Harbin Institute of Technology, China
Lei Chen	Fudan University, China
Wenliang Chen	Soochow University, China
Shuang Chen	Harbin Institute of Technology, China
Xinyu Chen	Soochow University, China
Ruijun Chen	Yunnan University, China
Bo Chen	Institute of Software, Chinese Academy of Sciences, China
Yulong Chen	Zhejiang University/Westlake University, China
Huadong Chen	ByteDance, China
Pei Chen	Texas A&M University, USA
Lu Chen	Shanghai Jiao Tong University, China
Guanyi Chen	Utrecht University, The Netherlands
Yidong Chen	Xiamen University, China
Liwei Chen	Peking University, China
Xinchi Chen	Amazon Web Services, USA
Hanjeie Chen	University of Virginia, USA
Wei Chen	Fudan University, China
Hannah Chen	University of Virginia, USA
Yubo Chen	Institute of Automation, Chinese Academy of Sciences, China
Kehai Chen	National Institute of Information and Communications Technology, Japan

Mingda Chen	Toyota Technological Institute at Chicago, USA
Jiangjie Chen	Fudan University, China
Jifan Chen	University of Texas at Austin, USA
Jiaao Chen	Georgia Institute of Technology, USA
Jiayang Cheng	Fudan University, China
Xin Cheng	Peking University, China
Qikai Cheng	Wuhan University, China
Zewen Chi	Beijing Institute of Technology, China
Zewei Chu	Google, USA
Chenhui Chu	Kyoto University, Japan
James Cross	Facebook, USA
Peng Cui	Harbin Institute of Technology, China
Yiming Cui	Harbin Institute of Technology, China
Mao Cunli	Kunming University of Science and Technology, China
Junqi Dai	Fudan University, China
Xinyu Dai	Nanjing University, China
Xiang Deng	Ohio State University, USA
Xiao Ding	Harbin Institute of Technology, China
Chenchen Ding	NICT, Japan
Mengxing Dong	Soochow University, China
Zhicheng Dou	Renmin University of China, China
Dejing Dou	University of Oregon, USA
Ziyi Dou	UCLA, USA
Rotem Dror	University of Pennsylvania, Israel
Wanyu Du	University of Virginia, USA
Jinhua Du	Investments AI, AIG, UK
Xinya Du	Cornell University, USA
Junwen Duan	Central South University, China
Chaoqun Duan	Harbin Institute of Technology, China
Xiangyu Duan	Soochow University, China
Philipp Dufter	Ludwig-Maximilians-Universität München, Germany
Kevin Duh	Johns Hopkins University, USA
Ismail El Maarouf	Fortia Financial Solutions, France
Alexander Fabbri	Yale University, USA
Chuang Fan	Harbin Institute of Technology, China
Zhihao Fan	Fudan University, China
Zhiyuan Fang	Arizona State University, USA
Xiachong Feng	Harbin Institute of Technology, China
Zhangyin Feng	Harbin Institute of Technology, China
Xiaocheng Feng	Harbin Institute of Technology, China
Shi Feng	Northeastern University, China
Jiazhan Feng	Peking University, China
Xingyu Fu	University of Pennsylvania, USA
Yao Fu	University of Edinburgh, UK

Guohong Fu	Soochow University, China
Zihao Fu	The Chinese University of Hong Kong, Hong Kong Special Administrative Region of China
Yi Fung	University of Illinois at Urbana Champaign, USA
Jun Gao	Harbin Institute of Technology, China
Shen Gao	Peking University, China
Yang Gao	Royal Holloway, University of London, UK
Yifan Gao	The Chinese University of Hong Kong, Hong Kong Special Administrative Region of China
Michaela Geierhos	Universität der Bundeswehr München, Germany
Ruiying Geng	Alibaba Group, China
Erfan Ghadery	KU Leuven, Belgium
Heng Gong	Harbin Institute of Technology, China
Yeyun Gong	Microsoft Research Asia, China
Jiachen Gu	University of Science and Technology of China, China
Yu Gu	Ohio State University, USA
Jian Guan	Tsinghua University, China
Yuhang Guo	Beijing Institute of Technology, China
Daya Guo	Sun Yat-Sen University, China
Qipeng Guo	Fudan University, China
Junliang Guo	University of Science and Technology of China, China
Jiale Han	Beijing University of Posts and Telecommunications, China
Fangqiu Han	Facebook, Inc., USA
Xudong Han	University of Melbourne, Australia
Lifeng Han	Dublin City University, Ireland
Wenjuan Han	Beijing Institute for General Artificial Intelligence, China
Tianyong Hao	South China Normal University, China
Shizhu He	Institute of Automation, Chinese Academy of Sciences, China
Zhongjun He	Baidu, Inc., China
Ziwei He	Shanghai Jiao Tong University, China
Ben He	University of Chinese Academy of Sciences, China
Ruifang He	Tianjin University, China
Yanqing He	Institute of Scientific and Technical Information of China, China
Yu Hong	Soochow University, China
Xudong Hong	Saarland University / MPI Informatics, Germany
Lei Hou	Tsinghua University, China
Yacheng Hsu	The University of Hong Kong, Taiwan, China
Huang Hu	Microsoft STCA NLP Group, China
Zhe Hu	Baidu, China
Zikun Hu	National University of Singapore, China
Minghao Hu	PLA Academy of Military Science, China
Baotian Hu	Harbin Institute of Technology, China

Wenpeng Hu	Peking University, China
Zechuan Hu	ShanghaiTech University, China
Xinyu Hua	Northeastern University, USA
Zhen Huang	National University of Defense Technology, China
Jiangping Huang	Chongqing University of Posts and Telecommunications, China
Shujian Huang	Nanjing University, China
Qingbao Huang	Guangxi University/South China University of Technology, China
Zhongqiang Huang	Alibaba Group, USA
Yongjie Huang	Sun Yat-sen University, China
Junjie Huang	Beihang University, China
Yi-Ting Huang	Academia Sinica, Taiwan, China
Julia Ive	Imperial College London, UK
Lei Ji	Microsoft Research Asia, China
Menglin Jia	Cornell University, USA
Zixia Jia	ShanghaiTech University, China
Ping Jian	Beijing Institute of Technology, China
Xuhui Jiang	Institute of Computing Technology, Chinese Academy of Sciences, China
Yong Jiang	Alibaba DAMO Academy, China
Xin Jiang	Huawei Noah’s Ark Lab, Hong Kong Special Administrative Region of China
Ming Jiang	University of Illinois at Urbana-Champaign, USA
Yichen Jiang	University of North Carolina at Chapel Hill, USA
Wenbin Jiang	Baidu Inc., China
Peng Jin	Leshan Normal University, China
Xisen Jin	University of Southern California, USA
Zhijing Jin	Max Planck Institute for Intelligent Systems, Germany
Zhilong Jin	Soochow University, China
Lin Jun	Alibaba Group, China
Zixuan Ke	University of Illinois at Chicago, USA
Dongjin Kim	KAIST, South Korea
Fang Kong	Soochow University, China
Bernhard Kratzwald	ETH Zurich, Switzerland
Tuan Lai	University of Illinois at Urbana-Champaign, USA
Viet Lai	University of Oregon, USA
Yuxuan Lai	Peking University, China
Man Lan	East China Normal University, China
Chongshou Li	Southwest Jiaotong University, China
Maoxi Li	Jiangxi Normal University, China
Bin Li	Nanjing Normal University, China
Mingzhe Li	Peking University, China
Hongzheng Li	Beijing Institute of Technology, China
Chenliang Li	Wuhan University, China
Zhongyang Li	Harbin Institute of Technology, China

Manling Li	University of Illinois Urbana-Champaign, USA
Dongfang Li	Harbin Institute of Technology, China
Shasha Li	National University of Defense Technology, China
Xinyi Li	National University of Defense Technology, China
Zejun Li	Fudan University, China
Peifeng Li	Soochow University, China
Irene Li	Yale University, USA
Jian Li	Huawei Noah's Ark Lab, China
Piji Li	Tencent AI Lab, China
Xin Li	Alibaba Group, China
Lishuang Li	Dalian University of Technology, China
Fei Li	Wuhan University, China
Weikang Li	Peking University, China
Yachao Li	Soochow University, China
Liangyou Li	Huawei Noah's Ark Lab, Hong Kong Special Administrative Region of China
Qintong Li	Shandong University, China
Xinhang Li	Tsinghua University, China
Linjie Li	Microsoft, USA
Yanran Li	The Hong Kong Polytechnic University, Hong Kong Special Administrative Region of China
Jing Li	The Hong Kong Polytechnic University, Hong Kong Special Administrative Region of China
Junhui Li	Soochow University, China
Zuchao Li	Shanghai Jiao Tong University, China
Lin Li	Qinghai Normal University, China
Zhenghua Li	Soochow University, China
Ruizhe Li	University of Sheffield, UK
Xilai Li	Amazon, USA
Peng Li	Institute of Information Engineering, CAS, China
Liunian Harold Li	UCLA, USA
Binyang Li	University of International Relations, China
Xiaolong Li	SAS Institute Inc., USA
Zhixu Li	Soochow University, China
Zichao Li	McGill University, Canada
Xiao Li	University of Reading, UK
Zujie Liang	Sun Yat-sen University, China
Paul Pu Liang	Carnegie Mellon University, USA
Ying Lin	Apple, USA
Junyang Lin	Alibaba Group, China
Zhaojiang Lin	The Hong Kong University of Science and Technology, Hong Kong Special Administrative Region of China
Zi Lin	Google, USA
Zhouhan Lin	Shanghai Jiao Tong University, China
Ye Liu	National University of Singapore, Singapore

Chang Liu	Peking University, China
Ling Liu	University of Colorado Boulder, USA
Qun Liu	Chongqing University of Posts and Telecommunications, China
Tianyu Liu	Peking University, China
Shujie Liu	Microsoft Research Asia, China
Qun Liu	Huawei Noah's Ark Lab, China
Dayiheng Liu	Alibaba DAMO Academy, China
Qingbin Liu	Institute of Automation, Chinese Academy of Sciences/University of Chinese Academy of Sciences, China
Yongbin Liu	University of South China, China
Xuebo Liu	University of Macau, Macau
Jian Liu	Beijing Jiaotong University, China
Xianggen Liu	Tsinghua University, China
Jiangming Liu	University of Edinburgh, UK
Lemao Liu	Tencent AI Lab, China
Xiao Liu	Beijing Institute of Technology, China
Qian Liu	Beihang University, China
Yunfei Long	University of Essex, UK
Xin Lu	Harbin Institute of Technology, China
Hengtong Lu	Beijing University of Posts and Telecommunications, China
Yaojie Lu	Institute of Software, Chinese Academy of Sciences, China
Yinglong Ma	North China Electric Power University, China
Yun Ma	Peking University, China
Xianling Mao	Beijing Institute of Technology, China
Tao Meng	UCLA, USA
Haitao Mi	Ant Group, USA
Tao Mingxu	Peking University, China
Xiangyang Mou	Rensselaer Polytechnic Institute, USA
Preslav Nakov	Qatar Computing Research Institute, HBKU, Qatar
Feng Nie	Sun Yat-sen University, China
Qiang Ning	Amazon, USA
Yawen Ouyang	Nanjing University, China
Vardaan Pahuja	Ohio State University, USA
Huijie Pan	University of Hong Kong, Hong Kong Special Administrative Region of China
Jiaxin Pei	University of Michigan, USA
Xutan Peng	University of Sheffield, UK
Baolin Peng	Microsoft Research, USA
Longhua Qian	Soochow University, China
Tao Qian	Hubei University of Science and Technology, China
Libo Qin	Harbin Institute of Technology, China
Yanxia Qin	Donghua University, China

Liang Qiu	University of California, Los Angeles, USA
Zhaochun Ren	Shandong University, China
Shuo Ren	Beihang University, China
Pengjie Ren	Shandong University, China
Martin Schmitt	Ludwig-Maximilians-Universität München, Germany
Stephanie Schoch	University of Virginia, USA
Lei Sha	University of Oxford, UK
Zhihong Shao	Tsinghua University, China
Lei Shen	Institute of Computing Technology, Chinese Academy of Sciences, China
Haoyue Shi	Toyota Technological Institute at Chicago, USA
Weijia Shi	UCLA, USA
Weiyan Shi	Columbia University, USA
Xing Shi	DiDi Research America, USA
Lei Shu	Amazon Web Services AI, USA
Chenglei Si	University of Maryland, College Park, USA
Jyotiika Singh	ICX Media, Inc., USA
Yiping Song	National University of Defense Technology, China
Kaiqiang Song	University of Central Florida, USA
Linfeng Song	Tencent AI Lab, USA
Wei Song	Capital Normal University, China
Yixuan Su	University of Cambridge, UK
Elior Sulem	University of Pennsylvania, USA
Simeng Sun	University of Massachusetts Amherst, USA
Huan Sun	Ohio State University, USA
Kai Sun	Cornell University, USA
Chengjie Sun	Harbin Institute of Technology, China
Chuanqi Tan	Alibaba Group, China
Minghuan Tan	Singapore Management University, Singapore
Zhixing Tan	Tsinghua University, China
Danie Tang	Huazhong University of Science and Technology, China
Jintao Tang	National University of Defense Technology, China
Duyu Tang	Tencent, China
Buzhou Tang	Harbin Institute of Technology, China
Ruixuan Tang	University of Virginia, USA
Chongyang Tao	Microsoft Corporation, China
Zhiyang Teng	Westlake University, China
Zhiliang Tian	Hong Kong University of Science and Technology, Hong Kong Special Administrative Region of China
Zhoujin Tian	Beihang University, China
Lifu Tu	Toyota Technological Institute at Chicago, USA
Yu Wan	University of Macau, Macau
Liran Wang	Beihang University, China
Lingzhi Wang	The Chinese University of Hong Kong, China
Di Wang	Woobo Inc, USA

Wei Wang	Tsinghua University, China
Ruize Wang	Fudan University, China
Xing Wang	Tencent, China
Tao Wang	King's College London, UK
Jingjing Wang	Soochow University, China
Le Wang	Chinese Academy of Military Science, China
Hongwei Wang	University of Illinois Urbana-Champaign, USA
Hongling Wang	Soochow University, China
Shaonan Wang	Institute of Automation, Chinese Academy of Sciences, China
Sijia Wang	Virginia Tech, USA
Xun Wang	University of Massachusetts Amherst, USA
Zhen Wang	Ohio State University, USA
Rui Wang	Shanghai Jiao Tong University, China
Siyuan Wang	Fudan University, China
Yaqiang Wang	Chengdu University of Information Technology, China
Dingmin Wang	University of Oxford, UK
Zijian Wang	Amazon Web Services AI, USA
Qingyun Wang	University of Illinois at Urbana-Champaign, USA
Ke Wang	Peking University, China
Kexiang Wang	Peking University, China
Pancheng Wang	National University of Defence Technology, China
Ge Wang	ShanghaiTech University, China
Hong Wang	University of California, Santa Barbara, USA
Jianyou Wang	University of California, San Diego, USA
Zhuoyu Wei	Yuanfudao, China
Zhongyu Wei	Fudan University, China
Haoyang Wen	Carnegie Mellon University, USA
Shuangzhi Wu	Tencent, China
Sixing Wu	Peking University, China
Junshuang Wu	Beihang University, China
Yu Wu	Microsoft Research Asia, China
Lianwei Wu	Xi'an Jiaotong University, China
Chiensheng Wu	Salesforce, USA
Changxing Wu	East China Jiaotong University, China
Congying Xia	University of Illinois at Chicago, USA
Rui Xia	Nanjing University of Science and Technology, China
Wen Xiao	University of British Columbia, Canada
Tong Xiao	Northeastern University, China
Yuqiang Xie	Institute of Information Engineering, Chinese Academy of Sciences, China
Deyi Xiong	Tianjin University, China
Hao Xiong	Alibaba Group, China
Can Xu	Microsoft STCA NLP Group, China
Ruijian Xu	Peking University, China
Silei Xu	Stanford University, USA
Jingjing Xu	ByteDance AI Lab, China
Tong Xu	University of Science and Technology of China, China

Yiheng Xu	Microsoft Research Asia, China
Yan Xu	Hong Kong University of Science and Technology, Hong Kong Special Administrative Region of China
Jinan Xu	Beijing Jiaotong University, China
Yadollah Yaghoobzadeh	University of Tehran, Iran
Yuanmeng Yan	Beijing University of Posts and Telecommunications, China
Lingyong Yan	Baidu Inc., China
Songlin Yang	ShanghaiTech University, China
Liang Yang	Dalian University of Technology, China
Muyun Yang	Harbin Institute of Technology, China
Yaqin Yang	Paypal, USA
Jingxuan Yang	Beijing University of Posts and Telecommunications, China
Qiang Yang	King Abdullah University of Science and Technology, Saudi Arabia
Ziqing Yang	iFLYTEK Research, China
Haoran Yang	The Chinese University of Hong Kong, Hong Kong Special Administrative Region of China
Baosong Yang	Alibaba DAMO Academy, China
Cheng Yang	Beijing University of Posts and Telecommunications, China
Zixiaofan Yang	Columbia University, USA
Zhiwei Yang	Jilin University, China
Kai Yang	Peking University, China
Zhewei Yao	University of California, Berkeley, USA
Jianmin Yao	Soochow University, China
Yiqun Yao	University of Michigan, USA
Wenlin Yao	Tencent AI Lab, USA
Zhe Ye	Microsoft, China
Xiaoyuan Yi	Tsinghua University, China
Pengcheng Yin	Carnegie Mellon University, USA
Da Yin	University of California, Los Angeles, USA
Zhiwei Yu	Microsoft Research Asia, China
Adams Yu	Google Brain, USA
Tiezheng Yu	The Hong Kong University of Science and Technology, Hong Kong Special Administrative Region of China
Heng Yu	Alibaba Group, China
Pengfei Yu	Department of Computer Science, University of Illinois at Urbana-Champaign, USA
Dian Yu	Tencent AI Lab, USA
Xiaodong Yu	University of Pennsylvania, USA
Tao Yu	Yale University, USA
Chunyuan Yuan	Institute of Information Engineering, Chinese Academy of Sciences, China

Zheng Yuan	University of Cambridge, UK
Xiang Yue	Ohio State University, USA
Zhiyuan Zeng	Beijing University of Posts and Telecommunications, China
Qi Zeng	University of Illinois at Urbana-Champaign, USA
Shuang (Sophie) Zhai	University of Oklahoma, USA
Danqing Zhang	Amazon, USA
Jiajun Zhang	Institute of Automation, Chinese Academy of Sciences, China
Zixuan Zhang	University of Illinois Urbana-Champaign, USA
Han Zhang	Vispek, China
Peng Zhang	Tianjin University, China
Meishan Zhang	Tianjin University, China
Zhifei Zhang	Tongji University, China
Wenxuan Zhang	The Chinese University of Hong Kong, Hong Kong Special Administrative Region of China
Hongming Zhang	Hong Kong University of Science and Technology, Hong Kong Special Administrative Region of China
Biao Zhang	University of Edinburgh, UK
Qi Zhang	Fudan University, China
Liwen Zhang	ShanghaiTech University, China
Chengzhi Zhang	Nanjing University of Science and Technology, China
Iris Zhang	Cornell University, USA
Zhirui Zhang	Alibaba DAMO Academy, China
Shuaicheng Zhang	Virginia Tech, USA
Tongtao Zhang	Siemens Corporate Technology, USA
Dongxu Zhang	University of Massachusetts, Amherst, USA
Xuanyu Zhang	Du Xiaoman Financial, China
Fan Zhang	Tianjin University, China
Hao Zhang	Agency for Science, Technology and Research, Singapore
Zhuosheng Zhang	Shanghai Jiao Tong University, China
Mengjie Zhao	Ludwig-Maximilians-Universität München, Germany
Yufan Zhao	Microsoft, China
Yang Zhao	Institute of Automation, Chinese Academy of Sciences, China
Xiang Zhao	National University of Defense Technology, China
Zhenjie Zhao	Nanjing University of Information Science and Technology, China
Sanqiang Zhao	University of Pittsburgh, USA
Lulu Zhao	Beijing University of Posts and Telecommunications, China
Kai Zhao	Google, USA
Jie Zhao	Amazon, USA
Renjie Zheng	Baidu Research, USA
Chen Zheng	Michigan State University, USA
Ming Zhong	University of Illinois at Urbana-Champaign, USA

Wanjun Zhong	Sun Yat-sen University, China
Junsheng Zhou	Nanjing Normal University, China
Wenxuan Zhou	University of Southern California, USA
Guangyou Zhou	Central China Normal University, China
Qingyu Zhou	Tencent, China
Ben Zhou	University of Pennsylvania, USA
Xian Zhou	Information Research Center of Military Science, China
Jie Zhu	Soochow University, China
Muhua Zhu	Tencent, China
Junnan Zhu	Institute of Automation, Chinese Academy of Sciences, China
Yazhou Zhang	Zhengzhou University of Light Industry, China
Shi Zong	Nanjing University, China
Wangchunshu Zhou	Beihang University, China

Organizers

Organized by

China Computer Federation, China



Hosted by

Shandong University



In cooperation with



Lecture Notes in Computer Science



Springer



ACTA Scientiarum Naturalium Universitatis Pekinensis

Sponsoring Institutions

Diamond Sponsors

China Mobile



Alibaba



AISpeech



ARC



Platinum Sponsors

Microsoft



ByteDance



Baidu



GTCOM



Huawei



Data Grand

达观数据
DATA GRAND

LaiYe



Hisense

Hisense

Tencent AI Lab



Golden Sponsors

Gridsum

GRIDSUM 国双

Sougou

 搜狗搜索

Xiaomi



Silver Sponsors

NiuTrans



Leyan Technologies



Contents – Part I

Oral - Fundamentals of NLP

Coreference Resolution: Are the Eliminated Spans Totally Worthless?	3
<i>Xin Tan, Longyin Zhang, and Guodong Zhou</i>	
Chinese Macro Discourse Parsing on Dependency Graph Convolutional Network.	15
<i>Yaxin Fan, Feng Jiang, Xiaomin Chu, Peifeng Li, and Qiaoming Zhu</i>	
Predicting Categorial Sememe for English-Chinese Word Pairs via Representations in Explainable Sememe Space	27
<i>Baoju Liu, Lei Hou, Xin Lv, Juanzi Li, and Jinghui Xiao</i>	
Multi-level Cohesion Information Modeling for Better Written and Dialogue Discourse Parsing	40
<i>Jinfeng Wang, Longyin Zhang, and Fang Kong</i>	
ProPC: A Dataset for In-Domain and Cross-Domain Proposition Classification Tasks.	53
<i>Mengyang Hu, Pengyuan Liu, Lin Bo, Yuting Mao, Ke Xu, and Wentao Su</i>	
CTRD: A Chinese Theme-Rheme Discourse Dataset	65
<i>Biao Fu, Yiqi Tong, Dawei Tian, Yidong Chen, Xiaodong Shi, and Ming Zhu</i>	
Machine Translation and Multilinguality	
Learning to Select Relevant Knowledge for Neural Machine Translation	79
<i>Jian Yang, Juncheng Wan, Shuming Ma, Haoyang Huang, Dongdong Zhang, Yong Yu, Zhoujun Li, and Furu Wei</i>	
Contrastive Learning for Machine Translation Quality Estimation	92
<i>Hui Huang, Hui Di, Jian Liu, Yufeng Chen, Kazushige Ouchi, and Jinan Xu</i>	
Sentence-State LSTMs For Sequence-to-Sequence Learning	104
<i>Xuefeng Bai, Yafu Li, Zhirui Zhang, Mingzhou Xu, Boxing Chen, Weihua Luo, Derek Wong, and Yue Zhang</i>	

Guwen-UNILM: Machine Translation Between Ancient and Modern Chinese Based on Pre-Trained Models	116
<i>Zinong Yang, Ke-jia Chen, and Jingqiang Chen</i>	
Adaptive Transformer for Multilingual Neural Machine Translation	129
<i>Junpeng Liu, Kaiyu Huang, Jiuyi Li, Huan Liu, and Degen Huang</i>	
Improving Non-autoregressive Machine Translation with Soft-Masking	141
<i>Shuheng Wang, Shumin Shi, and Heyan Huang</i>	
 Machine Learning for NLP	
AutoNLU: Architecture Search for Sentence and Cross-sentence Attention Modeling with Re-designed Search Space	155
<i>Wei Zhu</i>	
AutoTrans: Automating Transformer Design via Reinforced Architecture Search	169
<i>Wei Zhu, Xiaoling Wang, Yuan Ni, and Guotong Xie</i>	
A Word-Level Method for Generating Adversarial Examples Using Whole-Sentence Information	183
<i>Yufei Liu, Dongmei Zhang, Chunhua Wu, and Wei Liu</i>	
RAST: A Reward Augmented Model for Fine-Grained Sentiment Transfer	196
<i>Xiaoxuan Hu, Hengtong Zhang, Wayne Xin Zhao, Yaliang Li, Jing Gao, and Ji-Rong Wen</i>	
Pre-trained Language Models for Tagalog with Multi-source Data	210
<i>Shengyi Jiang, Yingwen Fu, Xiaotian Lin, and Nankai Lin</i>	
Accelerating Pretrained Language Model Inference Using Weighted Ensemble Self-distillation	224
<i>Jun Kong, Jin Wang, and Xuejie Zhang</i>	
 Information Extraction and Knowledge Graph	
Employing Sentence Compression to Improve Event Coreference Resolution	239
<i>Xinyu Chen, Sheng Xu, Peifeng Li, and Qiaoming Zhu</i>	
BRCEA: Bootstrapping Relation-Aware Cross-Lingual Entity Alignment	251
<i>Yujing Zhang, Feng Zhou, and Xiaoyong Y. Li</i>	
Employing Multi-granularity Features to Extract Entity Relation in Dialogue	262
<i>Qiqi Wang and Peifeng Li</i>	

Attention Based Reinforcement Learning with Reward Shaping for Knowledge Graph Reasoning	275
<i>Sheng Wang, Xiaoying Chen, and Shengwu Xiong</i>	
Entity-Aware Relation Representation Learning for Open Relation Extraction	288
<i>Zihao Liu, Yan Zhang, Huižhen Wang, and Jingbo Zhu</i>	
ReMERT: Relational Memory-Based Extraction for Relational Triples.	300
<i>Chongshuai Zhao, Xudong Dai, Lin Feng, and Peng Liu</i>	
Recognition of Nested Entity with Dependency Information	312
<i>Yu Xia and Fang Kong</i>	
HAIN: Hierarchical Aggregation and Inference Network for Document- Level Relation Extraction.	325
<i>Nan Hu, Taolin Zhang, Shuangji Yang, Wei Nong, and Xiaofeng He</i>	
Incorporate Lexicon into Self-training: A Distantly Supervised Chinese Medical NER	338
<i>Zhen Gan, Zhucong Li, Baoli Zhang, Jing Wan, Yubo Chen, Kang Liu, Jun Zhao, Yafei Shi, and Shengping Liu</i>	
Summarization and Generation	
Diversified Paraphrase Generation with Commonsense Knowledge Graph . . .	353
<i>Xinyao Shen, Jiangjie Chen, and Yanghua Xiao</i>	
Explore Coarse-Grained Structures for Syntactically Controllable Paraphrase Generation	365
<i>Erguang Yang, Mingtong Liu, Deyi Xiong, Yujie Zhang, Yao Meng, Changjian Hu, Jinan Xu, and Yufeng Chen</i>	
Chinese Poetry Generation with Metrical Constraints.	377
<i>Yingfeng Luo, Changliang Li, Canan Huang, Chen Xu, Xin Zeng, Binghao Wei, Tong Xiao, and Jingbo Zhu</i>	
CNewSum: A Large-Scale Summarization Dataset with Human-Annotated Adequacy and Deducibility Level	389
<i>Danqing Wang, Jiaze Chen, Xianze Wu, Hao Zhou, and Lei Li</i>	
Question Generation from Code Snippets and Programming Error Messages	401
<i>Bolun Yao, Wei Chen, Yeyun Gong, Bartuer Zhou, Jin Xie, Zhongyu Wei, Biao Cheng, and Nan Duan</i>	
Extractive Summarization of Chinese Judgment Documents via Sentence Embedding and Memory Network.	413
<i>Yan Gao, Zhengtao Liu, Juan Li, and Jin Tang</i>	

Question Answering

ThinkTwice: A Two-Stage Method for Long-Text Machine Reading Comprehension	427
<i>Mengxing Dong, Bowei Zou, Jin Qian, Rongtao Huang, and Yu Hong</i>	
EviDR: Evidence-Emphasized Discrete Reasoning for Reasoning Machine Reading Comprehension	439
<i>Yongwei Zhou, Junwei Bao, Haipeng Sun, Jiahui Liang, Youzheng Wu, Xiaodong He, Bowen Zhou, and Tiejun Zhao</i>	

Dialogue Systems

Knowledge-Grounded Dialogue with Reward-Driven Knowledge Selection	455
<i>Shilei Liu, Xiaofeng Zhao, Bochao Li, and Feiliang Ren</i>	
Multi-intent Attention and Top-k Network with Interactive Framework for Joint Multiple Intent Detection and Slot Filling	467
<i>Xu Jia, Jiaxin Pan, Youliang Yuan, and Min Peng</i>	
Enhancing Long-Distance Dialogue History Modeling for Better Dialogue Ellipsis and Coreference Resolution.	480
<i>Zixin Ni and Fang Kong</i>	
Exploiting Explicit and Inferred Implicit Personas for Multi-turn Dialogue Generation	493
<i>Ruifang Wang, Ruifang He, Longbiao Wang, Yuke Si, Huanyu Liu, Haocheng Wang, and Jianwu Dang</i>	
Few-Shot NLU with Vector Projection Distance and Abstract Triangular CRF	505
<i>Su Zhu, Lu Chen, Ruisheng Cao, Zhi Chen, Qingliang Miao, and Kai Yu</i>	
Cross-domain Slot Filling with Distinct Slot Entity and Type Prediction	517
<i>Shudong Liu, Peijie Huang, Zhanbiao Zhu, Hualin Zhang, and Jianying Tan</i>	

Social Media and Sentiment Analysis

Semantic Enhanced Dual-Channel Graph Communication Network for Aspect-Based Sentiment Analysis.	531
<i>Zehao Yan, Shiguan Pang, and Yun Xue</i>	

Highway-Based Local Graph Convolution Network for Aspect Based Sentiment Analysis	544
<i>Shiguan Pang, Zehao Yan, Weihao Huang, Bixia Tang, Anan Dai, and Yun Xue</i>	
Dual Adversarial Network Based on BERT for Cross-domain Sentiment Classification	557
<i>Shaokang Zhang, Xu Bai, Lei Jiang, and Huailiang Peng</i>	
Syntax and Sentiment Enhanced BERT for Earliest Rumor Detection	570
<i>Xin Miao, Dongning Rao, and Zhihua Jiang</i>	
Aspect-Sentiment-Multiple-Opinion Triplet Extraction	583
<i>Fang Wang, Yuncong Li, Sheng-hua Zhong, Cunxiang Yin, and Yancheng He</i>	
Locate and Combine: A Two-Stage Framework for Aspect-Category Sentiment Analysis	595
<i>Yang Wu, Zhenyu Zhang, Yanyan Zhao, and Bing Qin</i>	
Emotion Classification with Explicit and Implicit Syntactic Information	607
<i>Nan Chen, Qingrong Xia, Xiabing Zhou, Wenliang Chen, and Min Zhang</i>	
MUMOR: A Multimodal Dataset for Humor Detection in Conversations	619
<i>Jiaming Wu, Hongfei Lin, Liang Yang, and Bo Xu</i>	
NLP Applications and Text Mining	
Advertisement Extraction from Content Marketing Articles via Segment-Aware Sentence Classification	631
<i>Xiaoming Fan and Chenxu Wang</i>	
Leveraging Lexical Common-Sense Knowledge for Boosting Bayesian Modeling	643
<i>Yashen Wang</i>	
Aggregating Inter-viewpoint Relationships of User’s Review for Accurate Recommendation	652
<i>Xingchen He, Yidong Chen, Guocheng Zhang, and Xuling Zheng</i>	
A Residual Dynamic Graph Convolutional Network for Multi-label Text Classification	664
<i>Bingquan Wang, Jie Liu, Shaowei Chen, Xiao Ling, Shanpeng Wang, Wenzheng Zhang, Liyi Chen, and Jiaxin Zhang</i>	
Sentence Ordering by Context-Enhanced Pairwise Comparison	676
<i>Haowei Du, Jizhi Tang, and Dongyan Zhao</i>	

A Dual-Attention Neural Network for Pun Location and Using Pun-Gloss Pairs for Interpretation	688
<i>Shen Liu, Meirong Ma, Hao Yuan, Jianchao Zhu, Yuanbin Wu, and Man Lan</i>	
A Simple Baseline for Cross-Domain Few-Shot Text Classification	700
<i>Chen Zhang and Dawei Song</i>	
Shared Component Cross Punctuation Clauses Recognition in Chinese	709
<i>Xiang Liu, Ruiyang Han, Shuxin Li, Yujiao Han, Mingming Zhang, Zhilin Zhao, and Zhiyong Luo</i>	
BERT-KG: A Short Text Classification Model Based on Knowledge Graph and Deep Semantics	721
<i>Yuyanzhen Zhong, Zhiyang Zhang, Weiqi Zhang, and Juyi Zhu</i>	
Uncertainty-Aware Self-paced Learning for Grammatical Error Correction	734
<i>Kai Dang, Jiaying Xie, Jie Liu, and Shaowei Chen</i>	
Metaphor Recognition and Analysis via Data Augmentation	746
<i>Liang Yang, Jingjie Zeng, Shuqun Li, Zhexu Shen, Yansong Sun, and Hongfei Lin</i>	
Exploring Generalization Ability of Pretrained Language Models on Arithmetic and Logical Reasoning	758
<i>Cunxiang Wang, Boyuan Zheng, Yuchen Niu, and Yue Zhang</i>	
Multimodality and Explainability	
Skeleton-Based Sign Language Recognition with Attention-Enhanced Graph Convolutional Networks	773
<i>Wuyan Liang and Xiaolong Xu</i>	
XGPT: Cross-modal Generative Pre-Training for Image Captioning	786
<i>Qiaolin Xia, Haoyang Huang, Nan Duan, Dongdong Zhang, Lei Ji, Zhifang Sui, Edward Cui, Taroon Bharti, and Ming Zhou</i>	
An Object-Extensible Training Framework for Image Captioning	798
<i>Yike Wu, Ying Zhang, and Xiaojie Yuan</i>	
Relation-Aware Multi-hop Reasoning for Visual Dialog	810
<i>Yao Zhao, Lu Chen, and Kai Yu</i>	
Multi-modal Sarcasm Detection Based on Contrastive Attention Mechanism	822
<i>Xiaoqiang Zhang, Ying Chen, and Guangyuan Li</i>	
Author Index	835

Contents – Part II

Posters - Fundamentals of NLP

Syntax and Coherence - The Effect on Automatic Argument Quality Assessment	3
<i>Xichen Sun, Wenhan Chao, and Zhunchen Luo</i>	
ExperienceGen 1.0: A Text Generation Challenge Which Requires Deduction and Induction Ability	13
<i>Hu Zhang, Pengyuan Liu, Dong Yu, and Sanle Zhang</i>	

Machine Translation and Multilinguality

SynXLM-R: Syntax-Enhanced XLM-R in Translation Quality Estimation	27
<i>Bin Ni, Xiaolei Lu, and Yiqi Tong</i>	

Machine Learning for NLP

Memetic Federated Learning for Biomedical Natural Language Processing	43
<i>Xinya Zhou, Conghui Tan, Di Jiang, Bosen Zhang, Si Li, Yajing Xu, Qian Xu, and Sheng Gao</i>	

Information Extraction and Knowledge Graph

Event Argument Extraction via a Distance-Sensitive Graph Convolutional Network	59
<i>Lu Dai, Bang Wang, Wei Xiang, and Yijun Mo</i>	
Exploit Vague Relation: An Augmented Temporal Relation Corpus and Evaluation	73
<i>Liang Wang, Sheng Xu, Peifeng Li, and Qiaoming Zhu</i>	
Searching Effective Transformer for Seq2Seq Keyphrase Generation	86
<i>Yige Xu, Yichao Luo, Yicheng Zhou, Zhengyan Li, Qi Zhang, Xipeng Qiu, and Xuanjing Huang</i>	

Prerequisite Learning with Pre-trained Language and Graph Embedding Models	98
<i>Bangqi Li , Boci Peng , Yifeng Shao, and Zhichun Wang</i>	

Summarization and Generation

- Variational Autoencoder with Interactive Attention for Affective Text Generation 111
Ruijun Chen, Jin Wang, and Xuejie Zhang

- CUSTOM: Aspect-Oriented Product Summarization for E-Commerce 124
Jiahui Liang, Junwei Bao, Yifan Wang, Youzheng Wu, Xiaodong He, and Bowen Zhou

Question Answering

- FABERT: A Feature Aggregation BERT-Based Model for Document Reranking 139
Xiaozhi Zhu, Leung-Pun Wong, Lap-Kei Lee, Hai Liu, and Tianyong Hao
- Generating Relevant, Correct and Fluent Answers in Natural Answer Generation 151
Yongjie Huang, Meng Yang, and Ni Yang

- GeoCQA: A Large-Scale Geography-Domain Chinese Question Answering Dataset from Examination 163
Zhen Cui, Bin Wang, and Jiangzhou Ju

Dialogue Systems

- Generating Informative Dialogue Responses with Keywords-Guided Networks 179
Heng-Da Xu, Xian-Ling Mao, Zewen Chi, Fanshu Sun, Jingjing Zhu, and Heyan Huang
- Zero-Shot Deployment for Cross-Lingual Dialogue System 193
Lu Xiang, Yang Zhao, Junnan Zhu, Yu Zhou, and Chengqing Zong
- MultiWOZ 2.3: A Multi-domain Task-Oriented Dialogue Dataset Enhanced with Annotation Corrections and Co-Reference Annotation 206
Ting Han, Ximing Liu, Ryuichi Takanabu, Yixin Lian, Chongxuan Huang, Dazhen Wan, Wei Peng, and Minlie Huang
- EmoDialoGPT: Enhancing DialoGPT with Emotion 219
Yuxiang Jia, Shuai Cao, Changyong Niu, Yutuan Ma, Hongying Zan, Rui Chao, and Weicong Zhang

Social Media and Sentiment Analysis

BERT-Based Meta-Learning Approach with Looking Back for Sentiment Analysis of Literary Book Reviews	235
<i>Hui Bao, Kai He, Xuemeng Yin, Xuanyu Li, Xinrui Bao, Haichuan Zhang, Jialun Wu, and Zeyu Gao</i>	
ISWR: An Implicit Sentiment Words Recognition Model Based on Sentiment Propagation	248
<i>Qizhi Li, Xianyong Li, Yajun Du, and Xiaoliang Chen</i>	

An Aspect-Centralized Graph Convolutional Network for Aspect-Based Sentiment Classification	260
<i>Weixiang Zhao, Yanyan Zhao, Xin Lu, and Bing Qin</i>	

NLP Applications and Text Mining

Capturing Global Informativeness in Open Domain Keyphrase Extraction	275
<i>Si Sun, Zhenghao Liu, Chenyan Xiong, Zhiyuan Liu, and Jie Bao</i>	
Background Semantic Information Improves Verbal Metaphor Identification	288
<i>Wentao Qin and Dongyan Zhao</i>	

Multimodality and Explainability

Towards Unifying the Explainability Evaluation Methods for NLP	303
<i>Diana Lucaci and Diana Inkpen</i>	

Explainable AI Workshop

Detecting Covariate Drift with Explanations	317
<i>Steffen Castle, Robert Schwarzenberg, and Mohsen Pourvali</i>	
A Data-Centric Approach Towards Deducing Bias in Artificial Intelligence Systems for Textual Contexts	323
<i>Shipra Jain, Sreya Dey, Prateek Bajaj, and A. S. Rohit Kumar</i>	

Student Workshop

Enhancing Model Robustness via Lexical Distilling	337
<i>Wentao Qin and Dongyan Zhao</i>	
Multi-stage Multi-modal Pre-training for Video Representation	345
<i>Chunquan Chen, Lujia Bao, Weikang Li, Xiaoshuai Chen, Xinghai Sun, and Chao Qi</i>	

Nested Causality Extraction on Traffic Accident Texts as Question Answering	354
<i>Gongxue Zhou, Weifeng Ma, Yifei Gong, Liudi Wang, Yaru Li, and Yulai Zhang</i>	
Evaluation Workshop	
MSDF: A General Open-Domain Multi-skill Dialog Framework	365
<i>Yu Zhao, Xinshuo Hu, Yunxin Li, Baotian Hu, Dongfang Li, Sichao Chen, and Xiaolong Wang</i>	
RoKGDS: A Robust Knowledge Grounded Dialog System.	377
<i>Jun Zhang, Yuxiang Sun, Yushi Zhang, Weijie Xu, Jiahao Ying, Yan Yang, Man Lan, Meirong Ma, Hao Yuan, and Jianchao Zhu</i>	
Enhanced Few-Shot Learning with Multiple-Pattern-Exploiting Training	388
<i>Jiali Zeng, Yufan Jiang, Shuangzhi Wu, and Mu Li</i>	
BIT-Event at NLPCC-2021 Task 3: Subevent Identification via Adversarial Training	400
<i>Xiao Liu, Ge Shi, Bo Wang, Changsen Yuan, Heyan Huang, Chong Feng, and Lifang Wu</i>	
Few-Shot Learning for Chinese NLP Tasks	412
<i>Liang Xu, Xiaojing Lu, Chenyang Yuan, Xuanwei Zhang, Hu Yuan, Huilin Xu, Guoao Wei, Xiang Pan, Junyi Li, Jianlin Su, Zhenyu Yang, Renfen Hu, and Hai Hu</i>	
When Few-Shot Learning Meets Large-Scale Knowledge-Enhanced Pre-training: Alibaba at FewCLUE	422
<i>Ziyun Xu, Chengyu Wang, Peng Li, Yang Li, Ming Wang, Boyu Hou, Minghui Qiu, Chengguang Tang, and Jun Huang</i>	
TKB ² ert: Two-Stage Knowledge Infused Behavioral Fine-Tuned BERT.	434
<i>Jiahao Chen, Zheyu He, Yujin Zhu, and Liang Xu</i>	
A Unified Information Extraction System Based on Role Recognition and Combination.	447
<i>Yadong Zhang and Man Lan</i>	
An Effective System for Multi-format Information Extraction	460
<i>Yaduo Liu, Longhui Zhang, Shujuan Yin, Xiaofeng Zhao, and Feiliang Ren</i>	
A Hierarchical Sequence Labeling Model for Argument Pair Extraction.	472
<i>Jingyi Sun, Qinglin Zhu, Jianzhu Bao, Jipeng Wu, Caihua Yang, Rui Wang, and Ruifeng Xu</i>	

Distant Finetuning with Discourse Relations for Stance Classification	484
<i>Lifeng Jin, Kun Xu, Linfeng Song, and Dong Yu</i>	
The Solution of Xiaomi AI Lab to the 2021 Language and Intelligence Challenge: Multi-format Information Extraction Task.	496
<i>Wen Dai, Xinyu Hua, Rongrong Lv, Ruipeng Bo, and Shuai Chen</i>	
A Unified Platform for Information Extraction with Two-Stage Process	509
<i>Chongshuai Zhao, Dongjie Guo, Xudong Dai, Chengmin Gu, Lingling Fa, and Peng Liu</i>	
Overview of the NLPCC 2021 Shared Task: AutoIE2	519
<i>Weigang Guo, Xuefeng Yang, Xingyu Bai, Taiqiang Wu, Weijie Liu, Zhe Zhao, Qi Ju, and Yujiu Yang</i>	
Task 1 - Argumentative Text Understanding for AI Debater (AIDebater)	530
<i>Yuming Li, Maojin Xia, and Yidong Wang</i>	
Two Stage Learning for Argument Pairs Extraction	538
<i>Shuheng Wang, Zimo Yin, Wei Zhang, Derong Zheng, and Xuan Li</i>	
Overview of Argumentative Text Understanding for AI Debater Challenge	548
<i>Jian Yuan, Liying Cheng, Ruidan He, Yinzi Li, Lidong Bing, Zhongyu Wei, Qin Liu, Chenhui Shen, Shuonan Zhang, Changlong Sun, Luo Si, Changjian Jiang, and Xuanjing Huang</i>	
ACE: A Context-Enhanced Model for Interactive Argument Pair Identification	569
<i>Yi Wu and Pengyuan Liu</i>	
Context-Aware and Data-Augmented Transformer for Interactive Argument Pair Identification	579
<i>Yuanling Geng, Shuqun Li, Fan Zhang, Shaowu Zhang, Liang Yang, and Hongfei Lin</i>	
ARGUABLY @ AI Debater-NLPCC 2021 Task 3: Argument Pair Extraction from Peer Review and Rebuttals	590
<i>Guneet Singh Kohli, Prabsimran Kaur, Muskaan Singh, Tirhankar Ghosal, and Prashant Singh Rana</i>	
Sentence Rewriting for Fine-Tuned Model Based on Dictionary: Taking the Track 1 of NLPCC 2021 Argumentative Text Understanding for AI Debater as an Example	603
<i>Pan He, Yan Wang, and Yanru Zhang</i>	

Knowledge Enhanced Transformers System for Claim Stance Classification	614
<i>Xiangyang Li, Zheng Li, Sujian Li, Zhimin Li, and Shimin Yang</i>	
Author Index	625

Oral - Fundamentals of NLP



Coreference Resolution: Are the Eliminated Spans Totally Worthless?

Xin Tan^(✉), Longyin Zhang, and Guodong Zhou^(✉)

School of Computer Science and Technology, Soochow University, Suzhou, China
 {xtan9,lyzhang9}@stu.suda.edu.cn, gdzhou@suda.edu.cn

Abstract. Up to date, various neural-based methods have been proposed for joint mention span detection and coreference resolution. However, existing studies on coreference resolution mainly depend on mention representations, while the rest spans in the text are largely ignored and directly eliminated. In this paper, we aim at investigating whether those eliminated spans are totally worthless, or to what extent they can help improve the performance of coreference resolution. To achieve this goal, we propose to refine the representation of mentions with global spans including these eliminated ones leveraged. On this basis, we further introduce an additional loss term in this work to encourage the diversity between different entity clusters. Experimental results on the document-level CoNLL-2012 Shared Task English dataset show that the eliminated spans are indeed useful and our proposed approaches show promising results in coreference resolution.

Keywords: Coreference resolution · Representation learning · Document-level cohesion analysis

1 Introduction

As an important role in text understanding, coreference resolution is the task of identifying and clustering mention spans in a text into several clusters where each cluster refers to the same real world entity. With the increasing population of neural networks, varied neural-based approaches have been proposed for coreference resolution [4, 5, 8–13, 23–25] so far. Among these studies, Lee et al. [12] first propose an end-to-end neural model apart from syntactic parsers. After that, the coreference resolution task is much liberated from complicated hand-engineered methods, and more and more studies [9–11, 13, 25] have been proposed to refine the coreference resolution model of Lee et al. [12]. In general, neural-based studies usually perform coreference resolution in two stages: (i) a mention detector to select mention spans from all candidate spans in a document and (ii) a coreference resolver to cluster mention spans into corresponding entity clusters. In this way, the manually annotated cluster labels are well employed for both mention detection and clustering.

Although previous studies have achieved certain success in coreference resolution in recent years, these studies perform mention clustering heavily rely on the mention representations selected at the first stage for mention clustering, while the rest spans in each piece of text are largely filtered and directly eliminated. Considering that, we naturally raise the question: Are those eliminated spans worthless for coreference resolution? On the one hand, the mention spans selected by the mention scoring function are usually isolated with each other in format. Besides, eliminating these “worthless” spans only depending on the mention scoring function may aggravate the notorious error propagation problem in the pipeline workflow of coreference resolution. On the other, previous studies improve the performance of coreference resolution either from feature designing or model architecture perspectives, while the progress of improving on data utilization remains hysteretic. As far as we know, these eliminated spans have not been explored so far. Under this condition, we aim at increasing the data utilization rate and investigating to what extent these eliminated spans can help improve the performance of coreference resolution.

To achieve the above goal, based on the two-stage neural model of Lee et al. [12], we propose a mention representation refining strategy to well leverage the spans that highly related to the mention for representation enhancing. Following this way, the contribution of our approach is two-fold: (i) using the global spans (both mention spans and the eliminated spans) with high utilization rate to provide auxiliary information for mention representation enhancing; (ii) equipping the isolated mention representations with context-aware correlations through the trade-off among global spans in the document. In addition, to make full use of the annotated training instances, we utilize an additional loss term to learn from both positive and negative samples to encourage the diversity between different entity clusters. Notably, we also explore the effects of two different contextualized word embeddings (i.e., ELMo [17] and BERT [7]) in this paper for comparison. Experimental results on the document-level CoNLL-2012 English dataset show that our way of reusing these eliminated spans is quite useful for the coreference resolution task, and our approach shows promising results when compared with previous state-of-the-art methods.

2 Background

Task Definition. Following previous studies [8–13, 25], we cast the task of coreference resolution as an antecedent selection problem, where each span is assigned with an antecedent in the document. Specifically, given a span i , the possible antecedents are $Y_i = \{\epsilon, 1, 2, \dots, i - 1\}$ (*i.e.*, a dummy antecedent ϵ and all preceding spans). The non-dummy antecedent refers to the coreference link between span i and its antecedent y_i ($y_i \neq \epsilon$). The dummy antecedent denotes the coreference link between span i and ϵ , which represents two possible scenarios: (i) span i is not an entity mention or (ii) span i is an entity mention but not coreferent with any previous span.

Baseline. In this work, the baseline model [12] we use for coreference resolution learns a distribution $P(y_i)$ over the antecedents of each span i :

$$P(y_i) = \frac{e^{S(i,y_i)}}{\sum_{y' \in Y(i)} e^{S(i,y')}} \quad (1)$$

where $S(i,j)$ represents a pairwise score for the coreference link between span i and span j . And the pairwise coreference score $S(i,j)$ is calculated based on the mention scores of i and j (i.e., S_i^m and S_j^m which denote whether the spans i and j are mentions or not) and the joint compatibility score of i and j (i.e., $S_{i,j}^a$ which denotes whether mention j is an antecedent of mention i or not). And the final pairwise coreference score is written as:

$$S(i,j) = S_i^m + S_j^m + S_{i,j}^a \quad (2)$$

Given the vector representation g_i for each possible span i , the mention score of span i and the antecedent score between spans i and j can be calculated as:

$$S_i^m = W_m \cdot \text{FFNN}_m(g_i) \quad (3)$$

$$S_{i,j}^a = W_a \cdot \text{FFNN}_a([g_i, g_j, g_i \circ g_j, \phi(i, j)]) \quad (4)$$

where g_i is obtained via bidirectional LSTM models that learn context-dependent boundary and head representations, W_m and W_a denote two learnable parameter matrixes, \circ denotes the element-wise multiplication, $\text{FFNN}(\cdot)$ denotes a feed-forward neural network, and the antecedent score $S_{i,j}^a$ is calculated through explicit element-wise similarity of each span, $g_i \circ g_j$, and a feature vector $\phi(i, j)$ that encodes speaker and genre information from the metadata and the distance between spans i and j .

3 Coreference Resolution with Enhanced Mention Representation

Motivated by previous studies [11, 12, 25], we inherit the architecture that combines mention detection and coreference scoring for coreference resolution. Particularly, as stated before, we propose to reuse the spans that are eliminated at the mention detection stage to enhance the representation of mention spans for better coreference resolution performance.

3.1 Mention Detection

At the mention detection stage, we take the text spans within a certain length limitation as potential mention spans. Following previous studies, we take word-, character- and context-level information for span representation [12]. Moreover, we also incorporate the syntactic structural information (e.g. the path, siblings, degrees, and category of the current node) for representation enhancing [11]:

$$\mathbf{s}_i = [\mathbf{h}_{bi}, \mathbf{h}_{ei}, \hat{\mathbf{x}}_i, \mathbf{f}_i] \quad (5)$$

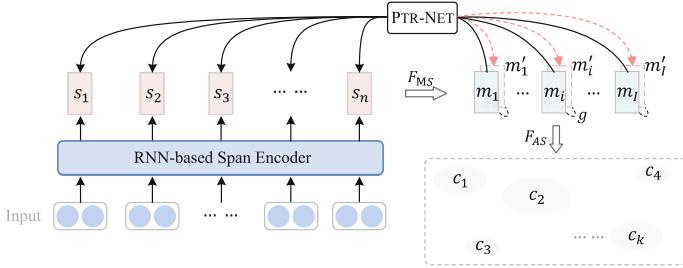


Fig. 1. Mention representation refining for coreference resolution. Here, F_{MS} and F_{AS} refer to the mention scoring function and antecedent scoring function respectively, and c_k denotes an independent entity cluster.

where \mathbf{h}_i is the contextual representation of input $x_i = [w_i, c_i]$ (w_i means the word embedding vector and c_i means the character embedding vector). $\hat{\mathbf{x}}_i$ is the weighted sum of the word representations that contained in the span i , where the attention weights are learned through the Head-finding attention mechanism, and one can refer to [12] for the detailed process of calculating $\hat{\mathbf{x}}_i$. And \mathbf{f}_i denotes the syntactic structural feature vector.

After that, we use a feedforward scoring function for mention determination as Lee et al. [12] did:

$$S_i^m = W_m \cdot \text{FFNN}_m(\mathbf{s}_i) \quad (6)$$

Then, the spans that assigned with high attention scores are selected as the resulting mention spans, noted by \mathbf{m}_i .

3.2 Coreference Resolving with Global Spans Perceived

After obtaining the mention representation, the following coreference scoring stage aims to determine the antecedent for each mention. In order to reduce the problem of error propagation in the pipeline workflow of coreference resolution, we propose to further enhance the mention representation with context-aware correlations. To achieve this, we propose to extract information from both mentions and all the eliminated spans to help refine the obtained mention representations, as illustrated in Fig. 1.

Concretely, we use a pointer net (PTR-NET) [21] in this work to extract the context-aware information from the spans that are strongly related to the current mention. More formally, we extract context information and project it into a parallel mention vector space as:

$$u_{i,t} = V^T \tanh(W_1 \mathbf{s}_t + W_2 \mathbf{m}_i) \quad (7)$$

$$\alpha_i = \text{softmax}(\mathbf{u}_i) \quad (8)$$

$$\mathbf{m}'_i = \sum \alpha_{i,t} \cdot \mathbf{s}_t, t \in \{1, \dots, n\} \quad (9)$$

where \mathbf{s}_t denotes the representation of the t -th span, \mathbf{m}_i refers to the representation of the i -th mention, V^T , W_1 and W_2 are learnable parameter matrixes,

$\alpha_{i,t}$ denotes the relevancy of each span, and \mathbf{m}'_i denotes the context-aware representation for the mention i in the parallel mention vector space. In this manner, (i) the spans that strongly related to the current mention are well leveraged to exploit context information and (ii) the correlations between eliminated spans and mentions are well learned through the trade-off among spans in the text during the attention calculation process.

After achieving the parallel mention representations \mathbf{m}_i and \mathbf{m}'_i , we add a gated model between them for information communication, corresponding to g in Fig. 1. And the enhanced mention representation is formulated as:

$$f_i = \delta(W_f[\mathbf{m}_i, \mathbf{m}'_i]) \quad (10)$$

$$\mathbf{m}_i^* = f_i \circ \mathbf{m}_i + (1 - f_i) \circ \mathbf{m}'_i \quad (11)$$

where f_i is a learnable vector for information filtering, \circ denotes the element-wise multiplication, and \mathbf{m}_i^* refers to the integrated mention representation.

Then, the pairwise antecedent score is calculated based on the integrated mention pairs. Following previous work [12], the coreference score is generated by summing up the mention score and the pairwise antecedent score as:

$$S_{i,j}^a = W_a \cdot \text{FFNN}_a([\mathbf{m}_i^*, \mathbf{m}_j^*, \mathbf{m}_i^* \circ \mathbf{m}_j^*, \phi(i,j)]) \quad (12)$$

$$S(i,j) = S_i^m + S_j^m + S_{i,j}^a \quad (13)$$

where $S_{i,j}^a$ denotes the antecedent score between spans i and j , and $S(i,j)$ denotes the final coreference score.

4 Model Training

Previous studies usually cluster candidate mentions only depending on gold cluster labels. Differently, based on the loss objective of Lee et al. [12], we introduce an additional loss term to maximize the distance between different entity clusters to encourage the diversity. Concretely, we make the loss term learn from both positive and negative samples, and the loss objective can be formulated as:

$$\mathcal{L} = -\log \prod_{i=1}^N \sum_{\hat{y} \in Y(i) \cap \text{GOLD}(i)} P(\hat{y}) + \log \prod_{i=1}^N \sum_{\hat{y} \notin Y(i) \cap \text{GOLD}(i)} P(\hat{y}) \quad (14)$$

$$P(\hat{y}) = \frac{e^{S(i,\hat{y})}}{\sum_{y' \in Y(i)} e^{S(i,y')}} \quad (15)$$

where $S(i,\hat{y})$ denotes the coreference score of the coreference link between span i and its antecedent \hat{y} , and $\text{GOLD}(i)$ denotes the set of entity mentions in the cluster that contains span i . If span i does not belong to any clusters, or if all gold antecedents have been pruned, $\text{GOLD}(i)$ equals $\{\varepsilon\}$. Through the proposed approach, the training instances are well utilized to cluster entity mentions and at the same time increase the diversity between different mention clusters.

5 Experimentation

5.1 Experimental Settings

Datasets. We carry out several experiments on the English coreference resolution data from the CoNLL-2012 Shared Task [18]. The dataset contains 2802 documents for training, 343 documents for validation, and 348 documents for testing with 7 different genres (i.e., newswire, magazine articles, broadcast news, broadcast conversations, web data, conversational speech data, and the New Testament). All the experimental data can be downloaded at <https://cemanix.org/conll/2012/data.html>.

Model Settings. We used three kinds of word representations for experimentation, i.e., (i) the fixed 300-dimensional GloVe [16] vectors and the 50-dimensional Turian [19] vectors, (ii) the 8-dimensional character embeddings learned from CNNs, where the window sizes of the convolution layers were 3, 4, and 5 characters respectively, and each layer consists of 50 filters, and (iii) two kinds of 1024-dimensional contextualized word representations provided by ELMo [17] and BERT [7]. The model hyper-parameters were directly borrowed from Lee et al. [12] for fair comparison.

Metrics. We report the Precision, Recall, and F_1 scores based on three popular coreference resolution metrics, i.e., MUC [20], B^3 [1], and $CEAF_{\phi 4}$ [14]. And we report the averaged F_1 -score as the final CoNLL score.

5.2 Experimental Results

In this paper, we select the model of Lee et al. [12] as our baseline system. We borrow the system implemented by Kong and Fu [11] for experimentation in two reasons: (i) Kong and Fu [11] incorporate varied syntactic structural features (e.g., the path, siblings, degrees, and category of the current node) into the model of Lee et al. [12] to better capture hierarchical information for span representation. (ii) Their implemented system can well reduce computational complexity whose training speed is 6 times faster than that of Lee et al. [12]. Moreover, we enhance their system by applying the ELMo embedding to it for better comparison. We report the results of the original and enhanced baseline systems of Kong and Fu [11] for performance comparison. Moreover, following previous work, we also present the performances of recent systems for reference, and the overall results are shown in Table 1.

From the results we can find that (i) Comparing our approach (line 10) with the baseline systems (lines 8 and 9), our method performs better on all the three metrics, which suggests the great effectiveness of our proposed method in utilizing all spans including the eliminated ones for coreference resolution. And the contextualized ELMo also helps improve the performance to a certain extent. (ii) Comparing our system with previous state-of-the-art methods, ours

Table 1. Performance comparison on coreference resolution. Sign “†” means the ELMo representation is used and “‡” means the powerful Bert model is employed. Compared with the enhanced baseline system, our performance improvements on F_1 are statistically significant with $p < 0.05$.

	MUC			B^3			CEAF $_{\phi_4}$			Avg. F_1
	Prec.	Rec.	F_1	Prec.	Rec.	F_1	Prec.	Rec.	F_1	
Wu et al. [24] ‡	88.6	87.4	88.0	82.4	82.0	82.2	79.9	78.3	79.1	83.1
Wiseman et al. [23]	77.5	69.8	73.4	66.8	57.0	61.5	62.1	53.9	57.7	64.2
Clark and Manning [5]	79.9	69.3	74.2	71.0	56.5	63.0	63.8	54.3	58.7	65.3
Clark and Manning [4]	79.2	70.4	74.6	69.9	58.0	63.4	63.5	55.5	59.2	65.7
Lee et al. [12]	78.4	73.4	75.8	68.6	61.8	65.0	62.7	59.0	60.8	67.2
Lee et al. [13] †	81.4	79.5	80.4	72.2	69.5	70.8	68.2	67.1	67.6	73.0
Zhang et al. [25]	79.4	73.8	76.5	69.0	62.3	65.5	64.9	58.3	61.4	67.8
Baseline [11]	80.5	73.9	77.1	71.2	61.5	66.0	64.3	61.1	62.7	68.6
Baseline [11] †	80.8	79.3	80.0	71.2	68.7	70.0	66.4	66.3	66.4	72.1
Ours †	81.8	80.0	80.9	72.6	70.0	71.3	68.1	67.9	68.0	73.4
-additional loss	80.9	80.6	80.8	70.6	70.6	70.6	67.2	67.9	67.5	73.0

outperforms most of the systems except the Bert-based model of Wu et al. [24]. (iii) Comparing our system under different model settings (the last two lines) we find that the additional loss term we use can improve the performance of coreference resolution to some extent, especially on the B^3 indicator. The overall results above indicate that our proposed methods are useful and can well increase the utilization rate of the coreference resolution data.

5.3 Analysis on Context-Aware Word Representations

In this subsection, we present our insight on exploring a better pre-trained word representation. More theoretically, we aim to figure out a question: Which kind of context-aware embeddings is better for coreference resolution? With this in mind, we employ two kinds of popular contextualized word representations (i.e., ELMo [17] and BERT [7]) for analysis. Briefly review:

- **ELMo.** Peters et al. [17] hold the view that word representation should contain rich syntactic and semantic information and be able to model polysemous words. On this basis, they provide the contextualized ELMo for word representation by training bidirectional LSTMs on a large-scale corpus.
- **BERT.** Devlin et al. [7] present two new pre-training objectives, i.e., the “masked language model (LM)” for word-level representation and the “next sentence prediction” for sentence-level representation. In essence, Bert achieves a good feature representation of words by running the self-supervised learning method on the basis of massive training data.

Comparing the two kinds of contextualized word embeddings above: (i) ELMo is better for feature-based methods, which transfers specific downstream NLP tasks to the process of pre-training to produce the word representations, so

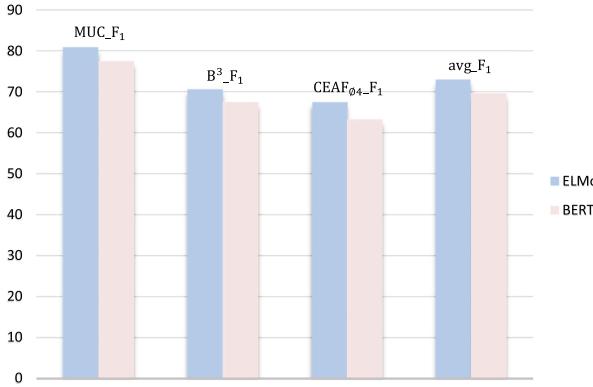


Fig. 2. Performance comparison between ELMo and Bert on coreference resolution.

as to obtain a dynamic word representation that changes constantly according to the context; (ii) Bert prefers fine-tuning methods, which fine-tunes network parameters to produce better word representation for specific downstream NLP tasks. Previous Bert-based work [24] has proven the great effectiveness of fine-tuning the Bert model in coreference resolution. Different from them, in this work, we analyze the effects of static features extracted by pre-trained LMs. To achieve this, we employ ELMo and Bert models to generate static context-aware embeddings and then apply them to the coreference resolution task. Performance comparison between the two kinds of word vectors are detailed in Fig. 2.

Usually, the context features generated by ELMo are known to be shallower than that generated by Bert. But the results show that the Bert-based coreference resolution system performs relatively worse than the ELMo-based one. In other words, the shallow-level features defeat deep-level ones in our coreference resolution experiments. One possible reason for this result is that the ELMo-based shallow context features are more easily understood and utilized by the resolution system while the Bert-based deep context features could be too much for the model to use directly. Nevertheless, fine-tuning the Bert model can transform the informative and complicated context information into a task-specific one for better understanding, which explains the recent success in fine-tuning Bert for better coreference resolution [24].

5.4 Case Study

To qualitatively analyze the mention representation refining process, we provide a visualization of the span pointing process, as shown in Fig. 3. From the example, there exist two entity clusters with different background colors. Here, we present the top three spans that are particularly relevant to each mention (i.e., M_1 and M_2) for reference. Obviously, both mentions, M_1 and M_2 , are pronouns and they pay much attention to those spans that semantically related to them (e.g., Violence between Israelis and Palestinians). However, most of

these spans with high scores are eliminated during the mention detection stage in previous studies, which much hinders the interaction between mentions and mention-related spans. Fortunately, in this work, the pointer mechanism [21] is well utilized to detect these mention-related spans, and the selected spans are reused for mention representation refining. It is worth noting that mention representation refining is effective especially for pronoun mentions with ambiguous meanings, e.g., the two mentions *its* and *them* in the example. And the proposed method can give these mentions more accurate context-aware representation for better coreference resolution.

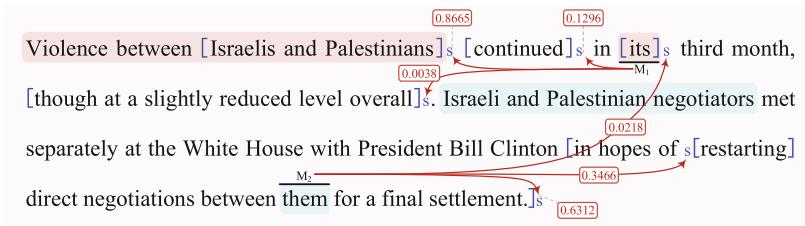


Fig. 3. Example of the mention-related span selection process, where text fragments of the same color (“Violence ...” and “its” are colored in orange; “Israeli ...” and “them” are colored in blue) belong to the same cluster. And the solid lines denote the scores of the most possible three spans assigned for each mention.

6 Related Work

Before the population of neural networks, traditional machine learning methods have a long history in the coreference resolution task. In general, three mainstream methods have been proposed: 1) Mention-pair models [2, 15] to determine if a pair of mentions are coreferent by training binary classifiers. 2) Mention-ranking models [4, 6, 22] to score all previous candidate mentions for current mention and select the most possible one as its antecedent. 3) Entity-mention models [3, 5, 23] to determine whether the current mention is coreferent with a preceding, partially-formed mention cluster. Although these methods have achieved significant performance gains, they still suffer from one major drawback: their complicated hand-engineered rules are difficult to adapt to new languages.

With the rapid spread of neural network in recent years, varied researchers turned to neural-based methods. Lee et al. [12] proposed the first end-to-end neural-based system that liberates coreference resolution from the complicated hand-engineered methods. Zhang et al. [25] proposed to improve the performance of coreference resolution by using a biaffine attention model to score antecedents and jointly optimize the two sub-tasks. Lee et al. [13] proposed an approximation of higher-order inference using the span-ranking architecture from Lee et al. [12] in an iterative manner. Kong and Fu [11] proposed to improve the

resolution performance by incorporating various kinds of structural information into their model. Kantor and Globerson [10] proposed to capture properties of entity clusters and use them during the resolution process. Most recently, Fei et al. [8] presented the first end-to-end reinforcement learning based coreference resolution model. Joshi et al. [9] and Wu et al. [24] proposed to improve the performance of coreference resolution with the help of the state-of-the-art Bert.

7 Conclusion

In this paper, we aim at increasing the data utilization rate and exploring the value of those spans eliminated at the mention detection stage. On this basis, we proposed a mention representation refining strategy where spans that highly related to the mentions are well leveraged through a pointer network for representation enhancing. Moreover, we introduced an additional loss term to encourage the diversity between different entity clusters. Notably, we also performed experiments on different contextualized word embeddings to explore the effectiveness of them on coreference resolution. Experimental results on the document-level CoNLL-2012 Shared Task English dataset indicate that these eliminated spans are indeed useful and our proposed approach can achieve much better results when compared with the baseline systems and competitive results when compared with most previous studies in coreference resolution.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (NSFC) via Grant Nos. 62076175, 61876118, and the Postgraduate Research & Practice Innovation Program of Jiangsu Province under Grant No. KYCX20_2669. Also, we would like to thank the anonymous reviewers for their insightful comments.

References

1. Bagga, A., Baldwin, B.: Algorithms for scoring coreference chains. In: The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference, vol. 1, pp. 563–566. Granada (1998)
2. Bengtson, E., Roth, D.: Understanding the value of features for coreference resolution. In: Proceedings of the 2008 Conference on EMNLP, pp. 294–303. Association for Computational Linguistics, Honolulu, Hawaii, October 2008. <https://www.aclweb.org/anthology/D08-1031>
3. Clark, K., Manning, C.D.: Entity-centric coreference resolution with model stacking. In: Proceedings of the 53rd Annual Meeting of the ACL and the 7th IJCNLP (Volume 1: Long Papers), pp. 1405–1415. Association for Computational Linguistics, Beijing, China, July 2015. <https://doi.org/10.3115/v1/P15-1136>, <https://www.aclweb.org/anthology/P15-1136>
4. Clark, K., Manning, C.D.: Deep reinforcement learning for mention-ranking coreference models. arXiv preprint [arXiv:1609.08667](https://arxiv.org/abs/1609.08667) (2016)

5. Clark, K., Manning, C.D.: Improving coreference resolution by learning entity-level distributed representations. In: Proceedings of the 54th Annual Meeting of the ACL (Volume 1: Long Papers), pp. 643–653. Association for Computational Linguistics, Berlin, Germany, August 2016. <https://doi.org/10.18653/v1/P16-1061>, <https://www.aclweb.org/anthology/P16-1061>
6. Denis, P., Baldridge, J.: A ranking approach to pronoun resolution. In: International Joint Conferences on Artificial Intelligence, vol. 158821593 (2007)
7. Devlin, J., Chang, M., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv: Computation and Language* (2018)
8. Fei, H., Li, X., Li, D., Li, P.: End-to-end deep reinforcement learning based coreference resolution. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 660–665. Association for Computational Linguistics, Florence, Italy, July 2019. <https://doi.org/10.18653/v1/P19-1064>, <https://www.aclweb.org/anthology/P19-1064>
9. Joshi, M., Levy, O., Zettlemoyer, L., Weld, D.: BERT for coreference resolution: baselines and analysis. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pp. 5807–5812. Association for Computational Linguistics, Hong Kong, China, November 2019. <https://doi.org/10.18653/v1/D19-1588>, <https://www.aclweb.org/anthology/D19-1588>
10. Kantor, B., Globerson, A.: Coreference resolution with entity equalization. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 673–677. Association for Computational Linguistics, Florence, Italy, July 2019. <https://doi.org/10.18653/v1/P19-1066>, <https://www.aclweb.org/anthology/P19-1066>
11. Kong, F., Fu, J.: Incorporating structural information for better coreference resolution. In: Proceedings of the 28th IJCAI, pp. 5039–5045. AAAI Press (2019)
12. Lee, K., He, L., Lewis, M., Zettlemoyer, L.: End-to-end neural coreference resolution. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 188–197. Association for Computational Linguistics, Copenhagen, Denmark, September 2017
13. Lee, K., He, L., Zettlemoyer, L.: Higher-order coreference resolution with coarse-to-fine inference. In: Proceedings of the 2018 Conference of the NAACL: Human Language Technologies, Volume 2 (Short Papers), pp. 687–692. Association for Computational Linguistics, New Orleans, Louisiana, June 2018. <https://doi.org/10.18653/v1/N18-2108>, <https://www.aclweb.org/anthology/N18-2108>
14. Luo, X.: On coreference resolution performance metrics. In: Proceedings of Human Language Technology Conference and Conference on EMNLP, pp. 25–32. Association for Computational Linguistics, Vancouver, British Columbia, Canada, October 2005. <https://www.aclweb.org/anthology/H05-1004>
15. Ng, V., Cardie, C.: Improving machine learning approaches to coreference resolution. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 104–111. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, July 2002. <https://doi.org/10.3115/1073083.1073102>, <https://www.aclweb.org/anthology/P02-1014>
16. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation, pp. 1532–1543 (2014)

17. Peters, M., et al.: Deep contextualized word representations. In: Proceedings of the 2018 Conference of the NAACL: Human Language Technologies, Volume 1 (Long Papers), pp. 2227–2237. Association for Computational Linguistics, New Orleans, Louisiana, June 2018. <https://doi.org/10.18653/v1/N18-1202>, <https://www.aclweb.org/anthology/N18-1202>
18. Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., Zhang, Y.: CoNLL-2012 shared task: modeling multilingual unrestricted coreference in OntoNotes. In: Joint Conference on EMNLP and CoNLL - Shared Task, pp. 1–40. Association for Computational Linguistics, Jeju Island, Korea, July 2012. <https://www.aclweb.org/anthology/W12-4501>
19. Turian, J., Ratinov, L., Bengio, Y.: Word representations: A simple and general method for semi-supervised learning, pp. 384–394 (2010)
20. Vilain, M., Burger, J., Aberdeen, J., Connolly, D., Hirschman, L.: A model-theoretic coreference scoring scheme. In: Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6–8, 1995 (1995). <https://www.aclweb.org/anthology/M95-1005>
21. Vinyals, O., Fortunato, M., Jaitly, N.: Pointer networks. In: Advances in Neural Information Processing Systems, pp. 2692–2700 (2015)
22. Wiseman, S., Rush, A.M., Shieber, S., Weston, J.: Learning anaphoricity and antecedent ranking features for coreference resolution. In: Proceedings of the 53rd Annual Meeting of the ACL and the 7th IJCNLP (Volume 1: Long Papers), pp. 1416–1426. Association for Computational Linguistics, Beijing, China, July 2015. <https://doi.org/10.3115/v1/P15-1137>, <https://www.aclweb.org/anthology/P15-1137>
23. Wiseman, S., Rush, A.M., Shieber, S.M.: Learning global features for coreference resolution. In: Proceedings of the 2016 Conference of the NAACL: Human Language Technologies, pp. 994–1004. Association for Computational Linguistics, San Diego, California, June 2016
24. Wu, W., Wang, F., Yuan, A., Wu, F., Li, J.: CorefQA: coreference resolution as query-based span prediction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6953–6963. Association for Computational Linguistics, Online, July 2020. <https://doi.org/10.18653/v1/2020.acl-main.622>, <https://www.aclweb.org/anthology/2020.acl-main.622>
25. Zhang, R., Nogueira dos Santos, C., Yasunaga, M., Xiang, B., Radev, D.: Neural coreference resolution with deep biaffine attention by joint mention detection and mention clustering. In: Proceedings of the 56th Annual Meeting of the ACL (Volume 2: Short Papers), pp. 102–107. Association for Computational Linguistics, Melbourne, Australia, July 2018



Chinese Macro Discourse Parsing on Dependency Graph Convolutional Network

Yixin Fan, Feng Jiang, Xiaomin Chu, Peifeng Li, and Qiaoming Zhu^(✉)

School of Computer Science and Technology, Soochow University, Suzhou, China
`{yxfansupery,fjiang}@stu.suda.edu.cn, {xmchu,pfli,qmzhu}@suda.edu.cn`

Abstract. The macro-level discourse parsing, as a fundamental task of macro discourse analysis, mainly focuses on converting a document into a hierarchical discourse tree at paragraph level. Most existing methods follow micro-level studies and suffer from the issues of semantic representation and the semantic interaction of the larger macro discourse units. Therefore, we propose a macro-level discourse parser based on the dependency graph convolutional network to enhance the semantic representation of the large discourse unit and the semantic interaction between those large discourse units. Experimental results on both the Chinese MCDTB and English RST-DT show that our model outperforms several state-of-the-art baselines.

Keywords: Macro discourse parsing · Graph convolutional network · Dependency graph

1 Introduction

Discourse parsing aims to study the internal structure of a document and understand the nuclearity and relation between discourse units (such as phrases, sentences, and paragraphs). It has been widely used in various natural language processing applications, such as summarization [16] and event extraction [4].

As one of the famous theories in discourse parsing, Rhetorical Structure Theory (RST) [20] represents a document as a hierarchical discourse tree. Commonly, the research on discourse parsing can be divided into two levels: micro and macro levels. The micro-level mainly studies the structure within or between sentences, while the macro-level mainly studies the structure between paragraphs.

We take a macro discourse tree as an example, as shown in Fig. 1. The leaf nodes are paragraphs, which are called Paragraph-level Discourse Units (PDUs). The directed edge indicates that the node is a nucleus and the undirected edge indicates that the node is a satellite. The internal node has a relation label that is the discourse relation between discourse units. In this paper, we mainly focus on the construction of macro Chinese discourse trees.

With the success of the micro-level [17, 18], the macro discourse parsing [5, 10, 11] always follows the micro level. However, it faces more challenges because

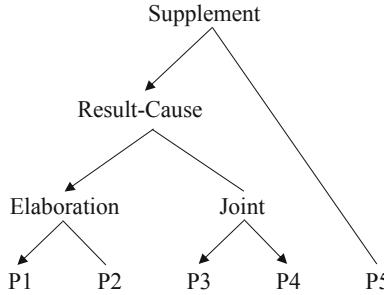


Fig. 1. An example of the macro discourse tree. This document contains five paragraphs (i.e., five PDUs from P1 to P5)

of the larger size of discourse units, i.e., PDUs. Jiang et al. [11] mentioned that the average token lengths of the leaf nodes in Chinese MCDTB [13] and English RST-DT [1] are 22 and 8 at micro level, respectively, while these figures are 100 and 52 at macro level. The larger discourse units bring two challenges to macro discourse parsing. The first is how to model the semantic representation of a large macro discourse unit, and the second is how to model the semantic interaction between large macro discourse units.

Some previous works [10, 11] modeled the semantic representation and semantic interaction of discourse units via the pre-trained model BERT. Due to the limited ability of BERT to process long text, it is difficult to capture the core semantics of the macro discourse unit that often contains a lot of redundant information. Besides, although BERT uses a self-attention mechanism to model the semantic interaction of discourse units, the self-attention mechanism can only capture the dependencies between words but not word groups. However, it is also important for the dependencies between word groups to model the semantic interaction between discourse units.

Recently, some studies [19, 26] have succeeded in achieving enhanced semantic representation or semantic interaction by incorporating syntactic information. Syntax provides structural information representing human understanding of the text and is helpful to obtain the core information from discourse units.

To represent the core information of macro discourse units, we propose a root-oriented approach to obtain the internal topic graph from the original dependency tree. By integrating the internal topic graphs of discourse units, we can enhance the semantic representation of large discourse units and alleviate the first challenge. To address the second challenge, we propose a topic interaction mechanism to obtain the interactive topic graph between discourse units. It models the dependencies between word groups in different discourse units based on the topic consistency and topic correlation. Finally, we combine the internal topic graph and interactive topic graph and feed them into the graph convolutional network to obtain the enhanced semantics representation of large discourse units and the semantic interaction between large discourse units.

2 Related Work

In English, RST-DT [1] is one of the popular discourse corpora. The research of discourse parsing on this corpus has two levels: micro-level and macro-level.

Most of the work focuses on the micro level. Hernault et al. [8] proposed a HILDA parser, which uses the Support Vector Machine (SVM) and constructs the discourse structure tree with the bottom-up algorithm. Joty et al. [14] and Feng and Hirst [6] proposed models using Conditional Random Field (CRF) for discourse parsing. With the development of neural networks, Ji and Eisenstein [9] adopted neural network for discourse parsing and achieve comparable performance. Recently, some works [17, 18] introduced the pointer network to the micro discourse parser that got close to human performance.

However, only a few works on RST-DT focus on the macro level. Sporleder and Lascarides [23] built macro discourse trees on the bottom-up algorithm after pruning and revising the original discourse trees on RST-DT.

In Chinese, the MCDTB [13] is the only available macro Chinese discourse corpus. Zhou et al. [27] mined semantic interaction among discourse units through the multi-view word-pair similarity model and constructed discourse structure tree through the shift-reduce algorithm. Fan et al. [5] adopted a pointer network, which performs well at the micro level, to construct the discourse structure tree with a top-down algorithm. Jiang et al. [10] proposed the method of global backward reading and the local reverse reading to mine the semantics between discourse units and constructed the discourse structure tree by using the shift-reduce algorithm. Jiang et al. [11] proposed a method based on topic segmentation and achieved the SOTA performance. It first splits a document into several sections using the topic boundaries that the topic segmentation mechanism detects. Then it builds a discourse sub-tree using the shift-reduce algorithm in each section and sequentially forms a whole tree for a document.

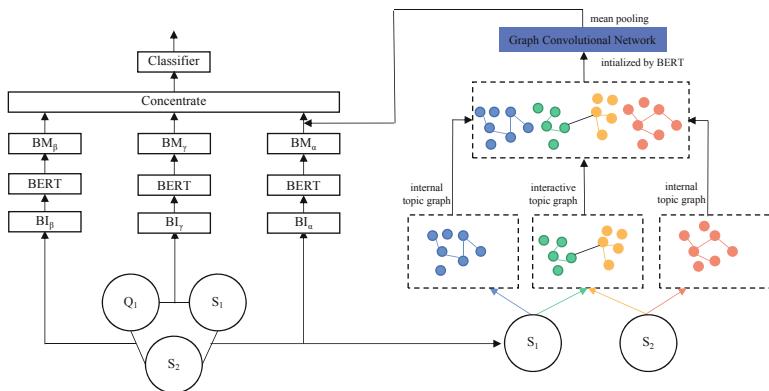


Fig. 2. The architecture of our proposed model DGCNParser-TS.

3 Basic Model: MDParse-TS

Since our work is based on MDParse-TS [11], we first introduce MDParse-TS in this section. The basic model MDParse-TS includes two components, the topic segmentation and the macro discourse Parser. In the stage of topic segmentation, MDParse-TS first segment a document into several sections according to their different topics. In the stage of discourse parsing, MDParse-TS first constructs the naked structure tree inside and between sections and then recognizes the nuclearity and relationship of the node in the tree individually.

MDParse-TS used the shift-reduce algorithm to build a discourse structure tree. To decide whether to shift or reduce, MDParse-TS proposed an action classifier TM-BERT by considering three adjacent discourse units from the top two PDUs of the stack S_1 and S_2 and the first PDUs of the queue Q_1 . As shown on the left of Fig. 2, TM-BERT employs pairs of $\alpha(S_1, S_2)$ and $\beta(S_2, Q_1)$ for semantic matching but also matches the across-DUs pair $\gamma(S_1, Q_1)$, as shown in the following equation.

$$BM_k = BERT(BI_k), k \in \{\alpha, \beta, \gamma\} \quad (1)$$

where BI is the input of BERT, BM is the embedding of [CLS] position in the output of BERT. Then, the triple semantic match output $(BM_\alpha, BM_\beta, BM_\gamma)$ is concentrated and fed into a Softmax layer to get the probabilities of action (Shift or Reduce).

4 Chinese Macro Discourse Parsing on Dependency Graph Convolutional Network

Our model DGCNParser-TS is based on MDParse-TS [11], and we incorporate the internal topic graph and the interactive topic graph into a graph convolutional network to TM-BERT to enhance the semantic representation of large discourse units and the semantic interaction between large discourse units in macro discourse parsing. The overall framework is shown in Fig. 2. Our approach consists of three parts, as shown on the right of Fig. 2: 1) Internal topic graph Construction. 2) Interactive topic graph Construction. 3) Dependency Graph Convolutional Network.

4.1 Internal Topic Graph Construction

We propose a root-oriented method based on a dependency tree to obtain the core contents as the internal topic graph. Specifically, let $P = \{s_1, s_2, \dots, s_n\}$ denote a discourse unit P where s_i is the i -th sentences in this unit and n is the sentence number. We first use LTP tools [2] to obtain the dependency tree dt_i of each sentence s_i in the discourse unit P . For each dependency tree dt_i , we do the following operations: **a)** we search the root node (e.g., “met” in Fig. 3) and nodes connected with it (e.g., there are six nodes connected with “met” in

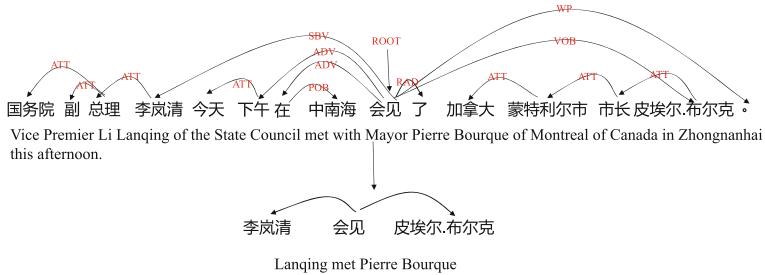


Fig. 3. An example to obtain the internal topic graph.

Fig. 3); **b)** we obtain the nodes that have a subject-predicate (e.g., “Li Lanqing” in Fig. 3) and verb-object relationship (e.g., “Pierre Bourque” in Fig. 3) with root; **c)** the root node is used as the predicate to form the core contents as the internal topic graph with the subject-verb-object form, as shown in Fig. 3.

In Fig. 3, the sentence above is the original sentence and the sentence below is its core content following the above method. Besides, if it doesn’t form a subject-predicate-object form, we keep the whole dependency tree as the internal topic graph. A discourse unit with multiple sentences that form its internal topic graph by connecting the root nodes of internal topic graphs of adjacent sentences.

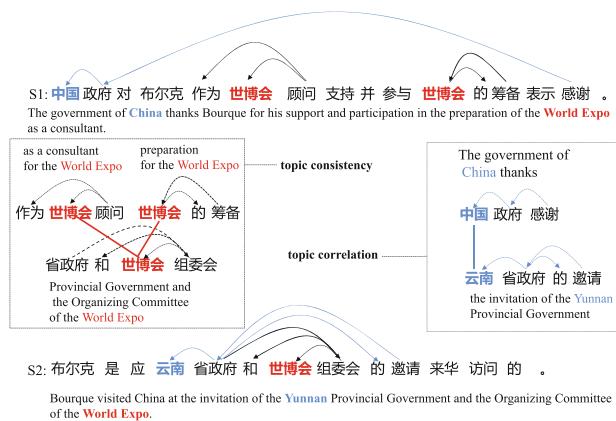


Fig. 4. An example to obtain the interact topic graph where S1 and S2 are the sentences of two discourse units, respectively.

4.2 Interactive Topic Graph Construction

The construction of the interactive topic graph between discourse units consists of two parts, one is to model the topic consistency, and the other is to model the topic correlation.

Topic Consistency. By constructing interactive edges of the same topic in different discourse units, the interaction of related topics centered on the same topic can be enhanced. We keep the two-hop of the same topics (we used nouns as topics) for different discourse units according to dependency trees and construct interactive edges for the same topics. As shown in Fig. 4, the topic “World Expo” appears in different sentences simultaneously, and we construct the interactive edges of the topic “World Expo”.

Topic Correlation. We construct interactive edges between different topics in different discourse units by using external knowledge. External knowledge has achieved effects in recognition of the implicit discourse relation [7]. Inspired by this, we obtain the set of external knowledge from CN-DBpedia [25], expressed in KG. $Topic^1 = \{w_1^1, w_2^1, \dots, w_m^1\}$ and $Topic^2 = \{w_1^2, w_2^2, \dots, w_m^2\}$ are topic sets from different discourse units. for words pair (w_i^1, w_j^2) , where $w_i^1 \in Topic^1$, $w_j^2 \in Topic^2$. If $(w_i^1, w_j^2) \in KG$, then we construct interactive edges for different topics in the same way as the topic consistency. As shown in Fig. 4, we can obtain the knowledge of (China, province, Yunnan) from KG, and “China” and “Yunnan” appear in different sentences, so we construct the interactive edge between topic “China” and “Yunnan”.

4.3 Dependency Graph Convolutional Network

Nodes Initialization. Given a discourse unit pair P^1 and P^2 , where $P^1 = \{w_1^1, w_2^1, \dots, w_m^1\}$, $P^2 = \{w_1^2, w_2^2, \dots, w_n^2\}$. We feed P^1 and P^2 into BERT in the form of $[CLS]P^1[SEP]P^2[SEP]$ to obtain the semantic representation $P_H^1 = \{h_1^1, h_2^1, \dots, h_m^1\}$ and $P_H^2 = \{h_1^2, h_2^2, \dots, h_n^2\}$. The initialized node representation of the dependency graph convolutional network is as follows.

$$H^0 = \{h_1^1, h_2^1, \dots, h_m^1, h_1^2, h_2^2, \dots, h_n^2\} \quad (2)$$

Adjacency Matrix Construction. We can directly transform the internal topic graph and the interactive topic graph into an adjacency matrix. Given an adjacency matrix $A \in \mathbb{R}^{l \times l}$ where $A_{ij} = 1$ if the word i is connected with the word j and $l = m + n$. Considering that BERT is trained by characters in Chinese, there is an inconsistency with Chinese words. A Chinese word usually consists of more than one character. Inspired by Meng et al. [21], we connect all characters once when there are edges between words. Following the idea of self-looping [15], we added an identity matrix I and then $A = A + I$.

Graph Convolutional Operation. In order to make information flow better, we add the residual connection to our model. Hence, the output of $(i+1)$ -th layer is calculated as follows.

$$H^{i+1} = \sigma(Norm(A)H^iW^i) + H^i \quad (3)$$

where $Norm$ is the normalized function, σ is the activation function RELU. The output of the last layer is $H^l \in \mathbb{R}^{(m+n) \times d_h}$, where m, n are the sequence lengths of P^1 and P^2 , respectively. Then, we use the mean pooling to obtain H' , the final outputs of GCN, where $H' \in \mathbb{R}^{d_h}$. Finally, H^c , the semantic representation of the [CLS] token from the pre-trained model, is added to obtain the final enhanced semantic representation between P^1 and P^2 as follows.

$$H^f = H' + H^c \quad (4)$$

4.4 Classifier

Following the idea of TM-BERT [11], we consider the information of three adjacent discourse units. The enhanced semantic representations of discourse unit pairs $(S1, S2)$, $(S2, Q1)$ and $(S1, Q1)$ are H_1^f , H_2^f and H_3^f , respectively.

We combine the three enhanced semantic representations and feed them to a Softmax layer to obtain the probabilities of action (Shift or Reduce), as follows.

$$y = \text{Softmax}([H_1^f, H_2^f, H_3^f]) \quad (5)$$

When the action is predicted as Reduce, we feed H_1^f to the Relation classifier and the Nuclearity classifier respectively to get the relation and the nuclearity between the discourse units $(S1, S2)$.

5 Experimentation

In this section, we first describe the dataset and experimental settings and then evaluate our model and several baselines on Chinese MCDTB dataset. Finally, we report the experimental results.

5.1 Dataset and Experimental Settings

Our experiments are primarily evaluated on the Macro Chinese Discourse Treebank (MCDTB) [13]. Following previous work [11], there is 80% data (576 documents) for training and 20% data (144 documents) for testing and we report the micro-averaged F1 score for Span, Nuclearity, Relation.

We set the number of GCN layers as 2. The dimension of the input layer, hidden layer and output layer of GCN is set as 768 ($d_h=768$). We set the training epoch as 5, the learning rate as 1e-5 and the batch as 4. We use a Geforce Nvidia 1080Ti and adopt the gradient accumulation manner for training.

5.2 Baselines

To evaluate the performance of our **DGCNPParser-TS**, we compare it with the following two types of benchmarks: w/o pre-trained model and w/ pre-trained model.

Table 1. The performance comparison (in micro-F1) between our DGCNParser-TS and the baselines. Superscript * indicates that we reproduce the model. We did three-time experiments and reported the average performance.

Model	Span	Nuclearity	Relation
w/o pre-trained model			
LD-CM	54.71	48.38	26.28
MVM	56.11	47.76	27.67
PNGL	58.42	–	–
w/ pre-trained model			
TM-BERT*	61.82	51.62	32.30
PDParse-GBLRR*	65.38	55.64	35.39
MDParser-TS*	66.92	56.26	37.24
DGCNParser-TS(ours)	69.24	57.96	38.79

w/o Pre-trained Model

LD-CM [12]: By using manual features, a conditional random field is used to predict the structure. **MVM** [27]: It models the semantics of discourse units from three perspectives and proposes a word-pair similarity mechanism to measure the semantics of discourse units. **PNGL** [5]: It is based on the pointer network, which alleviates the insufficiency of the encoding layer to capture long-distance dependency by introducing a local interaction module.

w/ Pre-trained Model

TM-BERT [10]: The pre-trained model BERT encodes three adjacent discourse units to predict the action (Shift or Reduce). **PDParse-GBLRR** [10]: It proposes the method of global backward reading and the local reverse reading to mine the semantics between discourse units. **MDParser-TS** [11]: It proposes a method based on topic segmentation and achieves the SOTA performance on MCDTB.

5.3 Experimental Results

Table 1 shows the performance comparison between our model DGCNParser-TS and all the baselines. From Table 1 we can find out that our DGCNParser-TS outperforms all baselines on all three tasks: Span, Nuclearity and Relation. These results indicate the effectiveness of combining the internal topic graph and interactive topic graph to obtain the enhanced semantics representation of large discourse unit and the semantic interaction between large discourse units.

TM-BERT only uses the pre-trained model to obtain the semantic representation of discourse units, which has surpassed all the methods that do not use the pre-trained model. Compared with TM-BERT, PDParse-GBLRR improves the performance of constructing discourse structure trees by further mining the

semantics of discourse units. MDParse-TS adopts the idea of topic segmentation to achieve the benchmark of SOTA.

Compared with the best baseline MDParse-TS, our model DGCNParsor-TS enhances the semantic representation and semantic interaction of discourse units via dependency graph convolutional network and improves the micro-F1 score by 2.32, 1.70, and 1.55 on the discourse tree construction (Span), nuclearity recognition (Nuclearity), and relation classification (Relation), respectively.

6 Analysis

In this section, we first give the analysis on the proposed internal topic graph and interactive topic graph. Then we report the experiment on another dataset RST-DT.

6.1 Analysis on Internal Topic Graph

To prove the effectiveness of our root-oriented internal topic graph construction mechanism, we compare it with the other three methods to obtain the internal topic graph as follows.

- **Original topic graph.** We obtain the internal topic graph of a discourse unit by preserving all the nodes and edges of the original dependency tree.
- **Verb-oriented topic graph.** We preserve all the nodes in the dependency tree whose part of speech is a verb and all the nodes connected to them.
- **Noun-oriented topic graph.** We preserve all the nodes in the dependency tree whose part of speech is a noun and all the nodes connected to them.

The experimental results are shown in Table 2, where MDParse-TS is our baseline. There is a slight performance improvement on the task Span when integrated the internal topic graph from the original dependency tree (+Original). Compared with the original topic graph (+Original), the verb-oriented topic graph (+Verb-oriented) and the noun-oriented topic graph (+Noun-oriented) improves more obviously on the task Span. The above two kinds of topic graphs keep some relatively important dependency structures between words from different perspectives, enhancing the semantic representation of discourse units. Our root-oriented internal topic graph (+Root-oriented) keeps the most important dependency structures between words of discourse units in the form of subject-verb-object and obtains a stronger semantic representation of the discourse units.

6.2 Analysis on Interactive Topic Graph

In this paper, we construct the interactive topic graph from two aspects of the topic consistency and topic correlation. To explore the effectiveness of our method on the topic consistency and the topic correlation, we conducted ablation experiments shown in Table 3. It shows that removing topic consistency or

Table 2. The comparison (Micro-F1) of different approaches to obtain the internal topic graph on the task Span.

Approach	Span
MDParser-TS	66.92
+Original	67.23
+Verb-oriented	67.54
+Noun-oriented	67.69
+Root-oriented (ours)	68.16

Table 3. Ablation experiments for interactive topic graph on the task Span.

Approach	Span
DGCNParser-TS	69.24
w/o topic consistency	68.62
w/o topic correlation	68.46
w/o topic consistency and topic correlation	68.16

topic correlation will result in a performance decrease. Specifically, if the topic consistency or the topic consistency is removed from our model DGCNParser-TS, its Micro-F1 score will drop by 0.78 or 0.62, respectively. If both of them are removed, the performance will drop by 1.08, indicating that their combination complements each other in enhancing the semantic interaction of discourse units.

6.3 Experimentation on English RST-DT

To verify the generalization of the proposed model, we also evaluate our model on the English RST-DT. We use Stanford Parser [3] to obtain the dependency tree and use WordNet [22] to obtain external knowledge. The data division and pro-processing are the same as Jiang et al. [11], and the results at the macro level are shown in Table 4.

Table 4. The performance comparison on the RST-DT at the macro level. Superscript * indicates we reproduce the model.

Approach	Span	Nuclearity	Relation
SL04	34.29	–	–
WL17	37.40	28.83	18.70
MDParser-TS*	41.56	33.77	23.12
DGCNParser-TS (ours)	43.37	34.55	23.63

There are three baselines in Table 4 as follows: **SL04** [23] builds the macro discourse trees with the bottom-up algorithm after pruning and revising the original discourse trees on RST-DT. **WL17** [24] designs a pipeline two-stage parsing method and uses the shift-reduce algorithm to construct discourse structure trees. **MDParser-TS** [11] is one of the SOTA models at the macro level on RST-DT.

Compared with the best baseline MDParser-TS, our DGCNParser-TS improves the micro-F1 scores by 1.81, 0.78, 0.51 on Span, Nuclearity and Relation separately, which also proves the effectiveness of our model on English macro discourse parsing.

7 Conclusion

In this paper, we propose a macro-level discourse parser based on the dependency graph convolutional network to enhance the semantic representation of a discourse unit and the semantic interaction between discourse units. Our model achieves the SOTA performance in macro discourse parsing both on the MCDTB and RST-DT datasets. In the future, we will explore a more accurate action classifier based on the graph convolutional network to improve macro discourse parsing performance further.

Acknowledgements. The authors would like to thank the three anonymous reviewers for their comments on this paper. This research was supported by the National Natural Science Foundation of China (No. 61773276, 61772354 and 61836007.), and the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).

References

1. Carlson, L., Marcu, D., Okurowski, M.E.: Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory, pp. 85–112 (2003)
2. Che, W., Li, Z., Liu, T.: LTP: a chinese language technology platform. In: COLING, pp. 13–16 (2010)
3. Chen, D., Manning, C.: A fast and accurate dependency parser using neural networks. In: EMNLP, pp. 740–750 (2014)
4. Choubey, P.K., Lee, A., Huang, R., Wang, L.: Discourse as a function of event: profiling discourse structure in news articles around the main event. In: ACL, pp. 5374–5386 (2020)
5. Fan, Y., Jiang, F., Chu, X., Li, P., Zhu, Q.: Combining global and local information to recognize Chinese macro discourse structure. In: CCL, pp. 183–194 (2020)
6. Feng, V.W., Hirst, G.: A linear-time bottom-up discourse parser with constraints and post-editing. In: ACL, pp. 511–521 (2014)
7. Guo, F., He, R., Dang, J., Wang, J.: Working memory-driven neural networks with a novel knowledge enhancement paradigm for implicit discourse relation recognition. In: AAAI, pp. 7822–7829 (2020)
8. Hernault, H., Prendinger, H., du Verle, D.A., Ishizuka, M.: Hilda: a discourse parser using support vector machine classification. Dialogue Discourse **1**(3), 1–33 (2010)
9. Ji, Y., Eisenstein, J.: Representation learning for text-level discourse parsing. In: ACL, pp. 13–24 (2014)
10. Jiang, F., Chu, X., Li, P., Kong, F., Zhu, Q.: Chinese paragraph-level discourse parsing with global backward and local reverse reading. In: COLING, pp. 5749–5759 (2020)
11. Jiang, F., Fan, Y., Chu, X., Li, P., Zhu, Q., Kong, F.: Hierarchical macro discourse parsing based on topic segmentation. In: AAAI (2021)
12. Jiang, F., Li, P., Chu, X., Zhu, Q., Zhou, G.: Recognizing macro chinese discourse structure on label degeneracy combination model. In: NLPCC, pp. 92–104 (2018)
13. Jiang, F., Xu, S., Chu, X., Li, P., Zhu, Q., Zhou, G.: Mcdtb: a macro-level chinese discourse treebank. In: Coling, pp. 3493–3504 (2018)
14. Joty, S., Carenini, G., Ng, R., Mehdad, Y.: Combining intra- and multi-sentential rhetorical parsing for document-level discourse analysis. In: ACL, pp. 486–496 (2013)

15. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint [arXiv:1609.02907](https://arxiv.org/abs/1609.02907) (2016)
16. Li, Z., Wu, W., Li, S.: Composing elementary discourse units in abstractive summarization. In: ACL, pp. 6191–6196, July 2020
17. Lin, X., Joty, S., Jwalapuram, P., Bari, M.S.: A unified linear-time framework for sentence-level discourse parsing. In: ACL, pp. 4190–4200 (2019)
18. Liu, L., Lin, X., Joty, S., Han, S., Bing, L.: Hierarchical pointer net parsing. In: EMNLP-IJCNLP, pp. 1007–1017 (2019)
19. Ma, N., Mazumder, S., Wang, H., Liu, B.: Entity-aware dependency-based deep graph attention network for comparative preference classification. In: ACL, pp. 5782–5788 (2020)
20. Mann, W.C., Thompson, S.A.: Rhetorical Structure Theory: A Theory of Text Organization. University of Southern California, Information Sciences Institute Los Angeles (1987)
21. Meng, F., Feng, J., Yin, D., Chen, S., Hu, M.: Sentiment analysis with weighted graph convolutional networks. In: EMNLP: Findings, pp. 586–595 (2020)
22. Miller, G.A.: Wordnet: a lexical database for english. Commun. ACM **38**(11), 39–41 (1995)
23. Sporleder, C., Lascarides, A.: Combining hierarchical clustering and machine learning to predict high-level discourse structure. In: COLING, pp. 43–49 (2004)
24. Wang, Y., Li, S., Wang, H.: A two-stage parsing method for text-level discourse analysis. In: ACL, pp. 184–188 (2017)
25. Xu, B., et al.: Cn-dbpedia: a never-ending chinese knowledge extraction system. In: Benferhat, S., Tabia, K., Ali, M. (eds.) Advances in Artificial Intelligence: From Theory to Practice, pp. 428–438 (2017)
26. Zhang, C., Li, Q., Song, D.: Aspect-based sentiment classification with aspect-specific graph convolutional networks. In: EMNLP-IJCNLP, pp. 4568–4578 (2019)
27. Zhou, Y., Chu, X., Li, P., Zhu, Q.: Constructing chinese macro discourse tree via multiple views and word pair similarity. In: Tang, J., Kan, M.Y., Zhao, D., Li, S., Zan, H. (eds.) NLPCC, pp. 773–786 (2019)



Predicting Categorial Sememe for English-Chinese Word Pairs via Representations in Explainable Sememe Space

Baoju Liu^{1,2}, Lei Hou^{1,2(✉)}, Xin Lv^{1,2}, Juanzi Li^{1,2}, and Jinghui Xiao³

¹ Department of Computer Science and Technology, BNRist, Beijing, China
[{liu-bj17,lv-x18}@mails.tsinghua.edu.cn,](mailto:{liu-bj17,lv-x18}@mails.tsinghua.edu.cn)
{houlei,lijuanzi}@tsinghua.edu.cn

² KIRC, Institute for Artificial Intelligence, Tsinghua University,
Beijing 100084, China

³ Noah's Ark Lab, Huawei Inc., Shenzhen, China
xiaojinghui4@huawei.com

Abstract. Sememe is the minimum unambiguous semantic unit in human language. Sememe knowledge bases(SKB) have been proven to be effective in many NLP tasks. Categorial sememe, indicating the basic category of word sense to bridge the lexicon and semantics, is indispensable in SKB. However, manual categorial sememe annotation is costly. This paper proposes a new task to automatically build SKB: English-Chinese Word Pair Categorial Sememe Prediction. The bilingual information is utilized to resolve the ambiguity challenge. Our method proposes the sememe space, in which sememes, words, and word senses are represented as vectors with interpretable semantics, to bridge the semantic gap between sememes and words. Extensive experiments and analyses validate the effectiveness of the proposed method. Using this method, we predict categorial sememes for 113,014 new word senses, and the prediction MAP is 85.8%. Further we conduct expert annotations based on prediction results and increase HowNet nearly by 50%. We will publish all the data and code.

1 Introduction

Sememes are defined as the minimum unambiguous indivisible semantic units of human languages in the field of linguistics [1]. Sememe knowledge bases (SKBs), in which the meanings of words or word senses are described by a pre-defined closed set of sememes, benefit various NLP tasks, e.g., word sense disambiguation [3], word embedding enhancement [6], semantic composition [5], relation extraction [15], event detection [14], sentiment analysis [4] and textual adversarial attack [16]. Categorial sememe, indicating the basic category of word sense to bridge the lexicon and semantics, is indispensable in SKB. For example, three senses of word *kid* in Fig. 1 are distinguished by three categorial sememes, i.e., *human*, *tease*, and *livestock*.

HowNet is a widely used bilingual SKB that is manually developed by linguists. Specifically, HowNet consists of two parts, the sememe taxonomy and the word (sense) dictionary [13]. The sememe taxonomy is composed of 2,214 sememes and 116 semantic relations, which are organized in the form of taxonomy, as illustrated

in Fig. 1(b)¹. The word (sense) dictionary contains over 210,000 word senses, each of which is described by one categorial sememe and its extended elaborations. The categorial sememe indicates the basic category of word sense in sememe taxonomy, and the extended elaborations further illustrate the meaning of word sense through the semantic relations with the corresponding sememes.

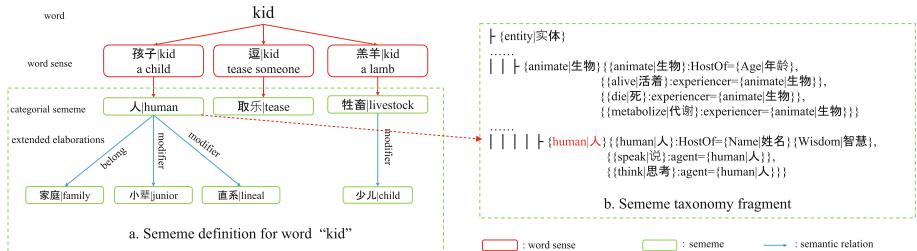


Fig. 1. Sememe definition for word “kid” and Sememe taxonomy fragment

In Fig. 1(a), the sememe definition for word sense “孩子|kid” corresponds to natural language description: “孩子|kid” is a junior human who belongs to a family. We can augment word sense “孩子|kid” with sememe-level common sense from its categorial sememe “人|human”, e.g., (思考|think, agent, 孩子|kid), (孩子|kid, HostOf, 智慧|wisdom). Therefore, the categorial sememe is critical to word sense definition and can benefit many NLP tasks with common sense reasoning.

Nowadays new words and phrases are emerging every day and the semantic meanings of existing concepts keep changing. Human annotating categorial sememes for new words is extremely time-consuming and labor-intensive². Although there emerged a series of recent works that tried to automatically predict the sememe set for new words [7–9], they neither recognized the categorial sememe among the sememe set nor distinguished multiple senses of polysemous words. In this paper, we expect to develop a technique for automatic categorial sememe prediction. Such a non-trivial task poses the following two problems: 1) Categorial sememe indicates the basic category of word sense. Polysemy is pervasive in human language, e.g., almost 2/3 of total senses in HowNet come from polysemous words. Therefore, it is difficult to identify all word senses of new words. 2) Unlike sememe set prediction that only relates words to sememes, categorial sememe prediction needs to make a bridge over new words, word senses and sememes.

Inspired by [10, 11], we introduce cross-lingual information to dissect polysemous words. HowNet also uses Chinese-English parallel word pairs to represent word senses, and preliminary statistic shows that 98.09% of all 221540 word pairs are unambiguous. Therefore, we assume that a Chinese-English bilingual word pair has an unambiguous word sense. So we formalize a new task, i.e., categorial sememe prediction for English-Chinese word pairs (**ECCSP**).

¹ The sememes are described in a Knowledge Database Mark-up Language, and can be parsed and used according to the language grammar.

² HowNet construction took linguists over 30 years.

To better model word, word sense and sememe, we propose an explainable semantic space named sememe space, where each dimension corresponds to a sememe. Hence the sememes of a word sense can be interpreted once its representation in sememe space is obtained. For a word sense in HowNet, we achieve its representations by composing its categorial sememe and extended elaborations in sememe space, and further calculate word representation in sememe space by aggregating all its senses. For a new word pair, we approximate its sememe representation using collaborative filtering and predict the categorial sememe.

The main contributions in this paper can be summarized as follows:

- We introduce a new task of categorial sememe prediction for English-Chinese word pairs, which leverages cross-lingual information to alleviate polysemy in single language.
- We propose the sememe space, where each dimension has explainable semantic meaning. Then we utilize the word sense definition in HowNet to achieve word and word sense representations, create a mapping from pre-trained word vector space to sememe space, and predict the categorial sememe for new word pairs using collaborative filtering.
- Extensive experiments and analysis validate the effectiveness of the proposed method. Using this method, we predict categorial sememes for 113,014 new word senses from Oxford English-Chinese dictionary and the prediction MAP is 85.8%. Further we conduct expert annotations based on prediction results.

2 Task Formalization

HowNet is an online common-sense knowledge base unveiling the inter-conceptual and inter-attribute of concepts connoted in lexicons of the Chinese and their English equivalent [12]. In HowNet, word senses are encoded by 2,214 primitive concepts called sememes and 116 semantic relations. In this section, we will formally define sememe, semantic relation, and word sense in HowNet, and then introduce the **ECCSP** task.

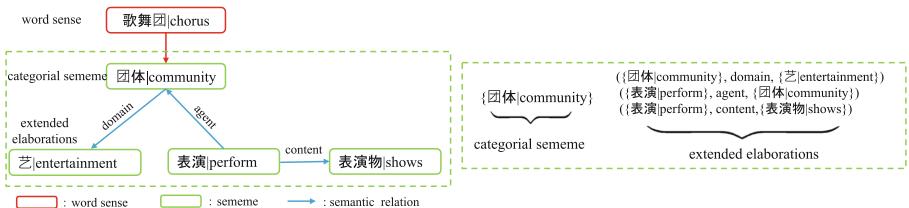


Fig. 2. Sememe definition for “歌舞团|chorus”

Definition 1 (Sememe). A *sememe* is an unambiguous indivisible semantic unit for human languages. Let s denote a sememe, e.g., “人|human”, “思考|think”, we represent the sememes in HowNet as a set $S = \{s_1, s_2, \dots, s_n\}$ where $n = 2,214$.

Definition 2 (Semantic Relation). Semantic relations are used to describe the semantics of the word pairs and sememes e.g., agent. We represent the semantic relation set as $R = \{r_1, r_2, \dots, r_m\}$ where $m = 116$.

Definition 3 (Word Sense with Sememe Definition). A word sense is represented as a bilingual parallel word pair $p = (w^z|w^e)$ with the w^z and w^e denote Chinese word and English word respectively. Each word sense p in HowNet is defined by two parts, i.e., its categorial sememe $\tilde{s}_p \in S$, which indicates the basic class it belongs to, and extended elaborations attaching to the categorial sememe. The elaborations can be parsed into a triple set $T = \{(h, r, t) | h, t \in S; r \in R\}$, $|T| = 7282$. Figure 2 illustrates the triple set of “歌舞团|chorus”.

Definition 4 (ECCSP). We use $H^p = \{p_i\}_{i=1}^{N_1}$ and $D^p = \{p_j\}_{j=1}^{N_2}$ to represent the HowNet word pair set and target word pair set (without sememe information). $H^p \cap D^p = \emptyset$. Given sememe set S , H^p , and D^p , bilingual categorial sememe prediction is to predict the categorial sememe \tilde{s}_p for each word pair $p \in D^p$.

From the above definitions, how to eliminate the semantic gap between sememes and word senses remains the key and challenging issue to solve the novel task. In this paper, we put efforts on the following two problems to tackle this issue: 1) properly model words, word senses and sememes using unstructured contextual texts and structured HowNet definitions; 2) make prediction via the interactions between word senses and sememes.

3 Methodology

In this section, we introduce the proposed novel model for ECCSP. Our model creates an explainable semantic space (named as sememe space O^s). Once the vector representations of words or word senses in O^s are obtained, their sememe semantics can be easily interpreted. Words and word senses in HowNet can be embedded into the O^s using their structured definitions. Unlike HowNet, the words and word senses in D^p do not have structured definitions. So their representations in O^s cannot be directly achieved. Fortunately, words in HowNet and D^p share the same representations in contextual word embeddings space (named as word vector space O), which can be utilized as a bridge between D^p and O^s . Although we cannot directly obtain the representations in O^s for words and word senses in D^p , we can use similar words to achieve an approximation. Figure 3 presents the overall framework. We use $H^z = \{w_i^z\}_{i=1}^{N_3}$, $H^e = \{w_j^e\}_{j=1}^{N_4}$ to represent the Chinese word set and English word set in HowNet. Correspondingly, $D^z = \{w_i^z\}_{i=1}^{N_5}$ and $D^e = \{w_j^e\}_{j=1}^{N_6}$ represent the Chinese word set and English word set in D^p .

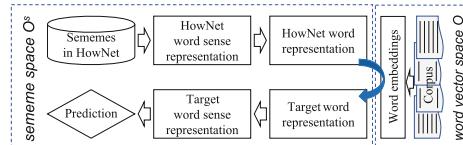


Fig. 3. Model framework.

Next, we first introduce the construction of word vector space O and sememe space O^s , then present the word and word sense representations in HowNet and D^p , and finally show the details of model training and prediction.

3.1 Word Vector Space O and Sememe Space O^s

Word embedding, which maps words into continuous low-dimensional vector space, has been widely used in various NLP tasks. Here we denote such space as word vector space O . Each vector in O reflects the co-occurrence probability of the context, but cannot be interpreted, i.e., each dimension in O represents a latent and inexplicable semantic meaning.

To explicitly represent words and word senses, we construct another semantic space based on sememes defined in HowNet, and thus name it as sememe space O^s . Sememes are finite and explicit. People can use a limited sememe set to describe all word senses. Each sememe is the smallest indivisible unambiguous semantic unit and cannot be described by other sememes. Therefore, we let each dimension in O^s denote a sememe, and the original 2,214 sememes constitute the base vectors of O^s (i.e., $\forall s_i \in S, V_{s_i}$ is a one-hot vector). Words and word senses are represented as vectors in O^s , which have sememe semantics.

3.2 HowNet in Sememe Space O^s

In this section, we present how to represent words and word senses in HowNet using the sememe space in previous section.

HowNet Word Sense Representation. As defined in Sect. 2, each word sense p_i in HowNet is defined by categorial sememe $\tilde{s}_{p_i} \in S$ and extended elaborations $T_{p_i}, T_{p_i} \subset T$. The extended elaborations attaching to the categorial sememe is to define the word sense. So, $\forall (\tilde{s}_{p_i}, r_j, t_j) \in T_{p_i}, \forall (h_k, r_k, \tilde{s}_{p_i}) \in T_{p_i}, (r_j, t_j)$ and $(h_k, r_k,)$ can be seen as descriptions for p_i . The semantic that each description expresses is modeled as a vector in O^s , i.e., $V_{(r_j, t_j)}$ and $V_{(h_k, r_k,)}$ are independent variables that need to be learned. So we achieve:

$$V_{p_i} = V_{\tilde{s}_{p_i}} + \sum_{(h_j, r_j, t_j) \in T_{p_i}, h_j = \tilde{s}_{p_i}} V_{(r_j, t_j)} + \sum_{(h_k, r_k, t_k) \in T_{p_i}, t_k = \tilde{s}_{p_i}} V_{(h_k, r_k,)}, \forall p_i \in H^p \quad (1)$$

e.g., the word sense $p_i = (\text{歌舞团}|\text{chorus})$ in Fig. 2 can be represented as:

$$V_{\text{歌舞团}|\text{chorus}} = V_{\text{团体}|\text{community}} + V_{(\text{表演}|\text{perform, agent,})} + V_{(\text{domain, 艺}|\text{entertainment})}$$

Due to the incompleteness of manual annotations, for the extended elaborations, people have annotated less than they actually exist, e.g., there are 114,876 word senses whose extended elaborations are empty. Therefore, we add a variable for each word sense to represent its missing information. And the final version is:

$$V_{p_i} = V_{\tilde{s}_{p_i}} + \sum_{(h_j, r_j, t_j) \in T_{p_i}, h_j = \tilde{s}_{p_i}} V_{(r_j, t_j)} + \sum_{(h_k, r_k, t_k) \in T_{p_i}, t_k = \tilde{s}_{p_i}} V_{(h_k, r_k,)} + V_{p_i}^s, \forall p_i \in H^p \quad (2)$$

e.g., the word sense $p_i = (\text{歌舞团}|\text{chorus})$ in Fig. 2 can be represented as:

$$\begin{aligned} V_{\text{歌舞团}|\text{chorus}} &= V_{\text{团体}|\text{community}} + V_{(\text{表演}|\text{perform, agent},)} \\ &\quad + V_{(, \text{domain, 艺}|\text{entertainment})} + V_{\text{歌舞团}|\text{chorus}}^s \end{aligned}$$

HowNet Word Representation. In HowNet, a word corresponds to single or multiple word senses, and we use $P_{w_i^z}$ and $P_{w_i^e}$ to denote the word sense set that a Chinese word w_i^z or an English word w_i^e corresponds to respectively, e.g., for English word $w_i^e = \text{"kid"}$ in Fig. 1, its corresponding word sense set $P_{w_i^e} = \{(\text{孩子}|\text{kid}), (\text{逗}|\text{kid}), (\text{羔羊}|\text{kid})\}$. To model a word in O^s , we calculate the average semantics of the word senses it corresponds to, i.e., the representations of a Chinese word w_i^z and an English word w_i^e are:

$$\begin{aligned} V_{w_i^z} &= \frac{\sum_{p_j \in P_{w_i^z}} V_{p_j}}{|P_{w_i^z}|}, \forall w_i^z \in H^z \\ V_{w_i^e} &= \frac{\sum_{p_j \in P_{w_i^e}} V_{p_j}}{|P_{w_i^e}|}, \forall w_i^e \in H^e \end{aligned} \quad (3)$$

3.3 Target Data in Sememe Space O^s

In this part, we will show how to model target words and target word senses in O^s .

Target Word Representation. As it is verified in [7], similar words should have similar sememes. We assume that similar words are more likely to have similar categorial sememes based on their work and data observation. A few counterexamples still exist, but this is why we need further study for the task. Although we cannot directly obtain the representations in O^s for words in D^z and D^e , we can use similar words to achieve an approximation. As stated in Sect. 3.1, word embeddings in O reflects the co-occurrence probability of the context, and words that are semantically similar are more likely to have similar contexts. Therefore, we use an idea similar to collaborative filtering, seeking similar words from HowNet to represent words in D^z and D^e . Representations obtained by this way are denoted as $V_{w_i^z}^d$ and $V_{w_i^e}^d$:

$$\begin{aligned} V_{w_i^z}^d &= \sum_{w_j^z \in H^z} \cos(\boldsymbol{w}_i^z, \boldsymbol{w}_j^z) \cdot V_{w_j^z} \cdot c^{r_j}, \forall w_i^z \in D^z \\ V_{w_i^e}^d &= \sum_{w_j^e \in H^e} \cos(\boldsymbol{w}_i^e, \boldsymbol{w}_j^e) \cdot V_{w_j^e} \cdot c^{r_j}, \forall w_i^e \in D^e \end{aligned} \quad (4)$$

where bolded \boldsymbol{w}_j^z , \boldsymbol{w}_j^e are the pre-trained embedding vectors in O for w_j^z and w_j^e . $\cos(\cdot, \cdot)$ returns the cosine similarity of input vectors. $V_{w_i^z}$ and $V_{w_i^e}$ are HowNet word representations in O^s . $c \in (0, 1)$ is a hyper-parameter that represent the descending factor, and r_j is the descend rank of word similarity. The idea behind c^{r_j} is that, we want to pay more attention on the most similar words.

Target Word Sense Representation. The sememe space we proposed is interpretable, each dimension of the vector in the space represents a corresponding weight score of one sememe. So for word senses in D^p , we use the average semantics of the corresponding Chinese and English words to calculate their representations in O^s , denoted as V_p^d . Various combination ways can be employed, and we simply choose their sum, i.e.,

$$V_{p_i}^d = V_{w_i^z}^d + V_{w_i^e}^d, \forall p_i \in D^p \quad (5)$$

3.4 Training and Prediction

In this section, we introduce the model prediction and training.

Prediction. As defined in Sect. 3.1, there is one-to-one correspondence between dimensions and sememes in sememe space O^s . Therefore, once we obtained the representations in O^s for $p \in D^p$ using Eq. 5, we can easily predict its categorial sememe by assuming $score(s_j|p) \propto V_p^d[j]$. We use \tilde{p}_p to represent the predicted categorial sememe. So we have:

$$\begin{aligned} \tilde{p}_p &= s_j, \\ \text{where } j &= \arg \max_j V_p^d[j] \end{aligned} \quad (6)$$

Note that all vectors are normalized after the operation in previous equations.

Training. The goal of training is to learn all of the different triple descriptions $V_{(r_j, t_j)}$, $V_{(h_k, r_k)}$ and word pair related supplementary information V_p^s in Eq. 2, so that we can properly represent words and word senses in HowNet. At first, we randomly initialize these variables, and calculate the words and word senses vectors in O^s using Eqs. 2 and 3. Then, for each word sense p in HowNet, we assume not to know its categorial sememe \tilde{s}_p , and make prediction to get predicted categorial sememe \tilde{p}_p using Eqs. 4 and 5. The objective is to make correct prediction, i.e., to minimize the following loss function:

$$\mathcal{L} = \sum_{p \in P} \tilde{s}_p \otimes \tilde{p}_p \quad (7)$$

where \otimes is the XOR operator, which returns 0 when the two sememes are the same and 1 otherwise.

During training, if $\tilde{s}_p \otimes \tilde{p}_p = 1$, we need to adjust $V_{(r_j, t_j)}$, $V_{(h_k, r_k)}$ and V_p^s to make V_p^d closer to $V_{\tilde{s}_p}$. Our gradient direction is not based on the derivative of the loss function, when $\tilde{s}_p \otimes \tilde{p}_p = 1$, we use $V_{\tilde{s}_p}$ as the gradient direction of V_p^d in Eq. 5 and use gradient descent method to update $V_{(r_j, t_j)}$, $V_{(h_k, r_k)}$ and V_p^s .

4 Experiment

In this section, we evaluate the proposed method. We first introduce the datasets and experiment settings, then report the overall results, and finally investigate several factors such as training set ratio, POS tags, ambiguity, and descending factor that might influence the prediction.

4.1 Datasets

HowNet and Oxford. Since we propose a novel task, and no benchmark datasets for categorial sememe prediction have been made available, we construct the evaluation datasets based on HowNet and Oxford English-Chinese Dictionary³.

Word Vector Space. Our proposed method requires both Chinese and English words to have embeddings in word vector space O , so we seek the publicly available pre-trained word vectors:

- For Chinese words, we choose the model from Tencent AI Lab⁴, which consists of about 8 million words and phrases.
- For English words, we use the Glove⁵ vectors, which contains 1.9 million words and phrases trained from 42 billion tokens.

After filtering out those word pairs that are not present in O , we achieve the statistics in Table 1. H and OX represent HowNet and Oxford Dictionary respectively. OX-H represents the Oxford Dictionary after filtering out the word pairs in HowNet.

Annotations. For HowNet dataset, all word pairs have sememe semantics, we can arbitrarily split the dataset for training and testing. For the Oxford Dictionary, we randomly select 1000 word pairs from those that are not in HowNet (denoted as “OX-H” in Table 1), and invite 5 experts who are proficient in Chinese, English and HowNet to annotate categorial sememes for them to construct the test data in Oxford.

Table 1. Datasets statistics

Data	Words		Word pairs		Exist in O
	Chinese	English	Total		
H	104,027	118,347	208,276	93,081	
OX	91,296	189,889	355,234	137,864	
OX-H	–	–	–	113,014	

Table 2. Overall results

Method	HowNet			Oxford		
	MAP	Hit@1	Hit@3	MAP	Hit@1	Hit@3
Basic	0.827	0.748	0.892	0.737	0.627	0.826
SPWE+	0.742	0.687	0.800	0.649	0.596	0.706
SPWE*	0.746	0.694	0.800	0.648	0.596	0.700
SS-RE	0.847	0.775	0.907	0.820	0.727	0.905
SS	0.858	0.790	0.919	0.843	0.757	0.919

4.2 Experiment Settings

Baseline Methods. We denote our model using sememe space as SS. To evaluate its effectiveness, we compare it with the following methods:

³ We use the New Oxford English-Chinese Dictionary, second edition published by Shanghai foreign language education press in 2013.

⁴ <https://ai.tencent.com/ailab/nlp/embedding.html>.

⁵ <https://nlp.stanford.edu/projects/glove/>.

- **Basic:** It is a straightforward model. In the basic model, we solve the problem with a classification model, which takes the Chinese word vector and the English word vector in O as features and outputs the categorial sememe via the softmax regression method.
- **SPWE variants:** SPWE is state-of-the-art mono-lingual sememe prediction model proposed in [7], whose idea is like collaborative filtering:

$$\text{score}(s_k|p_i) = \sum_{p_j \in H^p} \text{sim}(p_i, p_j) \cdot M_{jk} \cdot c^{r_j} \quad (8)$$

The notations H^p and c^{r_j} are the same as those in Eq. 4. M is the word pair-categorial sememe matrix, where M_{ij} is equal to 1 if the word pair p_i has the categorial sememe s_j in HowNet, otherwise M_{ij} is equal to 0. The key factor is how to measure the word pair similarity, and we adapt the mono-lingual word similarity to word pairs, and achieve the following two versions:

SPWE+:

$$\text{sim}(p_i, p_j) = \cos(\mathbf{w}_i^z, \mathbf{w}_j^z) + \cos(\mathbf{w}_i^e, \mathbf{w}_j^e)$$

SPWE*:

$$\text{sim}(p_i, p_j) = \cos(\mathbf{w}_i^z, \mathbf{w}_j^z) * \cos(\mathbf{w}_i^e, \mathbf{w}_j^e)$$

- **SS Variants:** In order to verify the validity of the sememe space, we also compare **SS** with its simplified version by ignoring the extended elaborations and supplementary information in Eq. 2 (thus denoted as **SS-RE**), i.e., the vector of word sense in O^s is a one-hot constant vector inherited from the categorial sememe.

Evaluation Protocol. For each word pair, our model outputs the score of all sememes in descending order. We use mean average precision (MAP), Hit@1 and Hit@3 as evaluation metrics, where Hit@n represents the correct categorial sememe appears in the top n of the list.

Parameter Settings. To make the comparison fair, all methods use the same word vector space O described previously. The dimensions of Chinese and English pre-trained word vectors are 200 and 300 respectively. We set c in Eq. 4 to 0.5, and use the top 100 most similar objects in the collaborative filtering steps. The default ratio of training, validation and test set is 8:1:1 within HowNet. The learning rate is 0.3, and we stop training when the loss on validation set does not drop.

4.3 Overall Results

Table 2 shows the overall results, from which we can find that our proposed **SS** model outperforms all single models on both datasets, which proves its rationality and effectiveness. Specifically,

- **Comparison with SPWE variants:** **SS**, **SPWE+** and **SPWE*** are all based on the collaborative filtering framework, but **SS** far exceeds the other two models. The difference is that **SPWE+** and **SPWE*** do not use the sememe space, but directly use the similarity between word pairs for prediction. Thus, the results confirm the significance of the sememe space.

- **Comparison with its variants:** **SS** outperforms **SS-RE**, indicating that the extended elaborations and supplementary information can enrich the word pair semantic modeling. Meanwhile, **SS-RE** outperforms all other models except **SS**, which also validates the rationality and effectiveness of the sememe space. Furthermore, compared with **SS**, **SS-RE** ignores the extended elaborations and supplementary information, and thus does not require training. Therefore, if a simple and efficient model is needed, **SS-RE** will be a good choice.

4.4 Results on Different POS Tags

Figure 4(a) shows the results of different POS tags in HowNet datasets. Accordingly, Table 3 presents the number of word pairs and corresponding categorial sememes.

Intuitively, the prediction (classification) is easier with a small number of categorial sememes. From Fig. 4(a), adjectives, verbs and adverbs all conform to this law, but the nouns do not. At first, we assume that the large number of noun word pairs provide sufficient training for the model. However, the best performing adverb has the least word pairs, so the number of word pairs is not the only cause of performance. Another possible reason from the linguistic perspective is that nouns are not very abstract and easier to be expressed by semantic combination.

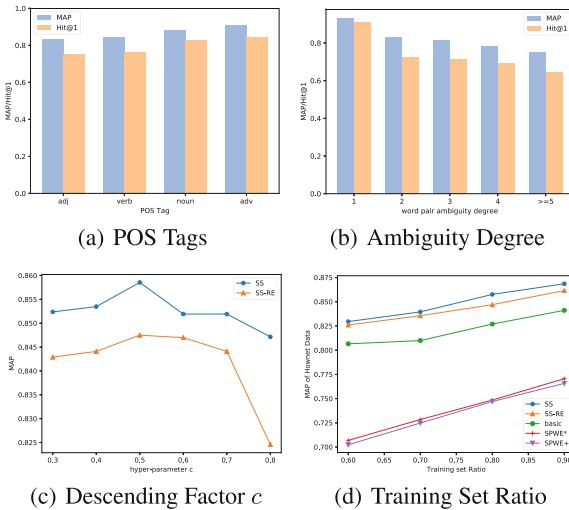


Fig. 4. Results on different experiment settings

4.5 Results on Different Ambiguity Degrees

As mentioned and used in Sect. 3.2, a word corresponds to single or multiple word pairs and each word pair has a categorial sememe, so a word may have multiple different categorial sememes. We define the total number of unique categorial sememes owned by a word as its **word ambiguity degree**. A word pair has a Chinese word and an

English word. We use the larger of the two word ambiguity degrees as the **word pair ambiguity degree**. Table 4 shows the number of word pairs with different ambiguity degrees in HowNet test data.

Table 3. POS tags statistics of test data

POS Tag	adj	Verb	Noun	adv
# word pair	1,874	1,960	4,849	505
# \tilde{s}_p	1,107	790	1,273	448

Table 4. Ambiguity degrees of test data

Ambiguity degree	1	2	3	4	≥ 5
# word pair	3,717	2,386	1,270	730	1,205

4.6 Effect of Descending Factor c

From the results shown in Fig. 4(b), it can be found that ambiguity has an impact on the prediction. Intuitively, the more ambiguous the word pair is, the more difficult it is to predict. The results are in good agreement with such common sense.

The descending factor in Eq. 4 controls the descending speed of the word weight. A smaller c results in a faster descending weight, which means that the prediction result mainly focuses on the top few words in the similarity ranking list. And a larger c indicates the prediction depends on more similar words in the ranking list. Figure 4(c) presents the results with c ranging from 0.3 to 0.8. The results show that, whether c is too small or too large, the prediction results will drop. The reason is that, using only a few words (small c) causes insufficient information utilization, while using too many words will introduce noise.

4.7 Effect of Training Set Ratio

In this experiment, we first sample a certain ratio of data as a training set, and the remaining data is divided into a validation set and a test set according to 1:1. Figure 4(d) shows the effect of training set ratio. It can be seen that both **SS** and **SS-RE** model have very good robustness, and the performance declines slowly as the training data decreases.

4.8 Categorial Sememe Knowledge Base

We select 113,014 English-Chinese word pairs that do not exist in HowNet from the Oxford bilingual dictionary, and use **SS** to predict. Then we create annotation website⁶ and invite experts to manually correct all prediction results. Each word pair is corrected by three linguistic experts. We published the expanded data and provided online prediction services on the website⁷.

⁶ <http://166.111.68.66:3080/AnnoTool/>.

⁷ http://thukeg.gitee.io/categorial_sememe/.

5 Related Work

Most researchers focus on the application of SKB, but pay little attention to how to automatically build it. Recently, few works attempted to automatically build the SKB [7–9]. [7] used collaborative filtering and matrix factorization to learn relationships between sememes and words to predict the sememe set for each word. [8] utilized cross-lingual information and employed word representation learning and collaborative filtering to predict the sememe set for each word. [9] incorporated characters for Chinese word sememe prediction which is not universal for all languages.

Except for low prediction accuracy, all the previous works have two limitations. Firstly, these methods did not recognize which one is the categorial sememe, and neglected semantic relations as well. Secondly, one word could have multiple word senses. Previous works treated all words as unambiguous words to predict.

6 Conclusion and Future Work

In this paper, we introduced a new task of ECCSP and used cross-lingual information to address the word sense ambiguity. We proposed the sememe space, where each dimension has explainable semantic meaning and proposed several models based on sememe space and word vector space. We evaluated our models with HowNet and Oxford English-Chinese Dictionary. The results revealed the effectiveness and significance of our models and we expanded HowNet nearly by 50%.

We will explore the following directions in the future: (1) We will explore better methods to learn vectors in sememe space. Various combination strategies can be employed when an object has multiple sources of information, and we simply choose their normalized sum in this paper. (2) Sememes are the minimum unambiguous semantic units of human languages, which are believed to be universal for all languages. Both Chinese and English in our experiment are used by a large number of people, and we will explore to add sememe knowledge to other languages. (3) The limited sememe set are used to express all word senses. The sememe space use sememe as basis vector. We will try to apply sememe space to other NLP tasks.

Acknowledgement. This work is supported by the NSFC Youth Project (62006136), and grants from the Institute for Guo Qiang, Tsinghua University (2019GQB0003) and Huawei Inc.

References

1. Bloomfield, L.: A set of postulates for the science of language. *Language* **2**(3), 153–164 (1926)
2. Dong, Z., Dong, Q.: HowNet - a hybrid language and knowledge resource. In: International Conference on Natural Language Processing and Knowledge Engineering, Proceedings, pp. 820–824 (2003)
3. Duan, X., Zhao, J., Xu, B.: Word sense disambiguation through sememe labeling. In: IJCAI, pp. 1594–1599 (2007)

4. Zeng, X., Yang, C., Tu, C., Liu, Z., Sun, M.: Chinese LIWC lexicon expansion via hierarchical classification of word embeddings with sememe attention. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
5. Qi, F., et al.: Modeling semantic compositionality with sememe knowledge. In: ACL 2019: The 57th Annual Meeting of the Association for Computational Linguistics, pp. 5706–5715 (2019)
6. Niu, Y., Xie, R., Liu, Z., Sun, M.: Improved word representation learning with sememes. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, pp. 2049–2058 (2017)
7. Xie, R., Yuan, X., Liu, Z., Sun, M.: Lexical sememe prediction via word embeddings and matrix factorization. In: International Joint Conference on Artificial Intelligence (2017)
8. Qi, F., Lin, Y., Sun, M., Zhu, H., Xie, R., Liu, Z.: Cross-lingual lexical sememe prediction. In: EMNLP 2018: Conference on Empirical Methods in Natural Language Processing, pp. 358–368 (2018)
9. Jin, H., et al.: Incorporating Chinese characters of words for lexical sememe prediction. In: ACL 2018: 56th Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 2439–2449 (2018)
10. Resnik, P., Yarowsky, D.: Distinguishing systems and distinguishing senses: new evaluation methods for word sense disambiguation. Nat. Lang. Eng. **5**(2), 113–133 (1999)
11. Adriani, M.: Using statistical term similarity for sense disambiguation in cross-language information retrieval. Inf. Retrieval **2**(1), 71–82 (2000)
12. Chen, K.-J., Huang, S.-L., Shih, Y.-Y., Chen, Y.-J.: Extended-hownet: a representational framework for concepts. In: Proceedings of OntoLex 2005-Ontologies and Lexical Resources (2005)
13. Dong, Z., Dong, Q.: HowNet and the Computation Of Meaning. World Scientific (2006)
14. Ding, N., Li, Z., Liu, Z., Zheng, H., Lin, Z.: Event detection with trigger-aware lattice neural network. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 347–356 (2019)
15. Li, Z., Ding, N., Liu, Z., Zheng, H., Shen, Y.: Chinese relation extraction with multi-grained information and external linguistic knowledge. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4377–4386 (2019)
16. Zang, Y., et al.: Textual adversarial attack as combinatorial optimization. Computation Computation and Language (2019)



Multi-level Cohesion Information Modeling for Better Written and Dialogue Discourse Parsing

Jinfeng Wang^{1,2}, Longyin Zhang^{1,2}, and Fang Kong^{1,2(✉)}

¹ Institute of Artificial Intelligence, Soochow University, Suzhou, China
 {jfwang9,lyzhang9}@stu.suda.edu.cn, kongfang@suda.edu.cn

² School of Computer Science and Technology, Soochow University, Suzhou, China

Abstract. Discourse parsing has attracted more and more attention due to its importance on Natural Language Understanding. Accordingly, various neural models proposed and have achieved certain success. However, due to the scale limitation of corpus, outstanding performance still depends on additional features. Different from previous neural studies employing simple flat word level EDU (Elementary Discourse Unit) representation, we improve the performance of discourse parsing by employing cohesion information (In this paper, we regard lexical chain and coreference chain as cohesion information) enhanced EDU representation. In particular, firstly we use WordNet and a coreference resolution model to extract lexical and coreference chain respectively and automatically. Secondly, we construct EDU level graph based on the extracted chains. Finally, using Graph Attention Network, we incorporate the obtained cohesion information into EDU representation to improve discourse parsing. Experiments on RST-DT, CDTB and STAC show our proposed cohesion information enhanced EDU representation can benefit both written and dialogue discourse parsing, compared with the baseline model we duplicated.

Keywords: Discourse parsing · Cohesion information · GAT · Written and dialogue text

1 Introduction

Discourse parsing aims to identify the relations and discover the discourse structure between discourse units. Due to the ability of providing the overall organization of an article, discourse parsing plays a central role in various downstream tasks, such as dialogue understanding [20], text categorization [10], and text summarization [22].

For written text, discourse parsing mainly based on the Rhetorical Structure Theory (RST), using hierarchical tree to represent the discourse structure. All illustrated in Fig. 1 (a), each leaf node in the *constituency-based* tree corresponds to an Elementary Discourse Unit (EDU). Each pair of related adjacent discourse units will be merged by specific rhetorical relations to form upper-layer Discourse

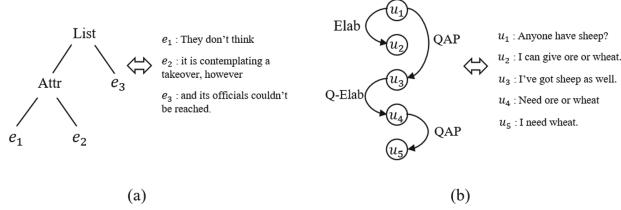


Fig. 1. (a) a written text example and its *constituency-based* tree structure from RST-DT Corpus; (b) a *dependency-based* graph structure for a dialogue text example from STAC Corpus.

Unit (DU) recursively. Moreover, each DU is assigned with either *nucleus* or *satellite* according to the importance.

By contrast, multi-party dialogue text which has more complex discourse structure in nature, uses *dependency-based* Directed Acyclic Graph (DAG) to express the structure of the text. As shown in Fig. 1 (b), EDUs¹ are directly linked without forming upper-layer structure.

Early studies on discourse parsing mainly donated their efforts on constructing handcraft features [8, 12, 13, 16], due to the corpus size limitation. Recent studies turn to top-down approaches for better leverage global document context. However, discourse parsing still has a long way to go.

In this paper, we put our sight on modeling discourse cohesion for better written and dialogue discourse parsing. Theoretically, discourse cohesion mainly consists of lexical cohesion, reference, ellipsis, substitution and conjunction. Lexical chain and coreference chain are regarded as cohesion information in this work. First, we duplicate three models as baseline. Second, cohesion information is obtained with the help of external tools. Third, we integrate the cohesion information into EDU representation with the help of two Graph Attention Network [21], and utilize a fusion layer to fuse all the obtained representations. Finally, the model employ the representation with cohesion information to identify the relations and structure for both written and dialogue texts.

Experimental results on the RST-DT, CDTB and STAC corpora show that our approach is effective for both written discourse constituency parsing and dialogue discourse dependency parsing.

2 Related Work

In the literature, previous work on written discourse parsing mainly consist of two categories, i.e., bottom-up [8, 9, 12, 13, 16] and top-down [11, 24] frameworks.

For the first category, discourse parser builds a tree by merging two adjacent discourse units into a large one recursively from bottom to up, where nuclearity and rhetorical relations were labeled simultaneously. Feng et al. [8] proposed

¹ In the dialogue text, each utterance corresponds to an EDU.

a two-stage bottom-up greedy parser with CRF as local classifier. Ji et al. [9] and Li et al. [13] focus on better representation using feed forward network and hierarchical neural network. Li et al. [12] proposed a recursive framework to jointly model distributed representation for discourse parsing. Yu et al. [16] build a transition-based parser with implicit syntactic features.

For the second category, by recursively splitting a larger text span into two small ones, discourse parser can build a tree in top-down fashion, where nuclearity and rhetorical relation were determined at the same time. Zhang et al. [23] introduced a unified neural architecture toward discourse rhetorical structure, which achieved competitive performance on both Chinese and English without any handcrafted features. Kobayashi et al. [11] proposed a neural top-down discourse parser in three granularities, i.e., document, paragraph and sentence level.

Previous studies for dialogue discourse dependency parsing can be divided into two categories. The first-class models parse discourse dependency structure in two stages. These approaches [1, 17] predict the local probability of dependency relation for each possible combination of EDU pairs, and then apply a decoding algorithm to construct the final structure. Afantinos et al. [1] used Maximum Spanning Trees (MST) to construct a dependency tree. Perret et al. [17] further used Integer Linear Programming (ILP) to construct a dependency graph. For the second category [18], discourse parser predicts dependency relation and constructs the discourse structure jointly and alternately. Shi et al. [18] proposed a sequential model based on neural network and utilized the currently constructed structure in dependency prediction.

Although neural networks have been widely applied in discourse parsing, the state-of-the-art model still relies on handcrafted features. We found that the cohesion information which can benefit this task has not received much attention. In this paper, we propose a Graph Attention Network based model to incorporate cohesion information into EDU representation, and use the updated representation for discourse parsing.

3 Baseline Model

In order to verify the effect of our proposed method of using cohesion information to enhanced EDU representation, we duplicate three baseline models: A top-down and a bottom-up model for written discourse parsing, a deep sequential model for dialogue discourse parsing. In this section, we will first introduce the EDU encoder and then give a brief introduction to these three models.

3.1 Attention-Based EDU Encoder

Given an EDU with m words w_1, \dots, w_m , where w_j is the vector concatenated by word embedding and POS embedding of the j -th words, we use a bidirectional GRU (BiGRU) [5] to encode EDU, obtaining h_1, \dots, h_m . We concatenate the last states in both direction of the BiGRU into x' . According to our common sense, the importance of each word in EDUs is usually different. On this basis, a

learnable vector q is implemented to calculate the weight assigned to each word. The detail is formulated as

$$w_i = \frac{q^T h_i}{\sum_{j=1}^n q^T h_j} \quad (1)$$

$$x_k = (x' + w_i h_i) \quad (2)$$

where $\alpha = \{\alpha_1, \dots, \alpha_m\}$ denotes the output weight of a softmax function with w_i, \dots, w_m as inputs, h_i corresponds to the output of embedding of each token in an EDU encoding by BiGRU. In this way, representation for EDUs in an entire discourse are generated, denoted as $X_o = \{x_1, \dots, x_n\}$.

3.2 Top-Down Baseline Model

For RST discourse parser based on Top-Down approach, we refer to the top-down model proposed by Zhang et al. [23]. The model mainly consists of two parts: Hierarchical Split Point Encoding and Top-Down Split Point Ranking.

Hierarchical Split Point Encoding. For split point² representation, Zhang et al. [23] used a hierarchical RNN-CNN architecture in their paper. Firstly, each EDU representation x_1, \dots, x_n was obtained by Attention-based Encoder. Then, another BiGRU is used to model EDU context. Finally, a CNN with a window size of 2 and stride size of 1 is used to encode split point.

Top-Down Split Point Ranking. After obtaining split point representations, an encoder-decoder is used to rank the split points. During encoding, the split point vectors which obtained previously are taken as inputs to the BiGRU encoder, obtaining H_0, H_1, \dots, H_{n-2} ³. During decoding, a uni-GRU with an internal stack is used to control the split point ranking process. At the j -th step, the tuple (B, E) is popped from the stack and we enter the concatenated $c_j = (H_B; H_E)$ into the decoder for d_j . After that, three biaffine function [6] based classifiers will be used to predict structure, nuclearity and relation respectively.

3.3 Bottom-Up Baseline Model

The Bottom-Up approach mainly include two kinds categories, i.e., Probabilistic CKY-like approaches and transition-based approaches. In this work, we build a bottom-up baseline model based on the latter one.

² The split position between any two neighboring EDUs is called the split point.

³ There will be $n - 2$ split points for n EDUs.

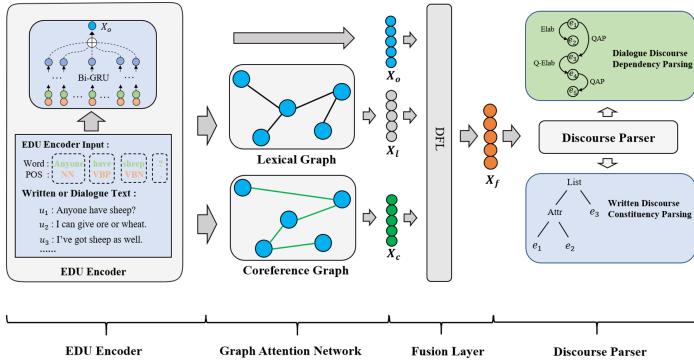


Fig. 2. The general architecture for our CGGAN model

Shift-Reduce Discourse Parsing. A shift-reduce parser maintains a stack and a queue; Initially the stack is empty and the first EDU x_1 is at the front of the queue. The parser then choose either to *shift* the element which is the front of the queue onto the top of the stack, or to *reduce* the top two elements on the stack in a discourse relation. The *reduce* operation must choose the type of relation and decide which element is nucleus.

3.4 Deep Sequential Baseline Model

For dialogue discourse dependency parsing, we duplicate the deep sequential model proposed by Shi et al. [18]. Their baseline model consists of two parts: Non-Structured Representation Construction, Link and Relation Prediction.

Non-structured Representation Construction. For each EDU u_i , firstly we use the attention-based EDU encoder to get EDU representation x_1, \dots, x_n . Then, these representation is taken as input to a global BiGRU encoder and the hidden states of BiGRU are viewed as the non-structured global representation of EDUs.

Link and Relation Prediction. For each EDU u_i , the link predictor predicts its parent node p_i , and the relation classifier categorizes the corresponding relation type r_i . In particular, the link predictor is a 2-class classifier and the relation predictor is a 16-class classifier.

4 Cohesion Modeling

In this section, we introduce the **Cohesion Guided Graph Attention Network (CGGAN)** for cohesion information modeling. The CGGAN includes four parts: (1) Auto Cohesion Information Extraction (2) Graph Construction (3) Cohesion Modeling (4) Fusion Layer, as shown in Fig. 2.

4.1 Auto Cohesion Information Extraction

As mentioned before, the lexical and coreference chain are regraded as cohesion information in this work. We will introduce how to extract cohesion information in this section.

Auto Lexical Chain Extraction. Lexical chain are sequences of semantically related words [15], which describe the lexical cohesion structure of an entire article. In this work, we use WordNet and *Tongyici cilin* (Cilin for short) to define word senses, relations and semantics for English and Chinese corpus respectively. Only nouns are selected as candidate words for lexical chain computation.

For English Corpus, we use the *path_similarity*⁴ for word similarity computation; for Chinese corpus, the method proposed by Zhu et al. [25] will be used to calculate Chinese word similarity. A threshold is set as a filter to select words for lexical. For a given document, we achieve a corresponding lexical set, $C_l = \{C_{l1}, C_{l2}, \dots, C_{ln}\}$, where $C_{li} = \{W_{i1}, W_{i2}, \dots, W_{in}\}$ and W_{im} means the m -th word of the i -th chain.

Auto Coreference Chain Extraction. Coreference chain is a set of mentions which refer to the same entity in the real world. In this work, we employ a coreference resolution model (CRM) proposed by Kong et al. [7] to detect all the coreference chains in a document for both English and Chinese Corpus. In particular, as the dialogue coreference resolution is still in its infancy, we also use this CRM to analyze the coreference chains of dialogue texts. Finally, the coreference chain set $C_c = \{C_{c1}, C_{c2}, \dots, C_{cn}\}$ will be obtained, where $C_{ci} = \{M_{i1}, M_{i2}, \dots, M_{in}\}$ and M_{im} means the m -th mention of the i -th chain.

4.2 Graph Construction

We propose a Lexical Graph (\mathcal{G}_L) and a Coreference Graph (\mathcal{G}_C) built upon discourse units. Algorithm 4.2 describes how to construct both Graph⁵. For each chain, all the discourse units containing the element⁶ of same chain will be connected. This iterate over all the lexical chains to generate the final Graph (\mathcal{G}_L or \mathcal{G}_C). In particular, \mathcal{G}_L and \mathcal{G}_C are symmetric and self-loop is added to all the nodes.

Given the constructed graph $G = (V, E)$, the nodes V correspond to the EDUs in a document, and the edges E correspond to either the lexical chain or coreference chain. We then use Graph Attention Network [21] to update the representation of all EDUs, based on the built graph.

⁴ A word similarity calculation method provided by WordNet, return a score between 0 and 1, denoting how similar two word senses are, based on the shortest path that connects the senses. Moreover, when there is no path between two senses, -1 will be returned.

⁵ we use the same method to build lexical and coreference graph.

⁶ In order to simplify the expression, the word and mention in lexical and coreference chain are collectively referred to as *element*.

Algorithm 1. Construction of the Cohesion Graph \mathcal{G}

Require: Cohesion Chain $C = \{C_1, C_2, \dots, C_n\}$;
elements for each chain $C_i = \{W_{i1}, \dots, W_{im}\}$
Initialize the Graph \mathcal{G} without any edge $\mathcal{G}[*][*] = 0$
for $i = 0$ to n **do**
 Collect the location of all occurrences $\{W_{i1}, \dots, W_{im}\}$
 to $L = \{l_1, \dots, l_m\}$
 for $j = 1$ to m **do**
 $\mathcal{G}[j][j] = 1$
 for $k = 1$ to m **do**
 $\mathcal{G}[j][k] = 1$
 end for
 end for
end for
return Constructed Graph \mathcal{G}

4.3 Cohesion Modelling

As shown in Fig. 2, firstly we obtain the EDU representation $X_o = \{x_1, x_2, \dots, x_n\}$ which is the output of the EDU Encoder. The CGGAN is applied to integrate cohesion information into EDU representation before it is used for discourse parsing. The nodes in the CGGAN are initialized by the original EDU representation $x_i \in X_o$ respectively.

Graph Attention Networks (GAT) [21], leveraging masked self-attention layers to assign different importance to neighboring nodes. Formula 3–6 show the process of representation updating of node i in lexical and coreference graph.

$$z_i = Wx_i \quad (3)$$

$$e_{ij} = \text{LeakyReLU}(A(z_i; z_j)) \quad (4)$$

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})} \quad (5)$$

$$x'_i = \sigma \left(\sum_{j \in \mathcal{N}_i} a_{ij} z_{ij} \right) \quad (6)$$

Where ; represents concatenate operation, W and A are learnable parameters of the model, \mathcal{N}_i means all the neighbors of node i and σ is the activation function.

We use two different GAT layers to incorporate \mathcal{G}_L and \mathcal{G}_C into EDU representations. Given EDU representation of a document X_o , we will get two kinds of updated representation X_l and X_c , which are integrated with lexical and coreference chain respectively.

4.4 Fusion Layer

Now, we get the updated representations X_l and X_c using Graph Attention Network. We propose a fusion layer to fuse these two representation with the original one X_o .

Firstly, the attention score between the updated representation and the original one is calculated. Then, the scores will be taken as the weight of the updated representation, when added with the original one.

$$\alpha_l = \text{softmax}(W_l(X_o; X_l) + b_l)) \quad (7)$$

$$\alpha_c = \text{softmax}(W_c(X_o; X_c) + b_c)) \quad (8)$$

$$X_f = X_o + \alpha_l * X_l + \alpha_c * X_c \quad (9)$$

Where ; is concatenate operation, W_l , W_c , b_l , b_c are learnable parameters.

5 Experiments

5.1 Datasets

In this work, the written discourse parsing is experimented on two datasets, Rhetorical Structure Theory Discourse Treebank (RST-DT) [4] and Chinese Connective-driven Discourse Treebank (CDTB) [14]. What's more, STAC is used as the benchmark corpus for dialogue discourse parsing.

- The RST-DT corpus is composed of 385 articles from the Wall Street Journal (WSJ), where 347 and 38 articles for training and testing respectively. Following previous work [24], we randomly select 34 articles from training set as validation.
- In the CDTB corpus, each paragraph is annotated as a discourse tree. The entire corpus is divided into three parts, i.e., 2002, 105, 229 CDT trees for training, validation and testing.
- The STAC corpus [2]⁷ annotated according to Segmented Discourse Representation Theory (SDRT) [3]. Following previous studies [1, 17, 18], we transformed SDRT structure to dependency structure by eliminating CDUs. After that, training set contains 1062 dialogues, while testing set consists of 111 dialogues, and we retained 10% of the training set for validation.

5.2 Metric

To evaluate the parsing performance, we use three standard ways to measure the performance. We use 18, 16, 16 fine-grained relations for RST-DT, CDTB, STAC respectively, as Zhang et al. [18, 24]. Following previous work, micro averaged F_1 score is the evaluate metric of RST-DT and STAC, while macro averaged F_1 score for CDTB.

5.3 Experimental Result

Baseline Result. Table 1 shows the detailed performance of our duplicated baseline model. It should be emphasized that we mainly explore the effect of cohesion information in discourse parsing for both written and dialogue texts in this paper. All results in subsequent section are based on the duplicated models.

⁷ Following previous study, we used the version released on March 21, 2018.

Table 1. Results of baseline model on RST-DT, CDTB and STAC. “T2D” and “B2U” denotes our simplified architecture in Top-Down and Bottom-Up fashion.

System		S	N	R	F
EN	T2D	66.7	54.7	44.6	44.0
	B2U	67.1	55.3	43.7	43.3
CN	T2D	84.4	57.0	53.4	45.6
	B2U	83.6	55.5	50.3	47.1
System		Link	Link&Relation		
EN	DSM	71.1	53.6		

Table 2. Results for RST-DT, CDTB and STAC. “LC”, “CC”, “LC&CC” mean add lexical chain or coreference chain, or both of them respectively.

System		S	N	R	F	System		S	N	R	F
CN	T2D	84.4	57.0	53.4	45.6	B2U	83.6	55.5	50.3	47.1	
	+LC	85.5	57.7	53.7	46.4	+LC	84.5	55.8	51.5	47.4	
	+CC	85.4	58.0	53.8	46.9	+CC	84.8	56.4	51.7	47.2	
	+LC&CC	85.7	58.4	54.5	47.3	+LC&CC	85.4	56.6	52.4	47.9	
EN	T2D	66.7	54.7	44.6	44.0	B2U	67.1	55.3	43.7	43.3	
	+LC	67.2	55.2	45.3	44.2	+LC	67.6	55.6	44.6	44.2	
	+CC	66.9	55.2	45.2	44.3	+CC	67.8	55.9	44.6	44.5	
	+LC&CC	67.5	55.8	45.8	44.9	+LC&CC	68.2	56.3	45.3	44.9	
System		Link		Link&Relation							
DSM		71.1		53.6							
+LC		71.3		54.3							
+CC		72.3		54.8							
+LC&CC		72.5		55.2							

Contribution of Cohesion. As emphasized before, we explore the effect of cohesion information in discourse parsing by integrating the cohesion information into the EDU representation. Lexical chain (LC), coreference chain (CC) and all of them (LC&CC) will be incorporated into the baseline model respectively. By analysing the experimental results in Table 2, we can draw the conclusions as following:

- Lexical chain and coreference chain can effectively improve the performance of discourse parsing in all four indicators, compared with the baselines. And there is a certain complementary between them, when these two kinds of chains are added at the same time, the effect can be further improved.
- In the written text, in comparison with the top-down approach, the performance of bottom-up can be improved more obvious. This is because the

updated representation show more help to the detection of the structure and relations between the low-level nodes, While the top-down approach has lost part of the cohesion information when constructing the split point representation.

- In dialogue text, it is obvious that the *Link* and *Link&Rel* are both improved by 1.4% and 1.6%.
- For the reason of the much lower discourse tree, Chinese discourse parsing can get more benefits from cohesion information (each tree contains 4.2 EDUs on average [19]), improving greatly than English.

Table 3. Results of different chain length. 0 mean that chain is not used, 3, 5, ∞ means that the length of the chain is limit to 3 and 5, ∞ means do not limit chain's length.

		LC				CC				LC&CC				
		Length	S	N	R	F	S	N	R	F	S	N	R	F
T2D		0	66.7	54.7	44.6	44.0	66.7	54.7	44.6	44.0	66.7	54.7	44.6	44.0
		3	66.9	54.7	44.9	44.3	67.0	55.1	44.9	44.2	67.3	55.8	45.4	44.4
		5	67.2	55.2	45.3	44.2	66.9	55.4	45.2	44.3	67.5	55.8	45.8	44.9
		∞	66.4	54.2	44.3	43.9	66.5	53.9	44.6	44.3	66.7	54.9	44.7	44.2
EN		0	67.1	55.3	43.7	43.3	67.1	55.3	43.7	43.3	67.1	55.3	43.7	43.3
		3	67.4	55.1	44.2	44.0	67.5	55.9	44.4	44.1	67.8	56.1	44.8	44.5
		5	67.6	55.6	44.6	44.2	67.8	55.9	44.9	44.5	68.2	56.3	45.3	44.9
		∞	66.9	55.4	43.5	43.4	67.3	55.2	43.9	43.7	67.9	55.6	44.2	43.9
B2U		0	84.4	57.0	53.4	45.6	84.4	57.0	53.4	45.6	84.4	57.0	53.4	45.6
		3	85.5	57.7	53.7	46.4	85.4	58.0	53.8	46.9	85.7	58.4	54.5	47.3
		5	85.3	57.6	53.4	46.5	85.4	57.8	53.5	46.3	85.5	58.1	54.3	46.8
		∞	84.7	56.8	52.8	45.2	84.7	57.1	53.1	45.7	85.0	57.4	53.6	46.1
CN		0	83.6	55.5	50.3	47.1	83.6	55.5	50.3	47.1	83.6	55.5	50.3	47.1
		3	84.5	55.8	51.5	47.4	84.8	56.4	51.7	47.2	85.4	56.6	52.4	47.9
		5	84.0	55.4	50.8	47.2	64.2	55.9	51.3	47.0	85.2	56.2	52.1	47.6
		∞	83.6	55.2	50.4	46.9	83.8	55.9	50.5	47.1	84.1	55.8	50.8	47.3
		LC				CC				LC&CC				
		Length	Link	Link&Rel	Link	Link	Link&Rel	Link	Link&Rel	Link	Link	Link&Rel	Link&Rel	
DSM		0	71.1	53.6	71.1	53.6	71.1	53.6	71.1	53.6	71.1	53.6	53.6	
		3	71.1	54.0	71.7	54.3	71.8	54.3	71.8	55.0	71.8	55.0	55.0	
		5	71.3	54.3	72.3	54.8	72.5	55.2	72.5	55.2	72.5	55.2	55.2	
		∞	71.3	54.2	72.0	54.6	72.0	54.6	72.0	55.2	72.0	55.2	55.2	

Impact of Cohesion Chain Length. Generally, we believe that the structure and relation between EDUs have a strong connection with the distance between EDUs. The cohesion between two EDUs while are far apart is prone to be noisy. Therefore, we manually set a length limitation when use lexical and coreference

chains to build graph. From the results in Table 3, the following conclusions can be drawn:

- When the length of chains is not limited, a lot of noise is contained, leading to little improvement or even negative performance. On the contrary, after limiting the length of the chain, the integration of cohesion information can bring obviously positive improvement.
- In particular, in written text, the length of 3 is the more appropriate chain length limit for Chinese discourse parsing, and 5 for English. For the reason that the English discourse tree is annotated by discourse, while Chinese is annotated by paragraph.
- For dialogue texts, compared to a smaller length limit, a larger limitation is more conducive to the effect of mining cohesion information, even if there is no limit. This is because the relation in the dialogue might exist between two EDUs that are far apart. Reducing the length of the chain will result in the ineffective cohesion information when identify the relation between two EDUs with a larger span.

In general, the results indicate that the unlimited use of lexical chains may provide noise-filled conduction which is not applicable for cohesion information modeling, and an optimal distance value can maximize the effectiveness of cohesion information in discourse parsing.

6 Conclusion

In this research, we explored the effect of cohesion information on discourse parsing. Particularly, the Graph Attention Network is used to integrate the cohesion information into the EDU representation. Experimentation on the RST-DT, CDTB and STAC shows the great effectiveness of our proposed approach. Owing to the effective ascension, we will focus on exploring other cohesion information on discourse parsing.

Acknowledgements. The authors would like to thank the anonymous reviewers for the helpful comments. We are very grateful to Zixin Ni for her help in Reference Resolution we used in this work. This work was supported by Project 61876118 under the National Natural Science Foundation of China and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

1. Afantenos, S., Kow, E., Asher, N., Perret, J.: Discourse parsing for multi-party chat dialogues. Association for Computational Linguistics (ACL) (2015)
2. Asher, N., Hunter, J., Morey, M., Benamara, F., Afantenos, S.: Discourse structure and dialogue acts in multiparty dialogue: the STAC corpus (2016)
3. Asher, N., Lascarides, A.: Logics of Conversation. Peking University Press (2003)

4. Carlson, L., Marcu, D., Okurowski, M.E.: Building a discourse-tagged corpus in the framework of rhetorical structure theory. Association for Computational Linguistics (2001)
5. Cho, K., et al.: 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), pp. 1724–1734. Association for Computational Linguistics, Doha, Qatar, October 2014. <https://doi.org/10.3115/v1/D14-1179>. <https://www.aclweb.org/anthology/D14-1179>
6. Dozat, T., Manning, C.D.: Deep biaffine attention for neural dependency parsing (2016)
7. Fang, K., Fu, J.: Incorporating structural information for better coreference resolution. In: Twenty-Eighth International Joint Conference on Artificial Intelligence IJCAI-19 (2019)
8. Feng, V.W., Hirst, G.: A linear-time bottom-up discourse parser with constraints and post-editing. In: Meeting of the Association for Computational Linguistics (2014)
9. Ji, Y., Eisenstein, J.: Representation learning for text-level discourse parsing. In: Meeting of the Association for Computational Linguistics (2014)
10. Ji, Y., Smith, N.: Neural discourse structure for text categorization. arXiv preprint [arXiv:1702.01829](https://arxiv.org/abs/1702.01829) (2017)
11. Kobayashi, N., Hirao, T., Kamigaito, H., Okumura, M., Nagata, M.: Top-down RST parsing utilizing granularity levels in documents. In: Proceedings of the AAAI Conference on Artificial Intelligence (2020)
12. Li, J., Li, R., Hovy, E.: Recursive deep models for discourse parsing. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2061–2069 (2014)
13. Li, Q., Li, T., Chang, B.: Discourse parsing with attention-based hierarchical neural networks. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 362–371 (2016)
14. Li, Y., Feng, W., Jing, S., Fang, K., Zhou, G.: Building Chinese discourse corpus with connective-driven dependency tree structure. In: Conference on Empirical Methods in Natural Language Processing (2014)
15. Morris, J., Hirst, G.: Lexical cohesion computed by thesaural relations as an indicator of the structure of text. Comput. Linguistics (1991)
16. Nan, Y., Zhang, M., Fu, G.: Transition-based neural RST parsing with implicit syntax features (2018)
17. Perret, J., Afantinos, S., Asher, N., Morey, M.: Integer linear programming for discourse parsing. In: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016), pp. 99–109 (2016)
18. Shi, Z., Huang, M.: A deep sequential model for discourse parsing on multi-party dialogues (2018)
19. Sun, C., Fang, K.: A transition-based framework for Chinese discourse structure parsing. J. Chin. Inf. Process. (2018)
20. Takanobu, R., Huang, M., Zhao, Z., Li, F., Nie, L.: A weakly supervised method for topic segmentation and labeling in goal-oriented dialogues via reinforcement learning. In: Twenty-Seventh International Joint Conference on Artificial Intelligence IJCAI-18 (2018)
21. Velikovi, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks (2017)

22. Xu, J., Gan, Z., Cheng, Y., Liu, J.: Discourse-aware neural extractive model for text summarization (2019)
23. Zhang, L., Xing, Y., Kong, F., Li, P., Zhou, G.: A top-down neural architecture towards text-level parsing of discourse rhetorical structure. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (2020)
24. Zhang, L., Xing, Y., Kong, F., Li, P., Zhou, G.: A top-down neural architecture towards text-level parsing of discourse rhetorical structure. arXiv preprint [arXiv:2005.02680](https://arxiv.org/abs/2005.02680) (2020)
25. Zhu, X., Runcong, M.A., Sun, L., Chen, H.: Word semantic similarity computation based on HowNet and CiLin. J. Chin. Inf. Process. (2016)



ProPC: A Dataset for In-Domain and Cross-Domain Proposition Classification Tasks

Mengyang Hu¹, Pengyuan Liu^{1,2(✉)}, Lin Bo¹, Yuting Mao¹, Ke Xu¹, and Wentao Su¹

¹ School of Information Science Beijing Language and Culture University, Beijing 100083, China

liupengyuan@pku.edu.cn, {202021198379,202121198417,201911680719, 201911680316,201911580771}@stu.blcu.edu.cn

² Chinese National Language Monitoring and Research Center Print Media, Beijing 100083, China

Abstract. Correctly identifying the types of propositions helps to understand the logical relationship between sentences, and is of great significance to natural language understanding, reasoning and generation. However, in previous studies: 1) Only explicit propositions are concerned, while most propositions in texts are implicit; 2) Only detect whether it is a proposition, but it is more meaningful to identify which proposition type it belongs to; 3) Only in the encyclopedia domain, whereas propositions exist widely in various domains. We present ProPC, a dataset for in-domain and cross-domain propositions classification. It consists of 15,000 sentences, 4 different classifications, in 5 different domains. We define two new tasks: 1) In-domain proposition classification, which is to identify the proposition type of a given sentence (not limited to explicit proposition); 2) Cross-domain proposition classification, which takes encyclopedia as the source domain and the other 4 domains as the target domain. We use the Matching, Bert and RoBERTa as our baseline methods and run experiments on each task. The result shows that machine indeed can learn the characteristics of various types of propositions from explicit propositions and classify implicit propositions, but the ability of domain generalization still needs to be strengthened. Our dataset, ProPC, is publicly available at <https://github.com/NLUSoCo/ProPC>.

Keywords: proposition in NLP · In-domain classification · Cross-domain classification

1 Introduction

Propositions are defined as the meaning of declarative sentences in linguistics and logic, which can be identified true or false. The “meaning” here is understood as a non-linguistic entity shared by all sentences with the same meaning [17]. In this paper, we directly use propositions to refer to the statement representation

of propositions. Natural language contains a large number of propositions. The ones guided by complete logical connectives are called explicit propositions [1], otherwise we call them implicit propositions. We mainly focus on four types of propositions: Categorical proposition, Conjunctive proposition, Hypothetical proposition and Disjunctive proposition.



Fig. 1. An example involving the four types of propositions in our dataset and their logical relationship between sentences. Here, the first three sentences framed by dashed lines are implicit propositions, the others are explicit propositions. The definition of propositions we will explain in Sect. 2.

When we apply the general definition of propositions to NLP area, we find that about 81% statements can be mapped to propositions and their corresponding types in 100 randomly selected sentences within 5 domains. As is shown in Fig. 1, identifying the types of propositions correctly supports capturing the logical relationship between sentences. Propositions can thus assist natural language understanding, reasoning and generation, and promote the development of related tasks, such as reading comprehension, text inference, and text summarization.

The research on propositions in linguistics and logic has already been very in-depth [2,3,18], but in NLP has not been carefully explored. [1] introduced the concept of propositions into NLP, proposed the task of proposition identification, and constructed a dataset of explicit propositions by using trigger keywords, which is restricted to the encyclopedia field. However, in natural language, propositions exist widely in various domains, and implicit propositions (like the first three sentences in the Fig. 1), exist more widely than explicit propositions. Moreover, identifying the proposition type is more valuable than just detecting whether it is a proposition. In addition, there are problems such as the absence of keywords or the incorrect position of keywords, so the types of propositions cannot be determined only according to the keywords or even the language forms.

In this paper, we introduce a new dataset to further enable in-domain and cross-domain proposition classification tasks. The in-domain proposition classification is to identify the proposition type of a given sentence, which includes implicit propositions. The cross-domain proposition classification takes encyclopedia as the source domain and the other 4 domains as the target domain. For cross-domain data, we select the four popular domains in the current NLP, including medical, law, news, and finance, and extract the data from the answer sets of commonly used question-and-answer datasets in these domains. We built the ProPC dataset through large-scale manual annotation, and conduct a series of experiments to evaluate it.

We use Matching, Bert [6], and RoBERTa [15], as our baseline methods to experiment on each task. The results show that models can classify propositions to some extent, but are still lacking in the overall logic of sentences, and the ability to domain generalization also need to be strengthened.

The main contributions of this paper can be summarized as follows: 1) We redefine the proposition in NLP and propose two new tasks: in-domain and cross-domain proposition classification; 2) We present a dataset, ProPC, for the two tasks, which contains 15,000 manually annotated statements; 3) We use several baselines to conduct experiments and analyze the results, and we find that although AI did not do well in cross-domain tasks, it can indeed classify propositions to a certain extent.

2 Dataset Construction

To explore the in-domain and cross-domain propositions classification, we introduce a new dataset (ProPC) consisting of declarative sentences in various domains manually annotated for their classification. The overall construction process of the data set is shown in the Fig. 2 (a). In this section we first discuss the redefinition of the proposition in NLP, and subsequently present the data acquisition and annotation process as well as statistics of the dataset.

2.1 Proposition Definition

In linguistics and logic, propositions are defined as the meaning of declarative sentences, and the “meaning” here is not related to language forms [17]. Different propositions contain different logical relationships, which can be expressed with some keywords. When the concept of proposition [1] is put forward into NLP, these keywords are regarded as the features and the basis to classify propositions. However, the basis of proposition classification should be based on semantic logic rather than language form, so we redefine the proposition in NLP as follows:

Categorical Proposition: Make a direct and unconditional judgment on whether an object logically contains a certain property and whether it belongs to a certain category.

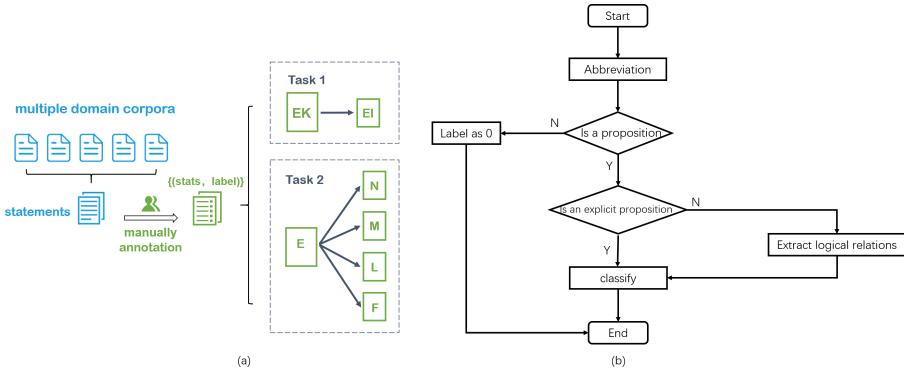


Fig. 2. Dataset construction and annotation workflow overview.

Conjunctive Proposition: A kind of compound proposition, which reflects the simultaneous existence of several situations or properties of objects, and logically has a conjunctive relationship.

Hypothetical Proposition: A kind of compound proposition, which contains a previous or tentative explanation, and logically has a conditional relationship.

Disjunctive Proposition: A kind of compound proposition, which reflects the fact that at least one existence of either conditions or properties of objects, and logically has a disjunctive relationship.

Explicit Proposition: A proposition guided by logical keywords¹, and the keywords are complete and at the right positions.

Implicit Proposition: A proposition that is not an explicit proposition, which means that a statement guided by logical keywords, once the keywords at wrong positions or partly missing, it may also be an implicit proposition².

2.2 Data Acquisition

Explicit propositions with logical constants are easier to judge the type than those implicit propositions who don't have enough guiding keywords. [1] not only introduce the concept of proposition into NLP, but also construct a dataset based on logical keywords. Therefore, we use this dataset as our source of explicit propositions and randomly crawled sentences containing implicit propositions from the same domain, Baidu Encyclopedia. Among them, explicit propositions are directly derived from the above data, while the homologous data are non-repeated data extracted randomly from Baidu Encyclopedia. For cross-domain data, we select four popular domains in the current NLP, including medical, law, news and finance. The data in the news domain is obtained from the official website of the People's Daily. And for other domains, question-and-answer

¹ Logical keywords, like “all...are...”, “both...and...”, “if...then...”, “either...or...”, etc.

² for example, “if you don't fight, you fail”, here the logical keywords should be “if...then”, it lose a “then”, so it is an implicit proposition.

datasets are more common and the sentences in them are relatively daily and standardized (few tone words and few unsmooth sentences). We obtained the data from the answer sets of commonly used question-and-answer datasets in these domains, which is webMedQA [4], lawzhidao [19] and financezhidao [20].

After cleaning and segmenting the obtained text, the collected set consist of 15,000 statements, in which 10,000 for encyclopedia domain data only with logical keywords, 1,000 for encyclopedia domain data with implicit propositions, and 1,000 data for each of the other 4 fields.

2.3 Data Annotation

Our annotation is divided into three parts: annotation training, trial annotation and formal annotation. Before organizing the annotation, we analyzed and labeled some extracted statements, and completed the annotation standard for proposition classification based on natural language processing (not looking at keywords but looking at the sentence logic), and identified 120 statements as the benchmark for annotation training and trial annotation.

The annotation was conducted by 8 undergraduate and graduate students. They first received the annotation training, which provided the project background introduction and annotation specification explanation, and carried out the annotation demonstration of 20 sentences to further explain the specific process of annotation.

Trial Annotation. In the trial annotation part, 8 annotator labelled the 100 encyclopedia statements (contains both explicit and implicit propositions) with the aboveIdentified labels to ensure that the annotator understand the annotation specifications. The consistency test results show that there was good consistency among the 8 annotator (Fleiss Kappa [5] = 0.7278), and the accuracy of each annotator reached more than 80% (compared with the identified labels). We also conducted a pairwise cross-validation(contains each type and each domain of our dataset). The statements labeled by each annotator has duplicate annotation statements with each other annotators (the number of repetitions in each part is equal), and the agreement rate between each two annotator is also greater than 80%. All these indicate that the trained annotator has understood the annotation specification and can proceed to the next step, formal annotation.

Formal Annotation. The formal annotation process is divided into three stages: only the encyclopedia statements with logical keywords, the encyclopedia statements that conform to the natural distribution (containing implicit propositions), and the other four domains that conform to the natural distribution. Figure 2 (b) shows the overall annotation process. While labeling the data, we also conduct experiments at the same time. We find that the accuracy of the model did not increase much after the amount of data reached 5,700. Therefore, in order to reduce manual consumption and improve annotation efficiency we only take 10,000 data for annotation in the first stage. In the remaining stages,

we have also double-checked the labeling results of each annotator every two days to ensure the quality of the dataset.

The resulting set of annotations consists of 10,000 encyclopedia statements with logical keywords(not all explicit propositions), 1,000 encyclopedia statements contains implicit propositions(conforms to the original distribution of various propositions in natural language), 1,000 news statements, 1,000 medical statements, 1,000 law statements and 1,000 finance statements, for a total of 15,000 statements.

2.4 Dataset Analysis

There are a total of 5 distinct domains including medical, law, news, finance and encyclopedias, and a total of 4 distinct classifications in the ProPC dataset. The entire dataset contains 15,000 sentences, the encyclopedia data guided by logical keywords contains 10,000 sentences, and the remaining 5 parts each contain 1,000 sentences.

Table 1. The proportion of explicit and implicit propositions in ProPC dataset propositions. Among them, the disjunctive proposition has no implicit form [9].

	EK	EI	News	Medical	Law	Finance
Category	1368	168	264	262	322	52
Conjunctive	3920	471	598	431	337	392
Hypothetical	2023	102	28	58	145	163
Disjunctive	235	15	5	7	6	3
Not	2454	244	105	242	190	390
Total	1,0000	1,000	1,000	1,000	1,000	1,000

In this paper, we use “EK” denotes encyclopedia data with logical keywords (not all explicit propositions), “EI” denotes encyclopedia data contains implicit propositions (conforms to the original distribution of various propositions in natural language), “N” denotes news data, “M” denotes medical data, “L” denotes law data and “F” denotes finance data, “Not” denotes a sentence which is not a proposition. Table 1 shows the overall distribution of propositions in ProPC, which indicates that propositions exist widely in natural language and occupy a considerable proportion.

Figure 3 shows the percentage of implicit propositions in different domains and types, we can observed that statements guided by logical keywords also can be implicit propositions, and this kind of situation occurs most often on the conjunctive propositions. Besides, implicit propositions exist more widely in natural language no matter what kind of domains or types, and almost all category propositions tend to be implicit propositions.

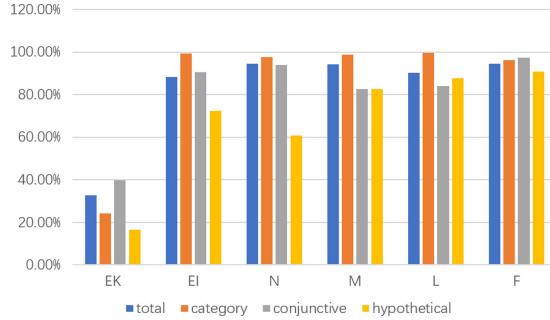


Fig. 3. The percentage of implicit propositions in the propositions of various domains and types.

Table 2 shows the average sentence length of each domain and type. The average length of sentence is 40.048. In respect of type, the average length of sentence of category propositions is the smallest of all. In respect of domain, the average length of sentence of finance is the maximal of all.

Table 2. Average sentence length in each domain of ProPC dataset.

	EK	EI	News	Medical	Law	Finance	Total
Category	22.676	44.393	29.352	20.592	22.981	46.135	25.203
Conjunctive	44.171	57.098	55.299	33.044	31.154	59.013	45.687
Hypothetical	27.903	32.480	34.964	30.520	27.469	50.607	29.717
Disjunctive	40.443	48.530	41.600	22.714	32.833	35.667	40.085
Not	45.498	58.963	34.000	30.529	28.200	77.679	47.599
Total	38.168	52.779	48.511	28.659	27.437	64.183	40.048

Table 3 shows the standard train, validation and test folds of the ProPC dataset. For EK, the dataset is divided according to the ratio of training: verification: testing = 8:1:1, and for the rest, we directly use them as the test sets.

3 Experiments

3.1 Baseline Methods

Matching. This baseline simply matches statements based on the logical keywords corresponding to the different classes, which establishes the corresponding template for each class by regular expression and then matches the statements one by one. We believe this method may indicates the effect bias of the machine on nature distributed statements when only explicit propositions are studied.

Table 3. Statistics of train, validation and test folds of ProPC dataset. Note: ‘EK’ denotes encyclopedia data with logical keywords, ‘EI’ denotes encyclopedia data with implicit propositions, ‘N’ denotes news data, ‘M’ denotes medical data, ‘L’ denotes law data and ‘F’ denotes finance data.

	Train (EK)	Val (EK)	Test (EK)	Test (EI)	Test (N)	Test (M)	Test (L)	Test (F)
Category	1086	133	149	168	264	262	322	52
Conjunctive	3118	400	402	471	598	431	337	392
Hypothetical	1622	211	190	102	28	58	145	163
Disjunctive	192	22	21	15	5	7	6	3
Not	1982	234	238	244	105	242	190	390
Total	8000	1,000	1,000	1,000	1,000	1,000	1,000	1,000

Bert. Both in-domain and cross-domain proposition classifications can be regarded as a multi-class problem. Bert [6] is one of the most popular models for text classification. For text classification tasks, Bert uses the final hidden state h of the first token [CLS] as the representation of the entire sequence, and add a simple softmax classifier to the upper layer to predict the probability of label c :

$$p(c|h) = \text{softmax}(Wh) \quad (1)$$

where W is the task-specific parameter matrix. We fine-tune all the parameters from BERT as well as W jointly by maximizing the log-probability of the correct label, and use the fine-tuned Bert [7] as our baseline to test these two tasks.

RoBERTa. RoBERTa [15] has a similar backbone architecture to Bert, is another popular SOTA model after Bert. It differs from Bert mainly in the following aspects. First, a larger batch was used to conduct more in-depth training on larger datasets; Second, no longer use NSP(Next Sentence Prediction) task; Third, dynamically change the MASK mode of training data [16]. In addition, RoBERTa has made remarkable achievements in many tasks and datasets. So we also used the fine-tuned RoBERTa as one of our baseline models.

3.2 Experimental Setup

We use the BERT-base-Chinese model [8] and RoBERTa-wwm-ext model [16], as our base models, fine-tune them for 10 epochs over all the reviews data, and save the best model on the validation set for testing. Hyper-parameter details for each baseline can be found at <https://github.com/NLUSoCo/ProPC>. During our experiments, we use F1-score as the main evaluation metric, weighted across all the classes.

3.3 Results and Analysis

Task 1: In-Domain Proposition Classification. We conduct three sets of experiments in encyclopedia domain to explore the proposition classification of

the mode (Table 4). For Bert and RoBERTa, We first train the model on EK, and then use the trained model to test EK and EI data. Table 4 shows the results. We observe that: The Matching method’s results are not good in each set of experimental data, which indicates that it is insufficient to focus only on logic keywords to classify statements, and it is of certain practical significance to construct implicit data sets that conform to natural distribution. Bert and RoBERTa can indeed learn the features of the different proposition types, and the features it learned from explicit propositions can be generalized to implicit proposition classification. The features of category and conjunctive propositions are more noticed by the model, whereas the model has a poor ability to judge disjunctive propositions. This may be due to the relatively small proportion of disjunctive propositions in the dataset, resulting in insufficient training.

Table 4. The performance of the models on in-domain tasks. Note: W-F1: weighted average F1-score, EK: encyclopedia data with logical keywords, EI: encyclopedia data with implicit proposition, EW: EK without logic keywords. cat: category proposition, Con: conjunctive proposition, Hyp: Hypothetical proposition, Dis: disjunctive proposition.

		Cat	Con	Hyp	Dis	Not	W-F1
EK	Matching	0.6042	0.2913	0.2959	0.1184	0.103	0.3741
	Bert	0.8591	0.8550	0.8350	0.5833	0.5143	0.7970
	RoBERTa	0.8125	0.8557	0.8302	0.4826	0.5723	0.7502
EI	Matching	0.3761	0.4210	0.3294	0.1471	0.1080	0.3182
	Bert	0.6617	0.8100	0.5882	N/A	0.6860	0.7438
	RoBERTa	0.6541	0.8492	0.5962	0.1250	0.6904	0.7609

Task 2: Cross-Domain Proposition Classification. For the cross-domain part, we treat the encyclopedia as the source domain and news, medical, law and finance as the target domain. Our modal is tuned on source domain and tested on the target domain. As is shown in Table 5, we conduct four experiments on Task 2, and obtain the following observations: The overall results of cross-domain tasks are not as good as that of in-domain tasks. The ability of domain generalization needs to be strengthened, while the attention of model to each classification is similar to Task 1. Among them, the closest domain to encyclopedia is news. This may be caused by the data source, because the source of the news corpus is the People’s Daily, and the language is more standardized and close to the encyclopedia, while the other three corpora are extracted from the QA datasets, which means the language is more colloquial.

4 Related Work

Proposition analysis has been among active areas of research in both linguistics and logic. [10] define the concept of linguistic logic, expounds the status of

Table 5. The performance of the models on cross-domain tasks. Note: W-F1: weighted average F1-score, EK: encyclopedia data with logical keywords, N: news data, M: medical data, L: Law data, F: finance data. cat: category proposition, Con: conjunctive proposition, Hyp: Hypothetical proposition, Dis: disjunctive proposition.

		Cat	Con	Hyp	Dis	Not	W-F1
N	Matching	0.5121	0.6590	0.0710	N/A	0.0953	0.5331
	Bert	0.6937	0.7976	0.5079	0.2857	0.3529	0.7128
	RoBERTa	0.6643	0.8342	0.6000	0.2500	0.2816	0.7166
M	Matching	0.4544	0.2700	0.1900	0.0451	0.1230	0.2760
	Bert	0.6024	0.6357	0.3699	0.2000	0.4611	0.5663
	RoBERTa	0.6936	0.7198	0.5182	0.1311	0.3745	0.6084
L	Matching	0.4553	0.3081	0.3790	0.0681	0.0384	0.3131
	Bert	0.5607	0.5524	0.5665	0.2857	0.4224	0.5308
	RoBERTa	0.6600	0.6032	0.6439	0.1905	0.4133	0.5863
F	Matching	0.1333	0.4700	0.4562	0.0222	0.1170	0.3112
	Bert	0.4000	0.5626	0.5926	N/A	0.6601	0.5954
	RoBERTa	0.4922	0.6350	0.6222	0.1436	0.6278	0.6188

linguistic logic, comb the methods of studying linguistic logic, and explain the significance of studying linguistic logic. [11] distinguish orthodox logic from natural language logic. [12] study from semantics, grammar, pragmatics, rhetoric and other fields, deeply explored the relationship between logic and linguistics, and affirmed the importance of logic for language understanding.

However, The research on proposition in NLP is still less. [1] is directly related to this area about explicit proposition. They build an explicit proposition corpus and proposes two tasks: the automatic explicit proposition recognition and the essential explicit proposition ingredients analysis. [13] proposes an analogical reasoning task on Chinese, build dataset CA8 for this task, and explore the influences of vector representations, context features, and corpora on analogical reasoning. [14] build a dataset based on news corpus, and explore the task of recognizing the relationship between sentences in Chinese text. According to whether there are textual connectives between text units, they divided the relationship into explicit textual relationship and implicit textual relationship.

5 Conclusion

In this paper, we present two tasks, in-domain and cross-domain proposition classification. The in-domain proposition classification is to identify the proposition type of a given statement include both explicit and implicit. The cross-domain proposition classification takes encyclopedia as the source domain and each of the other 4 domains as the target domain. To enable research on this 2 tasks, we introduce a novel dataset, ProPC, consist of explicit and implicit proposition

drawn from different sources. We use the Matching, Bert and RoBERTa methods as our baselines to run experiments on each of the tasks. Results of our experiments indicates that machine can identify proposition types to a certain extent, but still lacks attention to the logic level of sentences, and its domain generalization ability needs to be further strengthened. In the future, we will continue to expand the size of the dataset, optimize the model, and explore more methods for domain generalization.

Acknowledgements. Support by Beijing Natural Science Foundation (4192057) and Science Foundation of Beijing Language and Culture University (the Fundamental Research Funds for the Central Universities: 21YJ040005).

References

1. Liu, L., et al.: Automatic recognition and analysis of explicit propositions in natural language. *J. Chin. Inf. Process.* **35**(2), 41–51 (2021)
2. Tomasello, M.: Cognitive linguistics. In: *A Companion to Cognitive Science*, pp. 477–487(2017)
3. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: an annotated corpus of semantic roles. *Comput. Linguistics* **31**(1), 71–106 (2005)
4. He, J., Fu, M., Tu, M.: Applying deep matching networks to Chinese medical question answering: a study and a dataset. *BMC Med. Inf. Decis. Making* **19**(2), 91–100 (2019)
5. Fleiss, J.L.: Measuring nominal scale agreement among many raters. *Psychol. Bull.* **76**(5), 378 (1971)
6. Devlin, J., et al.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) (2018)
7. Sun, C., Qiu, X., Xu, Y., Huang, X.: How to fine-tune BERT for text classification? In: Sun, M., Huang, X., Ji, H., Liu, Z., Liu, Y. (eds.) *CCL 2019. LNCS (LNAI)*, vol. 11856, pp. 194–206. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32381-3_16
8. Cui, Y., et al.: Pre-training with whole word masking for Chinese BERT. arXiv preprint [arXiv:1906.08101](https://arxiv.org/abs/1906.08101) (2019)
9. Huang, S.: On the hidden form of logical constant. *J. Jianghan University (Soc. Sci. Ed.)* **4** (1991)
10. Li, X., et al.: Language, logic and logic of language. *Philos. Stud.*, 41–48 (1986)
11. Zhou, L.: Formal logic and natural language. *Philos. Stud.*, 29–35 (1993)
12. Gao, F.: on the role of formal logic in language research. *Mod. Chinese (Lang. Res. Ed.)*, 4–6 (2017)
13. Li, S., et al.: Analogical reasoning on Chinese morphological and semantic relations. arXiv preprint [arXiv:1805.06504](https://arxiv.org/abs/1805.06504) (2018)
14. Zhang, M., Song, Y., Qin, B., Liu, T.: Semantic relation recognition of Chinese text level sentences. *Acta Sinica Sinica* **27**(06), 51–57 (2013)
15. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. arXiv preprint [arXiv:1907.11692](https://arxiv.org/abs/1907.11692) (2019)
16. Xu, Z.: RoBERTa-wwm-ext Fine-Tuning for Chinese Text Classification. arXiv preprint [arXiv:2103.00492](https://arxiv.org/abs/2103.00492) (2021)
17. McGrath, M., Frank, D.: *The Stanford Encyclopedia of Philosophy*. 2nd edn. Metaphysics Research Lab, Stanford University (2020)

18. Allwood, J., et al.: Logic in Linguistics. Cambridge University Press (1977)
19. ChineseNlpCorpus. <https://github.com/SophonPlus/ChineseNlpCorpus/tree/master/datasets/lawzhidao>. Accessed 17 Jan 2019
20. ChineseNlpCorpus. <https://github.com/SophonPlus/ChineseNlpCorpus/tree/master/datasets/financezhidao>. Accessed 17 Jan 2019



CTRD: A Chinese Theme-Rheme Discourse Dataset

Biao Fu^{1,2}, Yiqi Tong^{1,2,3}, Dawei Tian⁴, Yidong Chen^{1,2(✉)}, Xiaodong Shi^{1,2}, and Ming Zhu⁵

¹ Department of Artificial Intelligence, School of Informatics,
Xiamen University, Xiamen, China

{biaofu,yqtong}@stu.xmu.edu.cn, {ydchen,mandel}@xmu.edu.cn
² Key Laboratory of Digital Protection and Intelligent Processing of Intangible Cultural Heritage of Fujian and Taiwan, Ministry of Culture and Tourism, Xiamen, China

³ Institute of Artificial Intelligence, Beihang University, Beijing, China

⁴ Yangzhou Institute of Technology, Yangzhou, China
tdw2011428@163.com

⁵ Xiamen Hospital of Traditional Chinese Medicine, Xiamen, China
syumei_amoy0592@163.com

Abstract. Discourse topic structure is the key to the cohesion of the discourse and reflects the essence of the text. Current Chinese discourse corpus are constructed mainly based on rhetoric and semantic relations, which ignore the functional information in discourse. To alleviate this problem, we introduce a new Chinese discourse analysis dataset called **CTRД**, which stands for **C**hinese **T**heme-**R**heme **D**iscourse dataset. Different from previous discourse banks, CTRД was annotated according to a novel discourse annotation scheme based on the Chinese theme-rheme theory and thematic progression patterns from Halliday’s systemic functional grammar. As a result, we manually annotated 525 news documents from OntoNotes 4.0 with a Kappa value greater than 0.6. And preliminary experiments on this corpus verify the computability of CTRД. Finally, we make CTRД available at <https://github.com/ydc/ctrд>.

Keywords: CTRД · Discourse analysis · Theme-Rheme theory

1 Introduction

Discourse is a kind of text analysis granularity beyond words and sentences [29], which plays a crucial role in natural language processing (NLP). However, many NLP applications like Neural Machine Translation (NMT) have not fully utilized contextual information [15, 31] and lead to negative results.

Figure 1 shows an error made by the state-of-the-art NMT system [27]. Where the arrow represents the progression between clauses and the second line is the translation result by the Google NMT system and the third line is the golden result. We can find that NMT cannot deal with the omission of subject well, which the subject “two cakes”

B. Fu and Y. Tong—Equal contribution.

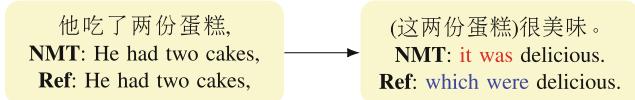


Fig. 1. A normal Chinese example.



Fig. 2. A simple linear thematic progression example of systemic functional grammar.

is omitted in parentheses. Actually, this problem can be solved by introducing theme-rheme theory and thematic progression patterns from Halliday's systemic functional grammar [7]. As shown in Fig. 2, according to the above two theories, the progression pattern of the whole sentence is a simple linear progression. Specifically, the theme of the second clause is the rheme of the previous clause. So, the plural ("were") should be used in the second clause.

In the past few years, the advent of large-scale collections of annotated data has demonstrated a paradigm shift in the research community for NLP [4, 8, 19, 20, 32]. While most of the existing works on Chinese discourse corpora construction are based on the framework of Rhetorical Structure Theory (RST) [14, 23] and Penn Discourse TreeBank (PDTB) [16, 18], a few of them consider introducing functional grammar information. On the other hand, recent works [22, 25] have demonstrated the significance of introducing theme-rheme information in document-level abstracting and NMT. Hence, we present a Chinese Theme-Rheme Discourse dataset called CTRD, which based on theme-rheme theory and thematic progression patterns to provide a new way for Chinese discourse analysis. Some examples of CTRD are shown in Fig. 3.

In summary, we make the following contributions in this paper:

- We built a Chinese theme-rheme annotation scheme based on theme-rheme theory and thematic progression patterns. And we have developed a collaborative annotation tool to simplify the annotation work based on the annotation scheme.
- We manually annotated a Chinese Theme-Rheme Discourse Dataset (CTRD). To the best of our knowledge, CTRD is the first dataset based on theme-rheme theory and thematic progression patterns.
- We implemented a baseline model for automatic recognition of theme-rheme information. In the ten-fold cross validation experiments of theme-rheme function type recognition, our model achieves 79.77 F1-score on the test set, which demonstrates the effectiveness of our annotation scheme.

```

<DOC>
  <S ID="1">
    <T WORD="中国 民生 银行 总资产" REF="Total assets of China Minsheng Bank"
        FUNCTION="TOP" CONNTOS="0" NUM="4"/>
    <R WORD="已 突破 二百亿 元" REF="have exceed 20 billion yuan" TYPE="NOR"
        LINKTOS1="1" LINKTO1="T" LINKTOT1="TOP" NUM="4"/>
  </S>
  ...
  <S ID="16">
    <T WORD="中国 民生 银
      行" REF="China Minsheng Bank" FUNCTION="TOP" CONNTOS="0" NUM="3"/>
  </S>
  ...
  <S ID="18">
    <R WORD="在 海内外 初步 树立 了 良好 的 社会 形
      象" REF="Initially set up a good social image at home and abroad" TYPE="NOR"
        LINKTOS1="16" LINKTO1="T" LINKTOT1="TOP" NUM="9"/>
  </S>
</DOC>

```

Fig. 3. Some examples from CTRД (simplified version).

2 Related Work

There are numerous theories that attempt to describe various discourse features from different perspectives. In these theories, the RST based on the rhetorical relations was widely accepted and applied. [2] annotated the RST Discourse Treebank (RST-DT) based on RST. For Chinese, [17] released the Chinese news commentaries dataset and [8] annotated Macro Chinese Discourse TreeBank (MCDTB). On the other hand, inspired by discourse-level extension of lexicalized tree-adjoining grammar (D-LTAG) [6], [18] annotated the PDTB, one of the most popular treebanks, and launched a series of studies on it. PDTB emphasizes the role of connectives in rhetorical relations. [34] annotated a Chinese discourse corpus called Chinese Discourse TreeBank (CDTB), which including 500 documents.

Furthermore, the study based on entity relations has attracted many attentions from researchers. Based on the OntoNotes Coreference Annotation Scheme (OCAS) [26], OntoNotes dataset was released, which is a large discourse corpus including various genres of text in three languages (English, Chinese, Arabic). [13] transferred the theories from English to Chinese by integrated RST-DT and PDTB and proposed a Connective-Driven Dependency Tree scheme. However, this method only considered the discourse relationship within the paragraph and did not annotate the discourse relationship between paragraphs and the macro discourse information of the whole text [8]. [28] annotated a Chinese Discourse Topic Corpus (CDTC) to represent discourse topic structure according to the theme-rheme theory, which the theme is the center of the topic and the rheme is a series of descriptions of the theme.

Overall, the number and the scale of Chinese discourse corpus are relatively small at present. Moreover, the existing discourse corpus still lack theme-rheme information [28]. So it is necessary to construct a theme-rheme discourse corpus to promote the development of discourse analysis.

Table 1. Some examples of theme-rheme from three consecutive sentences. The themes are usually entities, such as organization (T1), location (T2) or name (T3), and the rhemes (R1-R3) are the rest part of sentences expect themes.

Examples	Theme	Rheme
Sentence1	The conference(T1)	was held in Vancouver. (R1)
Sentence2	Vancouver(T2)	is an important port city in southwestern Canada. (R2)
Sentence3	Howard(T3)	was invited to attend the meeting. (R3)

Table 2. The classification of theme and rheme in our annotation scheme.

Theme/Rheme	Categories
Theme	Topical theme (TOP)
	Interpersonal theme (INT)
	Textual theme (TXT)
Rheme	Normal rheme (NOR)
	Independent rheme (OMN)

3 Theory Basis

The annotation processes of CTRD follows two guiding theories: the theme-rheme theory and the thematic progression patterns from Halliday's systemic functional grammar.

3.1 The Theme-Rheme Theory

The theme defined by Halliday is the element which serves as the point of departure of the message, and it represents the known or shared information from the immediate context [7]. The rheme is the remaining information in the clause except the theme. And it is not only a further supplement or interpretation of theme, but also the central part of information [7]. In other words, we can consider that theme and rheme are known information and new information respectively. So the theme provides the settings for the rheme. As shown in Table 1, the boundary between theme and rheme is clear, a theme is usually an entity word that represents the topic or subject of a sentence, while a rheme always provides information for theme.

Moreover, Halliday subdivided theme into single theme and multiple theme. Single theme is a whole independent unit which cannot be split into smaller units. While multiple theme has internal structures that can be further divided into topical theme, interpersonal theme and textual theme. Thus, as shown in Table 2, there are three kinds of theme and two kinds of rheme in our annotation scheme. Topical theme is the first experiential element of a clause whether it is a participant, circumstance or process. Interpersonal theme is personal judgement on meaning of the speaker or writer. Textual theme relates the meaning of the clause to the other parts of the text. For rheme, there are two main forms, which are normal rheme and independent rheme. If a sentence has no theme and the whole sentence consists of a single rheme, it is an independent rheme, and other cases are normal rheme.

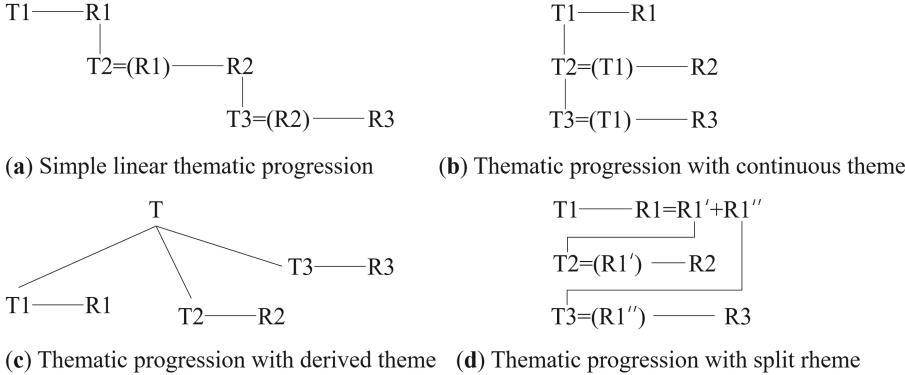


Fig. 4. Four kinds of thematic progression pattern in our annotation scheme. Here are four examples of patterns. (a): He (T1) bought an iPhone (R1). It (T2) was designed by Apple Inc (R2). Apple (T3) was founded by Steve Jobs (R3). (b): My brother (T1) lives America (R1). He (T2) is an engineer (R2). He (T3) works for an Internet company (R3). (c): His father (T1) is a lawyer (R1). His mother (T2) is a teacher (R2). His sister (T3) is an accountant (R3). (d): He (T1) has two life goals (R1). One (T2) is to travel to 50 countries (R2). The other (T3) is to live to be 100 years old (R3).

3.2 The Thematic Progression Patterns

Most of the discourse are composed of two or more sentences in discourse analysis. In these case, there will be some connections and changes between theme and theme, rheme and rheme, and theme and rheme in different sentences. These connections and changes are called progression.

Previous works summed up four basic patterns of thematic progression [1, 9, 21]. As shown in Fig. 4, simple linear progression notes an item in the rheme of the current clause becomes the theme of the next clause. Thematic progression with a continuous (constant) theme means that the item in the theme of the first clause is also selected as the theme of the following clause. In the patterns of thematic progression with derived themes, themes are derived from a hyper-theme. In the pattern of thematic progression with a split rheme, a rheme may include several different information, each of information may take up as the theme in several subsequent clauses.

These patterns are manifested through the connection between the themes and rhemes in different sentences. Therefore, when we constructed the corpus, we also stipulated the relationships between the themes and rhemes in the corpus.

4 Annotation Scheme

To meet the professional linguistic demand when annotating CTRД, we employed six masters and two senior doctors who majored in linguistic. To simplify the process of annotation, we designed and implemented a computer aided tool. To ensure annotation quality, the whole annotation process has two main phases:

Initialization Annotation Phase. In this phase, the main goal is to teach masters how to annotate and ensure the quality. Specifically, we selected a small extra dataset and

Table 3. Two examples of our theme-rheme annotation.

Example 1		
Content	Theme/Rheme	Type
但(but)	Theme	Textual
希望(hope)	Theme	Interpersonal
他们(they)	Theme	Topical
能够配合司法调查 (could cooperate with judicial investigation)	Rheme	Normal

Example 2		
Content	Theme/Rheme	Type
专门对女队员进行体能训练 (special physical training for female team members)	Rheme	Independent

labeled by masters, then the doctors will check the accuracy of each student's annotation. Only if someone reaches a certain accuracy, such as 95%, then the student will be allowed to participate in the next phase.

Formal Annotation Phase. Each of the two graduate students formed a team, and each team annotated 175 articles respectively. Then, the doctor checked and corrected the annotated documents to ensure the consistency of annotations. The ambiguous label was determined by the doctor. In this way, a total of 525 news articles were annotated.

4.1 Theme-Rheme Annotation Criteria

The theme was divided into three components, including topical component, interpersonal component and textual component. A theme unit may contain more than one component, but its order has a certain regularity. The topical theme (TOP) mainly discuss the experiential meaning of things, which is also the common boundary between theme and rheme. The interpersonal theme (INT) is usually informative, modal or interrogative words. Besides, INT can also appear in imperative sentences, which is intended to strengthen the expression and highlight the tone or emotion of the sentences. The textual theme (TXT) is divided into continuation conversation components and conjunction components, which generally appears in the form of adverbials or adverbs and indicating time or place (orientation or position).

The rheme was categorized as normal rheme (NOR) and independent rheme (OMN) based on the characteristics of Chinese. The NOR means that a rheme and a theme appear in a sentence simultaneously. And the most of rhemes were annotated as NOR. On the contrary, the speakers may omit the theme and directly describe their opinion, thus there is no theme in sentence. This type of rheme is called the independent rheme, which usually accounts for a very small proportion in Chinese.

As shown in Table 3 Example 1, “但 (but)”, “希望 (hope)” and “他们 (they)” are three theme units, but they are different components of the theme. “但 (but)” means a change in the meaning of a sentence and assures coherence of the sentence. “希望 (hope)” expresses tone of the speakers and interpersonal information. “他们 (they)” is the center

of speaker discussion, expanding the sentence around this topic. Rheme “能够配合司法 (could cooperate with judicial investigation)” is a concrete behavior description of “他们 (they)”. As shown in Table 3 Example 2, an independent rheme is an entire sentence, this elliptical phenomenon is very common in Chinese.

4.2 Thematic Progression Annotation Criteria

According to the description of theme-rheme theory above, the theme-rheme units can be annotated directly by our auxiliary tools. But the thematic progression patterns are implicit, which cannot be annotated with a single label. Thus, we scale this task to point out the relationship between themes and rhemes in different sentences. To ensure the standardization and consistency of annotations, we mainly follow three principles:

Forward Principle. As thematic progression patterns are generally promoted in a forward-to-backward manner, we stipulate that theme or rheme in current clause can only relate to theme or rheme in the previous clause when characterizing the relationship between sentences.

Neighbor Principle. We argue that when a theme (or rheme) matches more than one associate objects in different sentences, the unit in current clause should be related to the nearest clause. For example, here are three sentences: S1, S2 and S3. We suppose that T1, T2 and T3 are themes of these three sentences respectively and T1, T2 and T3 have relations with each other. Then T3 should be only associated with T2, and T2 should be associated with T1, and T3 can not be associated with T1 directly.

Additional Relationships. We not only consider theme-rheme relationships, but also consider the phenomenon such as anaphora, ellipsis, and coincidence. Relationship annotating involves a binary object, both sides of the relationship must be different components in a clause. There is no component pointing to itself, and object is not empty in a binary object.

5 Statistics

To compare with existing researches and make it easier for others to follow our work, instead of collecting theme-rheme information from unlimited internet web, we selected 525 news articles from OntoNotes4.0 which was consistent with [33]. Moreover, we carefully considered the domains of OntoNotes4.0 and generality of our corpus, so we guaranteed that CTRD covered newswire (News), broadcast news (BN), broadcast conversation (BC), telephone conversation (Tele) and web data (Web). Besides, we did not force the articles to have similar size, so each article contains numbers of sentences varies from dozens to hundreds.

Our annotated CTRD has 45,591 sentences, and the average number of sentences is 86 per document. Each article contains at most 989 sentences and at least 10 sentences. The average length of clause is 7, which indicates that news articles primarily focus on the middle passage.

We counted the number of theme-rheme pairs (TRs) for each document in CTRD. The maximum and minimum number of TRs are 1731 and 16, respectively. We found 56.6% of all documents contain 50 to 250 TRs, while 94.1% of all documents contain 16

Table 4. Clause types statistics on CTRD.

Theme type	Number	Proportion	Rheme type	Number	Proportion
Topical theme	25,389	32.8%	Normal rheme	29,693	38.4%
Interpersonal theme	1,337	1.7%	Independent rheme	6,879	8.9%
Textual theme	14,095	18.2%			
Total themes	40,821	52.7%	Total rhemes	36,572	48.3%

Table 5. Consistency of annotation on CTRD.

Indicators	Theme-Rheme	Function types	Progression patterns
Agreement	91.87%	82.98%	87.46%
Kappa	0.837	0.718	0.753

to 450 TRs. And We also counted the proportion of the theme-rheme function types. The number of topical themes is 25,389, accounting for 62.2%, which is the most among all types of themes. This result indicates the structure of most Chinese sentences are topical theme joined by rheme. This finding was consistent with the viewpoint of linguists [5], which they believed the topical component is an important symbol to distinguish theme from rheme in the same sentence.

As shown in Table 4, CTRD totally contains 40,821 themes and 36,572 rhemes, which are close in number. While the top three types are topical theme, normal rheme and textual theme. They accounted for 89.4% of the total number of TRs. In addition, we analyzed the distribution of the patterns of thematic progression. The CTRD contains 15,938 continuous thematic progression patterns and 259 simple linear progression patterns. Our statistical results show that the proportion of the thematic progression with continuous theme is the most among the four patterns. This phenomenon denoted that most of the Chinese discourse have an explicit central topic in the overall structure. However, the number of the thematic progression pattern is small, accounting for only one-third of all sentence pairs.

To test the reliability of the annotation, We evaluated inter-annotator agreement from three aspects: theme-rheme, function types and thematic progression patterns. To use a most conservative measure, we used the *exact match* criterion to calculate agreement rates. Moreover, we also used the Kappa coefficient [3] as an evaluation of annotation consistency to consider accidental consistency. We finally calculated the average value of agreement rates and Kappa value of the 525 documents to assess our corpus. As shown in Table 5, our corpus has an agreement rate above 80% and Kappa value above 0.7 in theme-rheme, function types and thematic progression patterns. The corpus has a good annotation quality when the Kappa value of annotation of the corpus is greater than 0.6 [11]. And the results of the consistency indicated that the difficulty of annotation is consistent with the number of annotation types. Specifically, the annotation of function types (five types) is greater difficulty than theme-rheme (two types) and thematic progression patterns (four types).

Table 6. Hyper-parameter values of our neural model.

Parameter	Value	Parameter	Value
Embedding size	256	Dropout	0.05
LSTM layer	2	L2 regularization	1e-8
Learning rate	0.015	Learning rate decay	0.05
LSTM hidden	1024	Batch size	8

Table 7. Ten-fold cross validation result of TR automatic recognition.

Model	Set	Theme-F1	Rheme-F1	Average
CRF	Dev	78.69	89.00	83.85
	Test	78.20	88.91	83.56
Ours	Dev	87.04	93.46	90.25
	Test	84.80	92.25	88.53

6 Experiments and Analysis

To give evidence of the computability of CTRD, we conduct a preliminary automatic recognition research on the identification of theme-rheme (TR) and their function type. We treated theme-rheme automatic recognition task as a sequence labeling problem [24]. So, we use BIO tagging scheme and build a BiLSTM-CRF model that achieved by NCRF++[30] to conduct experiments. Table 6 shows the values of hyper-parameters for our neural model.

It should be noted that we also use traditional Conditional Random Fields (CRFs) [12] as baseline model with the parameter C of 1.5, the feature window of 3, and the rest of the parameters were taken default values. Standard precision, recall and F1-score were used as evaluation metrics.

6.1 Theme-Rheme Automatic Recognition

For our annotation scheme, the theme is the beginning of the information and the rheme is the remaining information in the clause except the theme. Therefore, we believe that the research on automatic recognition of theme-rheme can lay a foundation for further research like anaphora resolution [10]. In the ten-fold cross validation experiments of theme-rheme automatic recognition on the CTRD, the performance of the development set and test set are shown in Table 7.

The BiLSTM + CRF model significantly outperforms the baseline model, which improves the average F1-score 6.40 and 4.97 respectively. We noted that theme automatic recognition results of both two models are underperform rheme automatic recognition results. The main reason is that the number of tokens in rheme were greater than theme to cause the imbalance of label proportion. There are twice as many tokens in rheme as in theme. Specifically, the number of tokens in rheme and theme accounted for 65.83% and 34.17% respectively.

Table 8. Ten-fold cross validation result of TR function type automatic recognition.

Model	Set	Precision	Recall	F1-Score
CRF	Dev	74.49	74.15	74.32
	Test	73.28	72.81	73.05
Ours	Dev	82.23	81.95	82.09
	Test	79.90	79.64	79.77

Table 9. Performance stratified by different function types.

Function types	CRF Dev F1	Ours Dev F1
Topical theme	74.04	85.74
Interpersonal theme	62.39	80.48
Textual theme	64.51	77.73
Normal rheme	81.47	86.84
OMN rheme	29.54	39.14
Others	99.61	99.93

6.2 Function Types Automatic Recognition

As we mentioned in Sect. 3, theme can be divided into three categories and rheme can be divided into two categories. Therefore, we conduct further automatic recognition researches on TR function types. In the ten-fold cross validation experiments of function type automatic recognition on the CTRD, the performance of the development set and test set are shown in Table 8.

Our experiments demonstrate that our model significantly outperforms the baseline model, which improves the F1-score by 7.77 and 6.72 respectively. To gain more insight into the performance of our Bi-LSTM + CRF model, as shown in Table 9, we fetch the model which perform best in the ten-fold and report its performance across the different function types. We respectively take each of function type as positive example to compute F1 score. Apparently, our model outperforms the CRF model on each Function type recognition result.

The model performs best on recognition normal rheme and topic theme, which is the commonest linguistic phenomenon in Chinese. The model is challenged more on other function types (i.e., interpersonal theme), because the amount of these labels was insufficient. It pointed out the future direction of our corpus expansion work.

7 Conclusion

In this paper, we propose a novel Chinese theme-rheme annotation scheme with the introduction of theme-rheme theory and thematic progression patterns as a representation for Chinese discourse functional information. Moreover, we annotated the Chinese Theme-Rheme Discourse Dataset (CTRD), which includes 525 news documents from

OntoNotes4.0. Finally, we perform a range of automatic recognition experiments to prove the appropriateness of annotation scheme and the computability of CTRD.

Acknowledgements. The authors would like to thank Hongkang Zhu and three anonymous reviewers for their comments on this paper. This research was supported in part by the National Natural Science Foundation of China under Grant Nos. 62076211, U1908216 and 61573294, and the Outstanding Achievement Late Fund of the State Language Commission of China under Grant WT135-38.

References

1. Alekseyenko, N.V.: A corpus-based study of theme and thematic progression in English and Russian non-translated texts and in Russian translated texts. Ph.D. thesis, Kent State University (2013)
2. Carlson, L., Marcu, D., Okurowski, M.E.: Building a discourse-tagged corpus in the framework of rhetorical structure theory. In: van Kuppevelt, J., Smith, R.W. (eds.) Current and New Directions in Discourse and Dialogue, pp. 85–112. Springer, Dordrecht (2003). https://doi.org/10.1007/978-94-010-0019-2_5
3. Cohen, J.: A coefficient of agreement for nominal scales. *Educ. Psychol. Measur.* **20**(1), 37–46 (1960)
4. Cui, L., Wu, Y., Liu, S., Zhang, Y., Zhou, M.: MuTual: a dataset for multi-turn dialogue reasoning. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 1406–1416 (2020)
5. Fang, Y.: A study of topical theme in Chinese: An SFL perspective. In: Meaning in Context: Implementing Intelligent Applications of Language Studies, pp. 84–114. Continuum, London (2008)
6. Forbes-Riley, K., Webber, B., Joshi, A.: Computing discourse semantics: the predicate-argument semantics of discourse connectives in D-LTAG. *J. Semant.* **23**(1), 55–106 (2006)
7. Halliday, M., Matthiessen, C.M., Matthiessen, C.: An Introduction to Functional Grammar. Routledge (2014)
8. Jiang, F., Xu, S., Chu, X., Li, P., Zhu, Q., Zhou, G.: MCDTB: a macro-level Chinese discourse treebank. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 3493–3504 (2018)
9. Kizil, M., Kushch, E.: Thematic progression and its types in English literary and legislative texts. *Adv. Educ.* **6**(12), 181–187 (2019)
10. Kong, F., Zhou, G.: A tree kernel-based unified framework for Chinese zero anaphora resolution. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 882–891 (2010)
11. Krippendorff, K.: Content Analysis: An Introduction to its Methodology. Seikeigeka Orthopedic Surgery (1980)
12. Lafferty, J.D., McCallum, A., Pereira, F.C.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning, pp. 282–289 (2001)
13. Li, Y., Feng, W., Sun, J., Kong, F., Zhou, G.: Building Chinese discourse corpus with connective-driven dependency tree structure. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2105–2114 (2014)
14. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: toward a functional theory of text organization. *Text* **8**(3), 243–281 (1988)

15. Miculicich, L., Ram, D., Pappas, N., Henderson, J.: Document-level neural machine translation with hierarchical attention networks. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2947–2954 (2018)
16. Miltsakaki, E., Prasad, R., Joshi, A.K., Webber, B.L.: The PENN discourse treebank. In: LREC (2004)
17. Ming, Y.: Rhetorical structure annotation of Chinese news commentaries. *J. Chinese Inf. Process.* **4** (2008)
18. Prasad, R., et al.: The PENN discourse treebank 2.0. In: LREC. Citeseer (2008)
19. Rajpurkar, P., Jia, R., Liang, P.: Know what you don't know: unanswerable questions for squad. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 784–789 (2018)
20. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100,000+ questions for machine comprehension of text. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 2383–2392 (2016)
21. Rutherford, A., Demberg, V., Xue, N.: 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: Volume 1, Long Papers, pp. 281–291 (2017)
22. Suárez, E.D.O., Cesteros, A.M.F.P.: A new approach for extracting the conceptual schema of texts based on the linguistic thematic progression theory. arXiv preprint [arXiv:2010.07440](https://arxiv.org/abs/2010.07440) (2020)
23. Taboada, M., Mann, W.C.: Rhetorical structure theory: looking back and moving ahead. *Discourse Stud.* **8**(3), 423–459 (2006)
24. Tong, Y., Chen, Y., Shi, X.: A multi-task approach for improving biomedical named entity recognition by incorporating multi-granularity information. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp. 4804–4813. Association for Computational Linguistics, August 2021
25. Tong, Y., Zheng, J., Zhu, H., Chen, Y., Shi, X.: A document-level neural machine translation model with dynamic caching guided by Theme-Rheme information. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 4385–4395 (2020)
26. Weischedel, R., et al.: Ontonotes release 4.0. LDC2011T03. Penn.: Linguistic Data Consortium, Philadelphia (2011)
27. Wu, Y., et al.: Google's neural machine translation system: bridging the gap between human and machine translation. arXiv preprint [arXiv:1609.08144](https://arxiv.org/abs/1609.08144) (2016)
28. Xi, X.F., Zhou, G.: Building a Chinese discourse topic corpus with a micro-topic scheme based on Theme-Rheme theory. *Big Data Anal.* **2**(1), 9 (2017)
29. Yan, H., Webster, J.J.: A corpus-based approach to linguistic function. In: Proceedings of the 27th Pacific Asia Conference on Language, Information, and Computation (PACLIC 27), pp. 215–221 (2013)
30. Yang, J., Zhang, Y.: NCRF++: an open-source neural sequence labeling toolkit. In: Proceedings of ACL 2018, System Demonstrations, pp. 74–79 (2018)
31. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1480–1489 (2016)
32. Yao, Y., et al.: DocRED: a large-scale document-level relation extraction dataset. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 764–777 (2019)
33. Zhang, M., Song, Y., Qin, B., Liu, T.: Chinese discourse relation recognition. *J. Chin. Inf. Process.* **27**(6), 51 (2013)
34. Zhou, Y., Xue, N.: The Chinese discourse treebank: a Chinese corpus annotated with discourse relations. *Lang. Resour. Eval.* **49**(2), 397–431 (2015)

Machine Translation and Multilinguality



Learning to Select Relevant Knowledge for Neural Machine Translation

Jian Yang¹, Juncheng Wan³, Shuming Ma², Haoyang Huang²,
Dongdong Zhang², Yong Yu³, Zhoujun Li¹(✉), and Furu Wei²

¹ State Key Lab of Software Development Environment, Beihang University,
Beijing, China

{jiaya, lizj}@buaa.edu.cn

² Microsoft Research Asia, Beijing, China

{shumma, haohua, dozhang, fuwei}@microsoft.com

³ Shanghai Jiao Tong University, Shanghai, China

{junchengwan, yyu}@apex.sjtu.edu.cn

Abstract. Most memory-based methods use encoded retrieved pairs as the translation memory (TM) to provide external guidance, but there still exist some noisy words in the retrieved pairs. In this paper, we propose a simple and effective end-to-end model to select useful sentence words from the encoded memory and incorporate them into the NMT model. Our model uses a novel memory selection mechanism to avoid the noise from similar sentences and provide external guidance simultaneously. To verify the positive influence of selected retrieved words, we evaluate our model on the single-domain dataset namely JRC-Acquis and multi-domain dataset comprised of existing benchmarks including WMT, IWSLT, JRC-Acquis, and OpenSubtitles. Experimental results demonstrate our method can improve the translation quality under different scenarios.

Keywords: Neural machine translation · Selective translation memory

1 Introduction

Neural machine translation (NMT) with encoder-decoder framework yields the state-of-the-art translation performance in recent years [2, 10, 27, 30], especially on large parallel corpora. Compared to phrase-based SMT explicitly manipulating phrases, NMT with the ability of capturing more complex functions has been widely used to build many advanced translation systems, such as syntax-based models [1, 20], context-aware models [22, 32], and multilingual models [9, 36].

However, NMT suffers from *catastrophic forgetting* problem [11], where the model tends to forget the translation of the low-frequency words or phrases and can not handle the translation across different domains. When there exist overlaps between the training corpus and the test set, one solution is the memory-based NMT models [6, 13, 31, 34], which retrieve similar sentence pairs from the

J. Yang and J. Wan—Equal contribution.

S: machinery and equipment . T: Maschinen und Ausrüstung . Source and Target	S-TM: - the purchase / leasepurchase of new <u>machinery and equipment</u> , including , computer software ; T-TM: - Kauf / Leasingkauf neuer <u>Maschinen und Ausrüstung</u> , einschließlich Computersoftware ;
Translation Memory of Source and Target	

Fig. 1. Example of the source ('S') and target ('T') sentence with their similar pair. 'S-TM' and 'T-TM' denote the similar source and target sentence retrieved from the training corpus. The underlined words provide the external knowledge for translation, while other words are irrelevant.

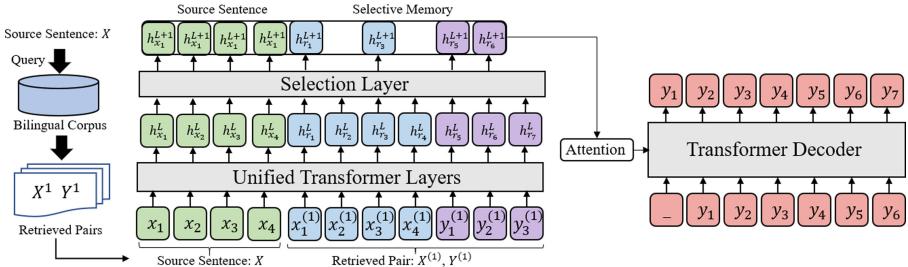


Fig. 2. Overview of our SelectNMT. Given the source sentence X , we retrieve the similar sentence pairs from the bi-lingual corpus and select the top-1 result $\Phi_{X,Y}^1 = \{X^1, Y^1\}$.

training corpus and use them to provide external guidance. But there still exist some noisy words which are irrelevant to the translation and degrade the performance. As shown in Fig. 1, the source piece “machinery and equipment” and target translation “Maschinen und Ausrüstung” guide translation while others are irrelevant and bring extra noise. Therefore, how to select useful retrieved words and avoid extra noise is still a challenging problem (Fig. 2).

In this paper, we propose an end-to-end memory-based NMT model called **SelectNMT**, which uses a novel selection mechanism to distinguish the important words and incorporates them into the translation model. This selection mechanism can select appropriate words from the retrieved sentence pairs as the relevant knowledge to guide translation. This method avoids the disturbance of irrelevant words and doesn't require an additional SMT system. In particular, the multi-task framework can further encourage the model to select useful information from memory. We jointly optimize a binary classification task of the encoder and the translation task of the decoder.

To show the effectiveness of SelectNMT, our model is evaluated on the JRC-Acquis dataset and significantly outperforms the Transformer baseline by an average improvement of +6.79 BLEU points. Then, we construct a multi-domain English-German corpus extracted from existing benchmarks, including JRC-Acquis, OpenSubtitles, IWSLT-14, and WMT-14 bilingual corpora. Our method also gains nearly +2.0 BLEU points on average demonstrating that our method can be extended to the multi-domain scenario.

2 Our Approach

2.1 Problem Definition

Given a source sequence $X = (x_1, \dots, x_m)$ and a target sequence $Y = (y_1, \dots, y_n)$, we retrieve a set of similar sentence pairs $\Phi_{X,Y}^N = \{(X^1, Y^1), \dots, (X^N, Y^N)\}$ using similarity metric, where $\Phi_X^N = \{X^1, X^2, \dots, X^N\}$ are neighbors of the source sentence X , retrieved from the training corpus and $\Phi_Y^N = \{Y^1, Y^2, \dots, Y^N\}$ are translations of the retrieved source sentences. The target translation probability is built as:

$$P(Y|X) = \prod_{i=1}^n P(y_i|X, y_1, \dots, y_{i-1}, \Phi_{X,Y}; \theta) \quad (1)$$

where y_i is the i^{th} word of the target translation. θ are parameters of the NMT model.

2.2 Retrieval Stage

The retrieval stage aims to search similar source sentences and target translations from the training corpus. We leverage the widely-used search engine Lucene¹ to build the index and search similar sentences efficiently. Given a source sentence X , we search its similar source sentences X^1, \dots, X^N and combine target sentences Y^1, \dots, Y^N as source-target pairs $\Phi_{X,Y}^N = \{(X^1, Y^1), \dots, (X^N, Y^N)\}$ in the bilingual corpus. For each input sentence, we offline prepare $M = 20$ searching results as candidates. Then we select top N retrieved sentences as the model input $\Phi_{X,Y}^N$, i.e. source and target memory, where $N \in [1, M]$. This is the same as the previous work [3, 13, 34].

2.3 Machine Translation via Selective Context

Input Concatenation. Our model depends on the concatenation of the source sentence and retrieved sentences. We explore three settings including the source monolingual retrieved sentences, target monolingual retrieved sentences, and bilingual retrieved pairs to perform a systematic study on this setup. The configurations are listed below:

$$[X; \Phi_X^N] \rightarrow Y; [X; \Phi_Y^N] \rightarrow Y; [X; \Phi_{X,Y}^N] \rightarrow Y \quad (2)$$

where “[;]” denotes concatenation operation. The Equations denote three settings of the translation memory: S-TM, T-TM, and Bi-TM, where memory is created by the source retrieved sentences, target retrieved sentences, and bilingual retrieved pairs.

¹ <https://github.com/apache/lucene-solr/tree/master/lucene>.

Segment Embedding. To distinguish the source sentence and retrieved sentences, we introduce the segment embedding. Given the retrieved sentences $\Phi_{X,Y}^N$ and the source sentence X , we project them into the space of word embeddings, position embeddings, and segment embeddings. Finally, we feed the sum of embeddings E into the encoder:

$$E = E_w + E_p + E_s \quad (3)$$

where E_w , E_p , and E_s separately denote the word embedding input, position embedding input, and the segment embedding input.

Encoder with Retrieved Sentences. The encoder can be split into several unified self-attention layers and selection layers. The unified self-attention layers are applied to encode the concatenation of the source sentence and its similar pairs. While the selection layer is used to extract useful information and avoid the noise of context.

Unified Transformer Layers. To incorporate the retrieved sentences $\Phi_{X,Y}^N$ into the NMT model, the input $E_w + E_p + E_s$ is encoded using the Transformer encoder layers:

$$h^L = [h_x^L; h_r^L] = \text{TransformerEnc}(E_w + E_p + E_s) \quad (4)$$

where L is the number of layers. $h_x^L = (h_{x_1}^L, \dots, h_{x_m}^L)$ and $h_r^L = (h_{r_1}^L, \dots, h_{r_t}^L)$ separately denote the representations of the source sentence and the retrieved sentence pairs.

Selection Layer. We use the weighted combinations of the L encoder layers features to predict which retrieved words are selected. We define a vector $W = (w_1, \dots, w_L) \in \mathbb{R}^L$, which is learned during training.

$$h^{L+1} = \text{Self-Attention}\left(\sum_{i=1}^L a_i h^i\right) \quad (5)$$

where a_i is calculated by:

$$a_i = \frac{e^{w_i}}{\sum_{k=1}^L e^{w_k}} \quad (6)$$

The weighted combinations h^{L+1} are used to select which retrieved words are fed into the decoder by the binary classification:

$$\delta = \text{argmax softmax}(W_s h^{L+1}) \quad (7)$$

where $W_s \in \mathbb{R}^{d \times 2}$ and $\delta = (\delta_1, \dots, \delta_t) \in \mathbb{R}^t$, where t is the length of the retrieved sentences. $\delta_j = 1$ denotes the j^{th} word, is selected, while $\delta_j = 0$ denotes the j^{th} word is discarded. After the selection operation, the selected retrieved words hidden states are $h_r^{L+1} = (h_{r_1}^{L+1}, \dots, h_{r_s}^{L+1})$, where s is the length of selected

Table 1. Evaluation results of different trained models on the En \leftrightarrow De, En \leftrightarrow Es, and En \leftrightarrow Fr translation tasks on the JRC-Acquis dataset.

Model	En-De		En-Es		En-Fr		Avg.
	\leftarrow	\rightarrow	\leftarrow	\rightarrow	\leftarrow	\rightarrow	
Transformer [27]	54.95	49.40	59.32	60.82	61.39	62.23	58.02
Transformer + Copy [13]	56.94	53.24	62.50	62.05	64.13	64.33	60.53
TM-Transformer [5]	59.48	55.65	62.82	65.73	66.06	66.88	62.77
Coupled encoder [6]	60.13	54.74	65.56	62.38	67.03	65.92	62.99
CSTM [3]	61.88	56.23	65.84	66.56	66.57	67.52	64.10
SelectNMT (our method)	62.23	56.85	66.67	67.08	67.54	68.48	64.81

words. In this work, we use 6 unified Transformer layers and 1 selection layer for all experiments.

Decoder. Given the representations of the source sentence $h_x^{L+1} = (h_{x_1}^{L+1}, \dots, h_{x_m}^{L+1})$ and selected representations $h_r^{L+1} = (h_{r_1}^L, \dots, h_{r_s}^L)$, the concatenation is fed into the Transformer decoder as below:

$$y_i = \text{TransformerDec}(y_{i-1}, [h_x^{L+1}; h_r^{L+1}]) \quad (8)$$

where y_i is the i^{th} prediction of the target.

2.4 Multi-task Learning Framework

The multi-task learning framework consists of main and auxiliary tasks. Herein, we refer to the machine translation task as the main task and retrieved pieces selection task as the auxiliary task. The overall loss function \mathcal{L} sums the loss of the main machine translation task \mathcal{L}_{MT} and that of the auxiliary selection task \mathcal{L}_{SEL} :

$$\mathcal{L} = \mathcal{L}_{MT} + \lambda \mathcal{L}_{SEL} \quad (9)$$

where λ is a hyper-parameter to control the learning of the selection task.

Machine Translation Task. The main task is the translation task trained on the bilingual dataset \mathcal{D} and retrieved pairs $\Phi_{X,Y}^N$ with the cross-entropy loss:

$$\mathcal{L}_{MT} = \mathbb{E}_{X,Y \in \mathcal{D}} - \log P(Y|X; \Phi_{X,Y}^N) \quad (10)$$

where $\Phi_{X,Y}^N$ denotes the retrieved sentence pairs.

Selection Task. The auxiliary task forces the model to learn to select important retrieved pieces. We use the vector $\delta_r = (\delta_{r_1}, \dots, \delta_{r_t})$ to indicate whether the retrieved words (r_1, \dots, r_t) are selected. $\delta_{r_i} = 1$ denotes the i^{th} retrieved word is selected, while $\delta_{r_i} = 0$ denotes the word is discarded:

$$\mathcal{L}_{SEL} = \mathbb{E}_{X,Y \in \mathcal{D}} - \log P(\delta_r^g | X; \Phi_{X,Y}^N) \quad (11)$$

where the δ_r^g is the ground-truth label. We simply assume that the i^{th} word in the source or target sentences is useful for the translation. Therefore, we set $\delta_{r_i}^g = 1$ if the i^{th} retrieved word in the source or target sentence, else $\delta_{r_i}^g = 0$.

3 Evaluation and Datasets

3.1 Evaluation

All datasets are tokenized with the Moses tokenizer [16] and mixed without any sampling. We train each model for 40 epochs at least, and choose the best checkpoint based on validation performance. BLEU points are computed with tokenized output and references with `multi-bleu.perl`² from Moses.

Table 2. Evaluation results on the En → De multiple domain dataset. The “Avg.” column means the averaged result of the JRC-Acquis, WMT-14, OpenSubtitles, and IWSLT-14 test sets.

En → De	JRC-Acquis	WMT-14	OpenSub	IWSLT-14	Avg.
Transformer [27]	50.18	23.71	24.60	28.64	31.78
Transformer + Copy [13]	54.23	23.26	24.03	28.54	32.51
TM-Transformer [5]	57.15	22.32	24.74	28.21	33.10
Coupled encoder [6]	54.90	22.95	24.30	28.48	32.80
CSTM [3]	56.92	23.94	24.40	29.03	33.57
SelectNMT (our method)	57.84	23.72	24.38	29.11	33.76

Table 3. Evaluation results on De → En multiple test sets. The “Avg.” column means the averaged result of the JRC-Acquis, WMT-14, OpenSubtitles, and IWSLT-14 test sets.

De → En	JRC-Acquis	WMT-14	OpenSub	IWSLT-14	Avg.
Transformer [27]	56.43	26.43	28.66	34.38	36.48
Transformer + Copy [12]	58.20	26.82	29.08	34.07	37.04
TM-Transformer [5]	62.87	25.08	28.37	34.64	37.74
Coupled encoder [6]	60.94	26.18	28.51	34.97	37.65
CSTM [3]	61.24	26.58	28.48	34.80	37.78
SelectNMT (our method)	63.21	26.70	28.70	35.53	38.54

3.2 Datasets

Single Domain. The JRC-Acquis dataset is used to evaluate our method, including En-De, En-Es, and En-Fr language pairs. Following the previous work,³ the same split of the training, valid, and test data are used in our work. All

² <https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl>.

³ <https://github.com/jingyiz/Data-sampled-preprocessed>.

sentence pairs are tokenized by Moses [16] and encoded by BPE [24] with a shared vocabulary of 40K symbols.

Multiple Domain. We construct an En-De heterogeneous dataset from a combination of existing benchmarks, including the WMT-14 training data (1M sentence pairs), the IWSLT-14 bilingual corpus (173K sentence pairs), JRC-Acquis (797K sentence pairs) and OpenSubtitles (1M sentence pairs). For the WMT-14 dataset, we use newstest-2013 for validation and newstest-2014 for test. For the IWSLT-14 benchmark, we use the tst2013 as the valid set and tst2014 as the test set.

3.3 Training Details

We deploy **Transformer_base** [27] for all experiments, which has 8 attention heads, 512 embedding size with a dropout rate of 0.1. We use Adam [15] to train all models, and apply label smoothing with an uncertainty of 0.1. All models are trained on 8 NVIDIA V100 GPUs. We use the same retrieved sentence pairs for all baselines. For the **single-domain dataset**, we separately train models of 6 directions on the bilingual dataset. The batch size is set as 6000 tokens in all directions and the learning rate is set as 0.1. For the **multi-domain dataset**, we accumulate the gradient for 2 iterations and then update model parameters to simulate a 16-GPU environment.

3.4 Baselines

To compare our method with previous baselines, we reimplement the following methods. **Transformer** [27] only encodes the source sentence for translation. **Transformer + Copy** [13] uses the copy mechanism on the Transformer architecture, which can copy words from retrieved target sentences. **TM-Transformer** [5] based on the Transformer architecture augments the source sentence with retrieved pairs through concatenation. **Coupled Encoder** [6] encodes retrieved sentence pairs into NMT with an extra encoder. **CSTM** [3] uses source-target memory to distinguish useful information from noise.

3.5 Results

Single Domain. In Table 1, we present the results of our method and other baselines on 6 translation directions of the JRC-Acquis dataset. It is noticeable that our proposed method significantly outperforms **Transformer** on average, which proves our method sufficiently selects the important information from retrieved sentences. **Transformer + Copy**, which copies words from retrieved target sentence, helps improve performance but not as much as **TM-Transformer**. Compared with **CSTM** employing the cross-attention to circumvent the noise of retrieved sentences, our method gains better performance. It shows that our method encodes semantic information by unified Transformer layers on the encoder side.

Table 4. Ablation experiments step by step on the En → De, En → Es, and En → Fr translation tasks on the JRC-Acquis dataset. The “Avg.” column means the averaged result of the 6 directions.

Operation	En-De		En-Es		En-Fr		Avg.
	←	→	←	→	←	→	
SelectNMT	62.23	56.85	66.32	67.08	67.54	68.48	64.81
w/o Selection mechanism	61.45	56.18	64.52	66.02	67.39	67.14	63.45
w/o Segment embedding	59.48	55.65	62.82	65.73	66.06	66.88	62.77

Multiple Domain. Results of our method and other baselines on the En ↔ De translation task are listed in Table 2 and 3. **TM-Transformer** outperforms the **Transformer baseline** on the JRC-Acquis test set but gets the worst performance on the WMT benchmark. It indicates using all retrieved words leads to worse performance when the retrieved sentences contain much irrelevant and noisy information. Our method improves performance on all test sets, which indicates that retrieved sentence pairs can provide useful information to improve the generalization of the model on the multi-domain dataset.

4 Analysis

Ablation Study. To analyze the effect of each component of SelectNMT, we conduct an ablation study on the JRC-dataset. Table 4 summarizes the results of the ablation study. We first ablate the selection mechanism and there is a decrease of 1.06 BLEU points. This means that the selection mechanism is important to avoid the noise of TM. Then, we ablate the segment embedding and there is another decrease of 0.98 BLEU points, showing the need to distinguish the source sentence and retrieved sentences.

Table 5. Different memory settings on the JRC-Acquis dataset. “No-TM” denotes no translation memory is used. “S-TM” and “T-TM” denote the source and target monolingual retrieved sentences. “Bi-TM” denotes using bilingual retrieved sentences.

	En-De		En-Es		En-Fr	
	←	→	←	→	←	→
No-TM	54.55	49.40	59.32	60.82	61.39	62.23
S-TM	55.32	49.16	59.29	61.74	62.73	63.83
T-TM	61.62	56.25	63.79	66.42	67.18	68.01
Bi-TM	62.23	56.85	66.67	67.08	67.54	68.48

Memory Usage. We report the performance of the only using source or target retrieved sentences in Table 5. “S-TM” denotes that the source monolingual

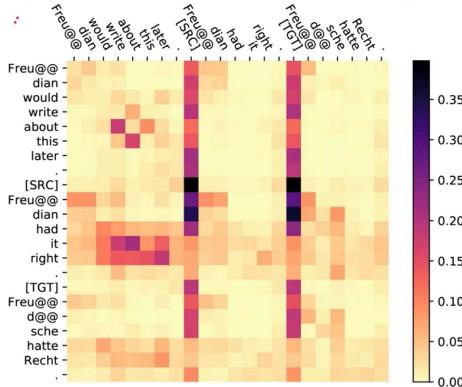


Fig. 3. Visualization for a case of self-attention weights of the selection layer.

retrieved sentences Φ_X^N are incorporated for our model. We further only use the target monolingual sentences Φ_Y^N as memory, which denoted as “T-TM”. “Bi-TM” denotes using the source and target of retrieved pairs simultaneously. The retrieved target sentences contribute greatly to the translation, around +4.47–7.07 BLEU points, as they contain the translated pieces with high probability. Besides, the source of retrieved pairs can also provide a positive contribution since the source contains contextual information.

Attention Visualization. In Fig. 3, we observe the self-attention weights between the source sentence and the retrieved sentences. First, we find that the attention is a kind of alignment between source sentence and translation memory. For example, the two columns immediately following “[SRC]” (“Freu@@”, “dian”) and the three columns immediately following “[TGT]” (“Freu@@”, “d@@”, and “sche”), which are aligned with the source words (“Freu@@”, “dian”), all show higher attention scores. Besides, words with high attention scores also tend to be selected. In the experiment, we found that the above words with higher attention scores were indeed selected. As we force “[SRC]” and “[TGT]” to be selected by adding them to the labels, most of the attention focuses on the delimiter “[SRC]” and “[TGT]”.

Retrieval Size. To investigate the influence of retrieval sizes on the translation quality and the inference speed, we conduct experiments given the different number of retrieved sentence pairs $\Phi_{X,Y}^N (N \in [1, 8])$ in the Fig. 4. We separately test the BLEU scores on the valid set and the inference speed in the same setting (NVIDIA GeForce GTX 1080Ti). With the increase of retrieval size, the BLEU score increases and the inference speed decreases. Although TM with retrieval size ≥ 3 can provide more meaningful pieces, the improvement of BLEU points is minuscule and the inference speed decreases a lot. Therefore, we set the retrieval size equal to 2 in this work.

Results on WMT. We conduct experiments on the 4.5M WMT-14 (En-De) training data. The Transformer baseline (re-implemented by ourselves) without

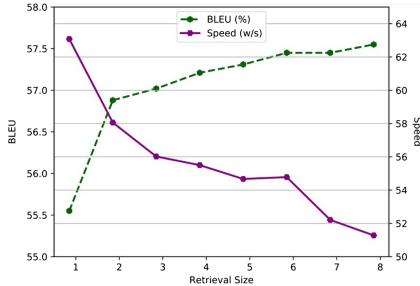


Fig. 4. The inference speed and performance of our method on the JRC-Acquis En → De valid set.

memory gets 27.1 BLEU points. We find that our method only outperforms the baseline by +0.1 BLEU points. The reason for the negligible improvement is that the overlap between the test set and the training set is too little to retrieve useful pieces for the translation.

Table 6. Comparison with other selection methods on the JRC-Acquis dataset.

	En-De		En-Es		En-Fr	
	←	→	←	→	←	→
SMT + Copy	56.11	50.67	62.20	60.03	64.16	63.32
SMT + Concat	56.05	51.92	61.26	62.38	62.26	65.92
Ours	62.23	56.85	64.52	67.08	67.54	68.48

Selection Method. Some works trying to use phrase tables or the outputs of phrase based MT within NMT [34]. Therefore, we also compare our method with the memory-based NMT using an alignment table. For each retrieved pair, we first extract words and their target counterparts using the processed alignment table. Then we concatenate the source sentence and retrieved pieces for the translation. From Table 6, we observe that our method outperforms the baseline by +3.26–6.17 BLEU points. The alignment table is limited by n-gram level alignment and can not capture the contextual information compared with the unified transformer layers.

Analysis of Low-Frequency Sentences. To verify the capability of mitigating the catastrophic forgetting problem [11], we compare our method with Transformer in Fig. 5, where the test set is categorized by the sentence frequency. For low- and medium-frequency sentences ($\leq 75\%$), our method has +5.6–7.4 BLEU points improvement over Transformer. For high-frequency sentences ($> 75\%$), there is +3.5 BLEU points improvement, which shows that SelectNMT is more helpful for low- and medium-frequency sentences than high-frequency sentences.

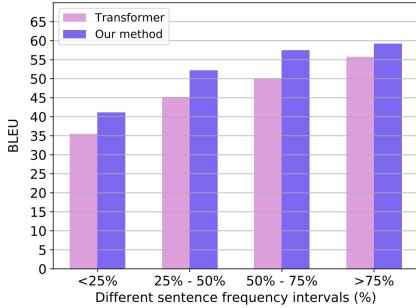


Fig. 5. Comparison of our method with Transformer on sentences of different frequency on the De → En translation tasks on the JRC-Acquis dataset.

Table 7. An example from the JRC-Acquis En → De test set. Our model SelectNMT retrieved sentence pair from the JRC-Acquis training set.

Ground truth	Source	Plant- health treatments
	Target	Behandlung mit Pflanzenschutzmitteln
Neighbor	Source	under no circumstances may <i>plant- health treatments</i> be applied to the fruit after harvest
	Target	nach ihrer Ernte dürfen die Äpfel auf keinen Fall noch <i>mit Pflanzenschutzmitteln</i> behandelt werden
Translation	Baseline	Pflanzenschutzbehandlungen:
	Ours	Behandlungen bei Pflanzenschutzmitteln:

Case Study. A typical comparison of the Transformer baseline and our proposed method is listed in Table 7. For example, our method correctly translates “Plant- health treatment” into “Behandlungen bei Pflanzenschutz”, which appears in the retrieved pair, while the baseline model chooses “Pflanzenschutzbehandlu”. The retrieve sentence pair helps our model generate a translation with better pattern and style, which improves the overall consistency of the translation.

5 Related Work

TM-SMT. The combination of translation memory (TM) and machine translation (MT) have already been used in statistical machine translation (SMT) [14, 18, 23, 26]. Furthermore, phrase-based SMT (PBSMT) systems are augmented with TM by constraining the output to contain retrieved TM matches [19] and enriching the phrase table [4, 25, 28]. Phrase table [7, 17, 33, 35] requires a separate SMT system.

TM-NMT. Neural machine translation (NMT) [2, 10, 27, 30] with the ability of capturing complex functions [1, 9, 20, 22, 32, 36] has been used to build amounts

of translation systems. The semi-parameter method finetunes NMT model on each language pair at inference [8, 21, 29]. The non-parametric method discards expensive gradient descent steps before translation [5, 13, 34]. The previous works [3, 6] use an extra encoder to handle the target sentence of TM with a gating mechanism and N -gram level retrieval approach to improve the retrieval accuracy.

6 Conclusion

In this work, we propose an end-to-end memory-based NMT model called Select-NMT, which focuses on the important retrieved pieces from noisy retrieved context. Our model inputs the concatenation of the source sentences and retrieved pairs at the same time. To avoid the extra noise, we introduce a selection mechanism to choose useful words from memory. Experimental results demonstrate our model can effectively improve performance on the single-domain and multi-domain dataset.

References

1. Akoury, N., Krishna, K., Iyyer, M.: Syntactically supervised transformers for faster neural machine translation. In: ACL 2019, pp. 1269–1281 (2019)
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: ICLR 2015 (2015)
3. Bapna, A., Firat, O.: Non-parametric adaptation for neural machine translation. In: NAACL 2019, pp. 1921–1931 (2019)
4. Biçici, E., Dymetman, M.: Dynamic translation memory: using statistical machine translation to improve translation memory fuzzy matches. In: Gelbukh, A. (ed.) CICLing 2008. LNCS, vol. 4919, pp. 454–465. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78135-6_39
5. Bulté, B., Tezcan, A.: Neural fuzzy repair: integrating fuzzy matches into neural machine translation. In: ACL 2019, pp. 1800–1809 (2019)
6. Cao, Q., Xiong, D.: Encoding gated translation memory into neural machine translation. In: EMNLP 2018, pp. 3042–3047 (2018)
7. Dahlmann, L., Matusov, E., Petrushkov, P., Khadivi, S.: Neural machine translation leveraging phrase-based models in a hybrid search. In: EMNLP 2017, pp. 1411–1420 (2017)
8. Farajian, M.A., Turchi, M., Negri, M., Federico, M.: Multi-domain neural machine translation through unsupervised adaptation. In: WMT 2017, pp. 127–137 (2017)
9. Firat, O., Cho, K., Bengio, Y.: Multi-way, multilingual neural machine translation with a shared attention mechanism. In: NAACL 2016, pp. 866–875 (2016)
10. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. In: ICML 2017, pp. 1243–1252 (2017)
11. Goodfellow, I.J., Mirza, M., Xiao, D., Courville, A., Bengio, Y.: An empirical investigation of catastrophic forgetting in gradient-based neural networks. CoRR [arXiv:1312.6211](https://arxiv.org/abs/1312.6211) (2013)
12. Gu, J., Lu, Z., Li, H., Li, V.O.K.: Incorporating copying mechanism in sequence-to-sequence learning. In: ACL 2016 (2016)

13. Gu, J., Wang, Y., Cho, K., Li, V.O.K.: Search engine guided neural machine translation. In: AAAI 2018, pp. 5133–5140 (2018)
14. Hewavitharana, S., Vogel, S., Waibel, A.: Augmenting a statistical translation system with a translation memory. In: EAMT 2005, vol. 5 (2005)
15. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: ICLR 2015 (2015)
16. Koehn, P., et al.: Moses: open source toolkit for statistical machine translation. In: ACL 2007, pp. 177–180 (2007)
17. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: NAACL 2003 (2003)
18. Koehn, P., Senellart, J.: Convergence of translation memory and statistical machine translation. In: AMTA 2010, pp. 21–31 (2010)
19. Koehn, P., Senellart, J.: Fast approximate string matching with suffix arrays and A* parsing. In: AMTA 2010 (2010)
20. Li, J., Xiong, D., Tu, Z., Zhu, M., Zhang, M., Zhou, G.: Modeling source syntax for neural machine translation. In: ACL 2017, pp. 688–697 (2017)
21. Luong, M.T., Manning, C.D.: Stanford neural machine translation systems for spoken language domains. In: IWSLT 2015, pp. 76–79 (2015)
22. Maruf, S., Martins, A.F.T., Haffari, G.: Selective attention for context-aware neural machine translation. In: NAACL 2019, pp. 3092–3102 (2019)
23. Ortega, J.E., Sánchez-Martínez, F., Forcada, M.L.: Fuzzy-match repair using black-box machine translation systems: what can be expected. In: AMTA 2016, vol. 1, pp. 27–39 (2016)
24. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: ACL 2016 (2016)
25. Simard, M., Isabelle, P.: Phrase-based machine translation in a computer-assisted translation environment. In: MT Summit XII, pp. 120–127 (2009)
26. Simard, M., Langlais, P.: Sub-sentential exploitation of translation memories. In: Machine Translation Summit, vol. 8, pp. 335–339 (2001)
27. Vaswani, A., et al.: Attention is all you need. In: NIPS 2017, pp. 5998–6008 (2017)
28. Wang, K., Zong, C., Su, K.: Integrating translation memory into phrase-based machine translation during decoding. In: ACL 2013, pp. 11–21 (2013)
29. Wuebker, J., Green, S., DeNero, J.: Hierarchical incremental adaptation for statistical machine translation. In: EMNLP 2015, pp. 1059–1065 (2015)
30. Xia, Y., et al.: Deliberation networks: sequence generation beyond one-pass decoding. In: NIPS 2017, pp. 1784–1794 (2017)
31. Xu, J., Crego, J.M., Senellart, J.: Boosting neural machine translation with similar translations. In: ACL 2020, pp. 1580–1590 (2020)
32. Yang, B., Li, J., Wong, D.F., Chao, L.S., Wang, X., Tu, Z.: Context-aware self-attention networks. In: AAAI 2019, pp. 387–394 (2019)
33. Zhang, J., Utiyama, M., Sumita, E., Neubig, G., Nakamura, S.: Improving neural machine translation through phrase-based forced decoding. In: IJCNLP 2017, pp. 152–162 (2017)
34. Zhang, J., Utiyama, M., Sumita, E., Neubig, G., Nakamura, S.: Guiding neural machine translation with retrieved translation pieces. In: NAACL 2018, pp. 1325–1335 (2018)
35. Zhou, L., Hu, W., Zhang, J., Zong, C.: Neural system combination for machine translation. In: ACL 2017, pp. 378–384 (2017)
36. Zhu, C., Yu, H., Cheng, S., Luo, W.: Language-aware interlingua for multilingual neural machine translation. In: ACL 2020, pp. 1650–1655 (2020)



Contrastive Learning for Machine Translation Quality Estimation

Hui Huang¹, Hui Di², Jian Liu¹, Yufeng Chen¹, Kazushige Ouchi²,
and Jinan Xu¹⁽⁾

¹ School of Computer and Information Technology, Beijing Jiaotong University,
Beijing, China

{18112023, jianliu, chenyf, jaxu}@bjtu.edu.cn

² Research & Development Center, Toshiba (China) Co., Ltd., Beijing, China
dihui@toshiba.com.cn, kazushige.ouchi@toshiba.co.jp

Abstract. Machine translation quality estimation (QE) aims to evaluate the result of translation without reference. Existing approaches require large amounts of training data or model-related features, leading to impractical applications in real world. In this work, we propose a contrastive learning framework to train QE model with limited parallel data. Concretely, we use denoising autoencoder to create negative samples based on sentence reconstruction. Then the QE model is trained to distinguish the golden pair from the negative samples in a contrastive manner. To this end, we propose two contrastive learning architectures, namely Contrastive Classification and Contrastive Ranking. Experiments on four language pairs of MLQE dataset show that our method achieves strong results in both zero-shot and supervised settings. To the best of our knowledge, this is the first trial of contrastive learning on QE.

Keywords: Quality estimation · Contrastive learning · Machine translation

1 Introduction

Machine translation quality estimation (QE) aims to evaluate the quality of machine translation automatically without golden reference [2], which has a wide range of applications in post-editing and quality control for MT. The quality can be measured with different metrics, such as DA (Direct Assessment) Score [8] or HTER (Human-targeted Edit Rrror) [22].

The main challenge of QE is data scarcity. Current QE datasets cover only a little proportion of language-pairs in limited domains, with only thousands of triples (source sentence, machine translated sentence and human-assessed score) for training. As an alternative, Fomicheva [7] treats QE as an unsupervised problem, and extracts useful information from the MT system as quality indicator.

H. Huang—Work was done when Hui Huang was an intern at Research and Develop Center, Toshiba (China) Co., Ltd., China.

But their method highly relies on model-related features (such as attention distribution), which may be inaccessible in real-word scenarios.

To perform estimation without QE data or model-related features, we propose a contrastive learning framework, which enables us to perform zero-shot learning on limited parallel data. Firstly, we use a generative pretrained model, BART [16], to create negative samples based on parallel sentence-pairs. We corrupt the reference text in the parallel data based on rules, and then reconstruct the text using BART as a denoising autoencoder. Noise would be inevitably introduced during the corruption-reconstruction, therefore the reconstructed text can be deemed as negative samples. Multiple negative samples can be generated for one sentence-pair based on variations of different corruptions. Secondly, we propose two contrastive learning architectures, namely Contrastive Classification (ConClass) and Contrastive Ranking (ConRank). Both methods learn to differentiate the golden pair from negative samples, formalizing zero-shot QE as a classification or ranking problem. Afterwards, the model could be directly used for estimation, or further finetuned when QE data is available in supervised setting.

We conduct experiments on four medium and low-resource language pairs of MLQE dataset [24], and our method achieves high correlations with human evaluation in both zero-shot and supervised settings, showing the potential of contrastive learning on QE. Besides, BART-based denoising reconstruction to produce negative samples is also a simple but effective paradigm for contrastive learning in natural language processing.

Our contributions can be summarized as follows:

- [1] We firstly propose to use contrastive learning on QE, formalizing zero-shot QE as a classification or ranking problem;
- [2] We firstly propose to use pre-trained denoising autoencoder to generate negative samples for contrastive learning;
- [3] Our method achieves valid results in both zero-shot and supervised settings, without relying on massive parallel data or model-derived features.

2 Related Work

2.1 Machine Translation Quality Estimation

During the trending of deep learning in the field of natural language processing, there are a few works aiming to integrate deep neural network into QE systems. Kim [13] propose for the first time to leverage massive parallel machine translation data to improve QE results. They apply RNN-based machine translation model to extract high-quality feature. Fan [6] replace the RNN-based MT model with Transformer and achieve further improvement.

After the emergence of BERT [5], there are also a few works to leverage pre-trained models on the task of QE [14, 23]. Language models pre-trained on large amounts of text documents are suitable for data-scarce QE task by nature, and have led to significant improvements without complex architecture engineering.

Also, some previous works propose to utilize massive parallel data to strengthen the pretrained model, by performing masked language modeling (MLM) on bilingual concatenated text, and then fine-tune the model on QE data [11, 14]. But MLM is expensive and time consuming. Besides, parallel data is not always readily accessible, especially for some low-resource language pairs (e.g., Sinhala-English and Nepali-English in our work).

Despite most models rely on artificial annotated data, Fomicheva [7] firstly propose to apply QE in an unsupervised manner. They propose to fit human DA scores with three categories of model-related features: A set of unsupervised quality indicators that can be produced as a by-product of MT decoding; the attention distribution inside the Transformer architecture; model uncertainty quantification captured by Monte Carlo dropout. Since these methods are all based on glass-box features, they can only be applied in limited scenarios where inner exploration into the MT model is possible.

2.2 Contrastive Learning

Contrastive learning aims to learn a representation by contrasting positive pairs and negative pairs, and has led to significant improvements in various domains [1, 3, 10]. It is also widely investigated in natural language processing tasks: word representation [19], language modeling [12], unsupervised word alignment [17], caption generation [18], and machine translation [26]. Self-supervised contrastive learning methods do not require any labeled data; instead they sample a mini batch from unsupervised data and create positive and negative samples using data augmentation techniques.

Recently, Wu [25] first propose to evaluate the summary qualities without reference summaries by unsupervised contrastive learning. They construct different types of negative samples with respect to different aspects of the summary qualities, and train the estimator with a ranking loss. But their construction is based on hand-crafted rules, which is non-extensible and easily leads to unnatural text.

3 Our Method

3.1 Denoising Reconstructed Samples

The primary concern for contrastive learning is how to construct negative samples. While most of previous works rely on hand-crafted rules or machine translation [6, 26], in this work, we propose to use BART [16], a pretrained denoising autoencoder, to create negative samples, by reconstructing corrupted reference text, as shown in Fig. 1. BART is trained by corrupting text with an arbitrary noising function, and then learning to reconstruct the original text. This procedure is conducted on massive monolingual data, enabling BART to reconstruct a sentence with corrupted information.

Firstly, we start from parallel sentence pairs, and introduce four types of corruption transformations to the reference text, i.e. token masking, replacement,

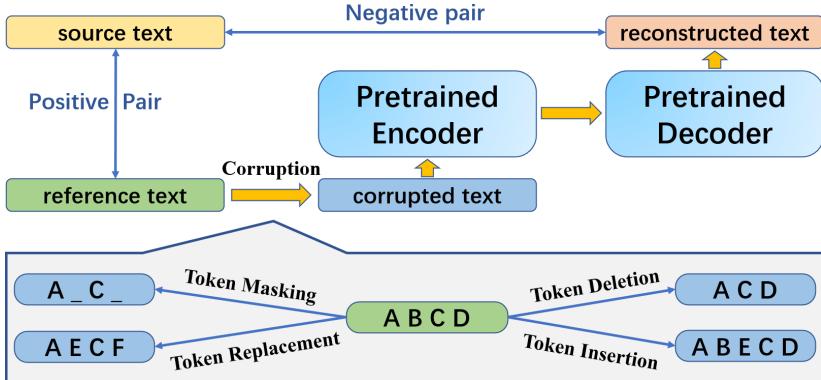


Fig. 1. The denoising reconstruction procedure. The positive pair is from parallel data. Pretrained BART is used to reconstruct corrupted text, and the corruption can be different combinations of four transformations.

deletion and insertion. The corruption is implemented on text spans with lengths drawn from a Poisson distribution with $\lambda = 3$ (except for deletion). The detailed corruption procedure is depicted in Algorithm 1.

Secondly, the corrupted text is fed to BART to generate a reconstruction. Noise would be inevitably introduced by the decoding procedure due to corrupted information, but the reconstructed text is both grammatically fluent and syntactically consistent with the original one. This serves as a decent start point for contrastive learning. The reconstructed text combined with the corresponding source text serves as the negative sample, while the original parallel pair serves as the positive sample, as shown in Fig. 1.

We perform multiple corruption-reconstructions for each reference, with different corruption combinations, positions and ratios, leading to various negative samples against one positive. Notice the corruption ratio can be used to control the noise level. More corruption leads to more noise, and vice-versa, simulating translated text with different qualities.

3.2 Contrastive Training

Intuitively, for a golden reference in parallel data, any noise would lead to a translation with comparatively lower quality. To better utilize the negative samples to improve QE performance, we propose two contrastive learning architectures, namely ConClass and ConRank, to enforce the model to distinguish golden and noised pairs, as shown in Fig. 2.

Both architectures are based on multiple QE models with shared parameters. Given the outstanding performance of pretrained model on QE task, we choose XLM-RoBERTa (abbreviated as XLM-R) [4] as our back-bone QE model. XLM-R is pretrained on massive multilingual data with optimized sampling schedule, and has achieved state-of-the-art results on various multilingual tasks. We follow

Algorithm 1. Text Corruption

Input: Input sentence x with N tokens, replace length $r_l \in \{0, 1\}$, random ratio $r_d \in [0, 1]$, random mask ratio $r_m \in [0, 1]$, and insertion ratio $r_i \in [0, 1]$.

Output: Corrupted sentence x' .

- 1: Draw J text spans from x with totally M tokens, where $M = N \times r_d$
- 2: **for** $i = 1, 2, \dots, J$ **do**
- 3: **if** $r_l = 0$ **then**
- 4: Delete i -th text span.
- 5: **else**
- 6: Generate a random number $f \in [0, 1]$.
- 7: **if** $f > r_m$ **then**
- 8: Replace i -th text span with mask token.
- 9: **else**
- 10: Replace i -th text span with a random token.
- 11: **end if**
- 12: **end if**
- 13: **end for**
- 14: Draw K positions from x , where $K = (N + 1) \times r_i$.
- 15: **for** $i = 1, 2, \dots, K$ **do**
- 16: Insert i -th position with a random token.
- 17: **end for**

Sun [24] to implement our QE model. The bilingual sentence pair is concatenated as the way defined by the pretrained model, and the first hidden representation is fed to a fully connected layer, to generate a logit as the estimation result.

Based on that, in ConClass, the logits for the positive pair and all negative pairs are normalized as a probability distribution and formed into a multi-class classification problem, with the loss function as:

$$L_{conclass} = \text{CrossEntropy}(l_i, y_i),$$

where l_i denotes the logit for the i -th pair and y_i denotes its category.

In ConRank, the logit of the positive pair is trained to be higher than that of the negative pair, with the loss function as:

$$L_{conrank} = \max(0, l' - l + margin),$$

where l and l' denote the logits of golden pair and negative pair, respectively, and $margin$ is a hyper-parameter.

By differentiating the golden and negative samples, we enforce our estimator to capture various aspects of the translation quality. The trained estimator can then be used to evaluate any source-translation pair directly in zero-shot setting. Besides, it can also be further fine-tuned if QE data is available, in which the learning criterion is defined as follow:

$$L_{supervised} = |l - \hat{l}|,$$

where \hat{l} denotes the generated logit, and l denotes the golden estimation. In that case, our contrastive training could be regarded as a pretraining step.

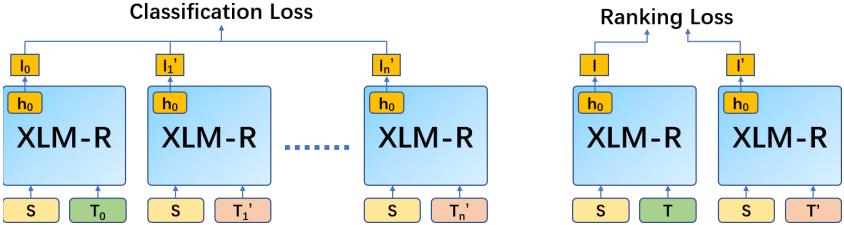


Fig. 2. Contrastive learning architecture. The left denotes ConClass model, and the right denotes ConRank model. \mathbf{S} and \mathbf{T} denote source and target sentences, and \mathbf{h}_0 denotes the first hidden representation of XLM-R, and \mathbf{l} denotes the evaluation logit derived from \mathbf{h}_0 . For ConClass, there are totally n pairs with 1 golden pair and $n - 1$ negative pairs each time, while for ConRank, there are 1 golden pair and 1 negative pair each time. Single quotation denotes noised input or output. Notice parameters of multiple models are shared in both architectures.

4 Experiments

4.1 Setup

Dataset. We use MLQE dataset [24] for evaluating the QE model. MLQE dataset contains source-translation pairs and DA scores [8] for six language pairs, from which we mainly focus on four medium and low-resource pairs, including Romanian-English (Ro-En), Estonian-English (Et-En), Sinhala-English (Si-En) and Nepali-English (Ne-En). We also use the training set of MLQE in supervised scenario to finetune XLM-R.

For parallel data used for denoising reconstruction, we randomly sample 10k from Europarl v8¹ for Ro-En and Et-En. For Si-En and Ne-En, we use parallel sentences from [9], which contains roughly 8k for each direction.

Implementations. We implement our back-bone QE model with XLM-R based on the framework of transformers². We follow Sun [23] and re-implement their system with XLM-R-base due to the limit of computation resource, while their original implementation is on XLM-R-large.

For contrastive learning, we use BART-base to perform reconstruction, and we try two architectures as depicted in Fig. 2. For ConClass, the sample number n is set as 12, with 11 negative and 1 positive. For ConRank, samples in each group are re-shuffled with contrastive pairs of 1 positive and 1 negative.

We adopt Adam optimizer [15] in both contrastive training and supervised fine-tuning, and Pearson coefficient is used as the evaluation metric.

Baselines. In zero-shot setting, we compare with the top-2 methods of Fomicewa [7], which is purely based on glass-box features.

¹ <http://data.statmt.org/wmt16/translation-task>.

² <https://github.com/huggingface/transformers>.

In supervised setting, we mainly compare with the reimplemented result of Sun [23]. On Ro-en and Et-en, we also compare with the MT model-based methods of Kim [13] and Fan [6], for which we use the open-source implementation³ with the default hyper-parameters. The MT model is pre-trained with the parallel data of Europarl v8.⁵ We do not compare with them on Si-En and Ne-En due to the absence of large-scale parallel data.

4.2 Results and Analysis

Table 1. Results on MLQE test set in zero-shot setting. D-TP and D-Lex-Sim refer to the top 2 unsupervised methods of Fomicheva [7].

Model	Ro-En		Et-En		Si-En		Ne-En	
	Pearson	MAE	Pearson	MAE	Pearson	MAE	Pearson	MAE
D-TP	0.693	–	0.642	–	0.460	–	0.558	–
D-Lex-Sim	0.669	–	0.612	–	0.513	–	0.600	–
ConClass	0.715	9.366	0.468	7.217	0.473	5.085	0.579	2.957
ConRank	0.694	5.210	0.508	3.081	0.511	4.016	0.556	3.374

Table 2. Results on MLQE test set in supervised setting. BASE refers to the supervised method of Sun [23]. Postech and bi-expert denote the results of Kim [13] and Fan [6], respectively. Notice the results of BASE are reimplemented by us with XLM-R-base, and our method serves as a pretraining step for them in supervised setting.

Model	Ro-En		Et-En		Si-En		Ne-En	
	Pearson	MAE	Pearson	MAE	Pearson	MAE	Pearson	MAE
postech	0.589	0.584	0.380	0.894	–	–	–	–
bi-expert	0.674	0.502	0.505	0.713	–	–	–	–
BASE	0.846	0.375	0.671	0.620	0.587	0.543	0.697	0.538
ConClass + BASE	0.850↑	0.373	0.696↑	0.611	0.619↑	0.584	0.728↑	0.620
ConRank + BASE	0.850↑	0.394	0.693↑	0.585	0.609↑	0.612	0.723↑	0.429

As we can see in Table 1, in zero-shot setting, we do not surpass the best results of Fomicheva [7]. But our method does not rely on model-related features, while their method fully depends on features extracted from the MT model (such as attention distribution or uncertainty quantification), which may be difficult to obtain in some scenarios, therefore our method is more practicable and extensible. Moreover, with more data and bigger model, our zero-shot performance could be further improved, which is not applicable for their method.

³ <https://github.com/Unbabel/OpenKiwi>.

⁴ <https://github.com/lovecambi/qbrain>.

⁵ <http://data.statmt.org/wmt16/translation-task>.

In supervised setting, as shown in Table 2, our method can improve the results by a large margin for all language-pairs, with the help of only limited parallel data and no additional QE data. As a contrast, MT model-based methods can not provide a valid estimation due to the absence of large-scale parallel data. Contrastive learning is helpful for the model to capture translation-related features. With only limited parallel data, different kinds of translation errors can be simulated by different combinations of corruptions, making the model more robust and extensible.

4.3 Different Methods to Create Negative Samples

In this section, we want to study different methods to create negative samples in our framework. There are roughly three methods to create negative samples in QE, i.e. denoising reconstruction, rule-based method, and machine translation, as exemplified in Table 3. We create different groups of negative samples with different methods, and then use them for contrastive training.

Table 3. Negative samples for Romanian-English, created via denoising reconstruction (abbreviated as DR), rule-based method (abbreviated as RB), and the provided machine translation (abbreviated as MT) model. Red denotes noise. Notice the rule-based sample is unnatural and grammatically erroneous, while denoising reconstructed text is grammatically valid and fluent. On the other hand, denoising reconstruction introduces noise while not changing the syntactic structure, while the machine translated sample is syntactically inconsistent yet semantically correct.

Source	Permiteți-mi să reiterez și să evidențiez principalele puncte din raportul meu
Reference	Allow me to reiterate and highlight the main points of my report
DR	Allow special mention to reiterate and highlight the representative of my article
RB	Allow consort to reiterate and highlight the of main points my report
MT	Let me repeat and highlight the main points in my report

For denoising reconstruction, we try different combinations and ratios of the four corruptions, and create multiple negative samples for each reference, following the procedure of Fig. 1. The TER score [22] between each reconstructed sample and its reference is calculated, and all samples are grouped into three categories with low, medium and high TER scores, simulating translated results with low, medium and high quality.

For rule-based method, we also corrupt the samples with different variations of the corruptions (except for token masking, since the *[mask]* token does not appear in real text), and group the results into three sets with low, medium

and high TER scores. For machine translation, we use the MT model (which is also used to generate the QE data) released by MLQE dataset [24] to translate the source text in parallel data. We adjust the beam size to make sure different groups contain the same number of samples.

Table 4. Pearson results on MLQE test set with different negative samples in zero-shot setting. Avg-TER refers to the average TER score. DR, RB and MT denotes results of denoising reconstruction, rule-based method and machine translation, respectively. Low, medium and high denote the noise level.

Direction		DR			RB			MT
		Low	Medium	High	Low	Medium	High	
Ro-En	avg-TER	0.132	0.299	0.561	0.128	0.303	0.565	0.316
	ConClass	0.638	0.703	0.715	0.432	0.468	0.329	0.083
	ConRank	0.537	0.552	0.578	0.409	0.441	0.317	0.328
Ne-En	avg-TER	0.145	0.325	0.614	0.149	0.321	0.620	0.663
	ConClass	0.422	0.516	0.549	0.318	0.317	0.204	0.230
	ConRank	0.321	0.516	0.545	0.303	0.335	0.197	0.196

As shown in Table 4, reconstructed samples with TER too low are harmful for contrastive learning. Too little noise means the sample is roughly correct, and in that case, enforcing the model to distinguish them is meaningless. To make sure the negative samples are truly “negative” is important for the model to capture quality-related features.

Negative samples obtained via rule-based method also underperform. This is because rule-based method always lead to unnatural text with evident grammatical error, which is not consistent with the real MT error distribution. The QE model can learn little knowledge about translation evaluation during contrastive training, since it is too easy to distinguish the negative samples.

Negative samples generated by MT system lead to even outrageous results. We believe this is because its semantic integrity and syntactic inconsistency, which is captured by the model as mendacious clue when doing classification. Actually, most translated candidates are semantically correct but syntactically inconsistent [20]. In other words, they are also not really “negative”.

4.4 Compare with Metric-Based Method

Machine translation metrics, such as BLEU [21], aim to evaluate the translation based on the reference text. As denoted by [6], given the noised target sentence and its corresponding reference, pseudo QE score can be obtained by automatic metrics, therefore zero-shot QE can be performed based on parallel data. Since no real QE data is incorporated, we refer to it as metric-based zero-shot QE.

We use the four groups of noised samples obtained in Sect. 4.3, and generate BLEU scores using sacremoses.⁶ The BLEU score between negative samples and references is used as pseudo QE score to train the QE model directly.

Table 5. Pearson results on MLQE test set with different zero-shot QE methods. We release the best results of different contrastive learning architectures in both directions. DR and MT denote the negative samples generated via denoising reconstruction and machine translation, respectively.

Direction	Method	DR			MT
		Low	Medium	High	
Ro-En	Metric-based	0.606	0.682	0.688	0.324
	Contrastive	0.638	0.703	0.715	0.083
Ne-En	Metric-based	0.401	0.504	0.519	0.376
	Contrastive	0.422	0.516	0.549	0.230

As shown in Table 5, the metric-based method does not outperform contrastive learning in both directions. Automatic metric itself is an approximation, therefore the pseudo score is not suitable to be directly used as the learning objective. On the contrary, contrastive learning transforms the regression problem into a ranking or classification problem, therefore the learning objective is valid and unbiased.

Interestingly, for MT-based negative samples, the metric-based method outperforms contrastive learning in both directions. We believe this is because BLEU is designed to tackle the inconsistent morphological or syntactical structures between MT-derived samples and references, adapting the metric-based pseudo score to our scenario.

5 Conclusion

In this paper, we propose a contrastive learning framework, to utilize limited parallel data to boost QE performance. We use denoising generative pretrained model to reconstruct corrupted sentences, leading to various negative samples with consistent error distribution and controllable noise. We also propose two contrastive architectures, viewing QE as a classification or ranking problem. Experimental results show our model achieves strong results in both zero-shot and supervised settings.

Denoising pretrained model is especially useful for generating negative samples for quality estimation. In the future, we will transfer our framework to the estimation of other generative tasks. We would also use denoising reconstructed samples for the training of automatic post-editing.

⁶ www.github.com/alvations/sacremoses.

Acknowledge. The research work described in this paper has been supported by the National Key R&D Program of China 2020AAA0108001 and the National Nature Science Foundation of China (No. 61976015, 61976016, 61876198 and 61370130). The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve this paper.

References

1. Baevski, A., Zhou, Y., Mohamed, A., Auli, M.: wav2vec 2.0: a framework for self-supervised learning of speech representations. In: Advances in Neural Information Processing Systems, vol. 33, pp. 12449–12460 (2020)
2. Blatz, J., et al.: Confidence estimation for machine translation. In: COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics, COLING, Geneva, Switzerland, 23 August–27 August 2004, pp. 315–321 (2004)
3. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning, pp. 1597–1607. PMLR (2020)
4. Conneau, A., et al.: Unsupervised cross-lingual representation learning at scale. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 8440–8451, July 2020. Online
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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), pp. 4171–4186, June 2019
6. Fan, K., Wang, J., Li, B., Zhou, F., Chen, B., Si, L.: “Bilingual expert” can find translation errors. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 6367–6374 (2019)
7. Fomicheva, M., et al.: Unsupervised quality estimation for neural machine translation. Trans. Assoc. Comput. Linguist. **8**, 539–555 (2020)
8. Graham, Y., Baldwin, T., Moffat, A., Zobel, J.: Can machine translation systems be evaluated by the crowd alone. Nat. Lang. Eng. **23**, 1–28 (2015)
9. Guzmán, F., et al.: Two new evaluation datasets for low-resource machine translation: Nepali-English and Sinhala-English (2019)
10. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9729–9738 (2020)
11. Hu, C., et al.: The NiuTrans system for the WMT20 quality estimation shared task. In: Proceedings of the Fifth Conference on Machine Translation, pp. 1018–1023. Association for Computational Linguistics, November 2020. Online
12. Huang, J., Li, Y., Ping, W., Huang, L.: Large margin neural language model. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October–November 2018, pp. 1183–1191 (2018)
13. Kim, H., Jung, H.Y., Kwon, H., Lee, J.H., Na, S.H.: Predictor-estimator: neural quality estimation based on target word prediction for machine translation. ACM Trans. Asian Low-Resource Lang. Inf. Process. **17**, 1–22 (2017)
14. Kim, H., Lim, J.H., Kim, H.K., Na, S.H.: QE BERT: bilingual BERT using multi-task learning for neural quality estimation. In: Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2), Florence, Italy, pp. 85–89, August 2019

15. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization (2017)
16. Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7871–7880. Association for Computational Linguistics, July 2020. Online
17. Liu, Y., Sun, M.: Contrastive unsupervised word alignment with non-local features. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI 2015, pp. 2295–2301. AAAI Press (2015)
18. Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A.L., Murphy, K.: Generation and comprehension of unambiguous object descriptions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 11–20 (2016)
19. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 1–9 (2013)
20. Negri, M., Turchi, M., Chatterjee, R., Bertoldi, N.: ESCAPE: a large-scale synthetic corpus for automatic post-editing. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan. European Language Resources Association (ELRA), May 2018
21. 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 of the Association for Computational Linguistics, pp. 311–318 (2002)
22. Snover, M.G., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A study of translation edit rate with targeted human annotation. In: AMTA (2006)
23. Sun, S., et al.: An exploratory study on multilingual quality estimation. In: Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, Suzhou, China, pp. 366–377, December 2020
24. Sun, S., Guzmán, F., Specia, L.: Are we estimating or guesstimating translation quality? In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6262–6267, July 2020. Online
25. Wu, H., Ma, T., Wu, L., Manyumwaa, T., Ji, S.: Unsupervised reference-free summary quality evaluation via contrastive learning. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 3612–3621. Association for Computational Linguistics, November 2020. Online
26. Yang, Z., Cheng, Y., Liu, Y., Sun, M.: Reducing word omission errors in neural machine translation: a contrastive learning approach. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, pp. 6191–6196. Association for Computational Linguistics, July 2019



Sentence-State LSTMs For Sequence-to-Sequence Learning

Xuefeng Bai¹, Yafu Li¹, Zhirui Zhang², Mingzhou Xu³, Boxing Chen², Weihua Luo²,
Derek Wong³, and Yue Zhang^{1,4(✉)}

¹ School of Engineering, Westlake University, Hangzhou, China

{baixuefeng, liyafu}@westlake.edu.cn

² Alibaba DAMO Academy, Hangzhou, China

{zhirui.zrz, boxing.cbx, weihua.luowh}@alibaba-inc.com

³ NLP2CT Lab, Department of Computer and Information Science,

University of Macau, Macau, China

derekfw@um.edu.mo

⁴ Institute of Advanced Technology, Westlake Institute for Advanced Study, Hangzhou, China

yue.zhang@wias.org.cn

Abstract. Transformer is currently the dominant method for sequence to sequence problems. In contrast, RNNs have become less popular due to the lack of parallelization capabilities and the relatively lower performance. In this paper, we propose to use a parallelizable variant of bi-directional LSTMs (BiLSTMs), namely sentence-state LSTMs (S-LSTM), as an encoder for sequence-to-sequence tasks. The complexity of S-LSTM is only $\mathcal{O}(n)$ as compared to $\mathcal{O}(n^2)$ of Transformer. On four neural machine translation benchmarks, we empirically find that S-SLTm can achieve significantly better performances than BiLSTM and convolutional neural networks (CNNs). When compared to Transformer, our model gives competitive performance while being 1.6 times faster during inference.

Keywords: Neural machine translation · Sentence-State LSTMs · Bi-directional LSTMs · CNN · Transformers

1 Introduction

Sequence encoding is the task for deriving dense representations for sequences of input tokens. It plays a fundamental role in neural models for natural language processing, where a sentence is naturally treated as token sequences. In the literature, three major types of sequence encoders have been dominantly used, namely Recurrent Neural Network (RNNs) [1, 14, 22, 35], Convolutional Neural Networks (CNNs) [11, 16], and self-attention networks (SANs) [31, 32, 40]. In particular, bi-directional LSTMs (BiLSTMs) [28], as the mostly adopted RNN variant, encode a sequence by executing a recurrent function from left to right and another recurrent function from right to left to consider bi-directional information flows. [28]. CNNs encode a sequence of input vectors by using stacked layers of convolutional neural networks over local n-grams. SANs encode each input token by linearly aggregating all input vectors using attention [1], with its mostly adopted variant being Transformer [40].

For classification [13, 16, 33] and structured prediction problems [20, 23], all three types of sequence encoders (and their variants) have been heavily investigated. For example, for syntactic parsing [25], different RNNs [9] and SANs [39] have been thoroughly discussed for their relative advantages. For example, for sentiment classification, numerous variants for both CNNs [16], RNNs [37] and SANs [44] have been investigated. Recently, with the rise of pre-training techniques, Transformer-based language model architectures [8, 21, 26] have become the dominant methods for these tasks.

For sequence-to-sequence modeling, in contrast, the number of variant architectures that have been investigated for RNNs and CNNs are relatively much fewer compared with variants of Transformer. In particular, for RNNs, the architecture of Bahdanau *et al.* [1] has been the mostly used architecture, while for CNNs, the method of Gehring *et al.* [11] has been one of the few structures considered. In contrast, for Transformer, a wide range of variants have been investigated [2, 7, 19]. One reason is that the highly competitive performance of Transformer, which gives the state-of-the-art results for machine translation, text summarization and other tasks such as dialogue generation [46]. On the other hand, there are potentially advantages of their CNN and RNN counterparts which can be potentially useful. For instance, LSTMs have the strength of avoiding gradient issues over large numbers of back-propagation. In fact, Chen *et al.* [4] show that a combination of Transformer encoder and LSTM decoder can give highly competitive results for machine translation when compared to Transformer.

The relative disadvantage of LSTM sequence-to-sequence models is that the highly sequential encoder structure precludes parallelization. As a result, the time complexity of bi-directional LSTM (BiLSTM) encoders is higher compared with Transformer encoders. In addition, it has been shown that BiLSTMs face difficulty capturing long-range dependencies due to large numbers of recurrent steps necessary for encoding a large sequence. One alternative to BiLSTM encoding is sentence-state LSTM (S-LSTM [47]), which has a parallel design of recurrent encoding functions, allowing efficient sequence encoding and direct node communications by the introduction of a global node. On classification and sequence labeling tasks, S-LSTM has been shown to give strong results compared with BiLSTMs, CNNs and Transformers [47]. However, relatively little work has been done to investigate S-LSTMs for sequence-to-sequence modeling.

We fill this gap by considering RNN sequence-to-sequence models, which has the S-LSTM encoder and a LSTM decoder. The model structure is shown in Fig. 1. Compared with BiLSTM, S-LSTM updates hidden states of all words in parallel and is better at modeling long-ranged dependency, thanks to the use of a global node. Compared with Transformer, S-LSTM only requires linear time $\mathcal{O}(n)$ to encode n words, which are significantly less compared to a quadratic complexity $\mathcal{O}(n^2)$ for Transformer, and thus potentially has better applicability to long input, which can be the case for documents [45], dialogue [24], etc.

Results on four machine translation benchmarks show that the S-LSTM sequence-to-sequence model obtains competitive performance compared with Transformer, and gives higher BLEU than the baseline BiLSTM and CNN sequence-to-sequence architectures. In addition, S-LSTM runs the fastest when compared to all the baseline encoders, being 1.6 times faster than Transformer during inference. We release our code at <https://github.com/muyeby/S-LSTM-nmt>.

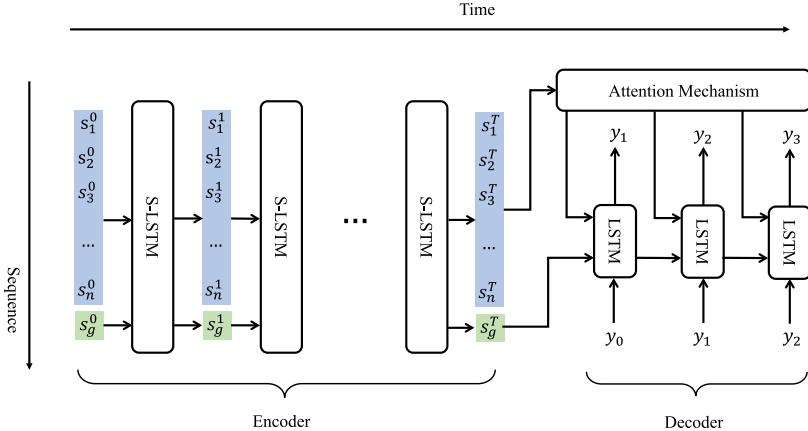


Fig. 1. Overview of the proposed model. The S-LSTM parameters are tied at different time steps.

2 Approach

Following previous work [1, 40], the proposed model has an encoder-decoder architecture [35]. As shown in Fig. 1, given an input sentence $X = \{w_1, w_2, \dots, w_n\}$, the encoder acts as a recurrent state transition function (named S-LSTM), which takes as input a set of initial states $\{s_1^0, s_2^0, \dots, s_n^0, s_g^0\}$, and generates more abstract states $\{s_1^t, s_2^t, \dots, s_n^t, s_g^t\}$ at each time step $t, t \in [0, T]$. In particular, $\{s_1^t, s_2^t, \dots, s_n^t\}$ denote the states of input tokens and s_g^t denotes a sentence-level state. The final sentence-level state s_g^T is used to initialize the decoder network. The decoder generates a target token sequence one by one. At each time step t , the attention mechanism is applied over the final encoder hidden states $\{s_1^T, s_2^T, \dots, s_n^T\}$ and combined with the current hidden state of the LSTM decoder to predict the next target word y_t .

2.1 Sentence-State LSTM Encoder

We follow Zhang *et al.* [47] to build a S-LSTM model for sentence encoding. S-LSTM can be regarded as a type of recurrent graph neural networks (GNNs, [18, 34, 41]) which views a whole sentence as an input graph and performs information exchange between words in an iterative manner. Figure 2(a) shows how nodes communicate with each other when context window size is 1. Each word is connected with its neighbors and the global node is connected with all words. At each time step, S-LSTM updates each word state according to its local context and a sentence-level global feature. The global state is computed based on all word states.

Formally, at step t , the S-LSTM updates a set of $\{s_1^t, s_2^t, \dots, s_n^t\}$ and a sentence-level global state s_g^t simultaneously. The word hidden states $\{s_1^t, s_2^t, \dots, s_n^t\}$ record features for words $\{w_1^t, w_2^t, \dots, w_n^t\}$ under the sentential context and are initialized by corresponding word embeddings $\{x_1, x_2, \dots, x_n\}$. The global state s_g^t records features for the whole sentence and is initialized by an averaged-sum of initial word hidden states. Following LSTM, each state s_*^t is divided into a hidden state h_*^t and a cell state c_*^t .

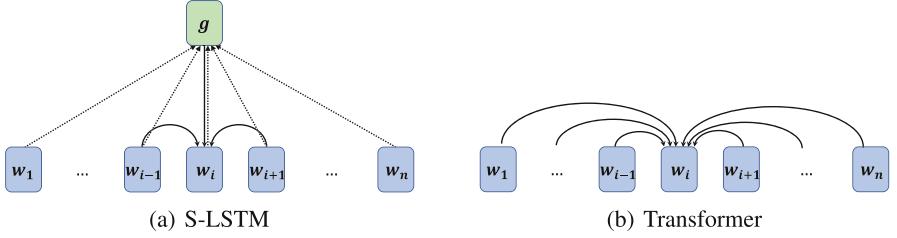


Fig. 2. Information exchange in S-LSTM and Transformer.

The former serves as working memory which carries information from immediately previous events and new inputs while the latter stores long-term memory.

Word State Transition. Given the k -th word, S-LSTM calculates its context hidden and cell states at time t as:

$$\begin{aligned} h_l^{t-1}, c_l^{t-1} &= \text{avg}(h_{k-w}^{t-1}, \dots, h_{k-1}^{t-1}), \text{avg}(c_{k-w}^{t-1}, \dots, c_{k-1}^{t-1}), \\ h_r^{t-1}, c_r^{t-1} &= \text{avg}(h_{k+1}^{t-1}, \dots, h_{k+w}^{t-1}), \text{avg}(c_{k+1}^{t-1}, \dots, c_{k+w}^{t-1}), \end{aligned} \quad (1)$$

where w is the context window size, and h_l^{t-1}, h_r^{t-1} are left and right context hidden states, c_l^{t-1}, c_r^{t-1} are left and right context cell states, respectively.

The hidden state h_k^t is then calculated based on $h_k^{t-1}, h_l^{t-1}, h_r^{t-1}, h_g^{t-1}$ and their corresponding cell states $c_k^{t-1}, c_l^{t-1}, c_r^{t-1}, c_g^{t-1}$, as well as its word embedding x_k :

$$\begin{aligned} m_k^t &= [h_k^{t-1}, h_l^{t-1}, h_r^{t-1}, h_g^{t-1}, x_k], \\ i &= \sigma(W_i m_k^t + b_i), \\ l &= \sigma(W_l m_k^t + b_l), \\ r &= \sigma(W_r m_k^t + b_r), \\ f &= \sigma(W_f m_k^t + b_f), \\ d &= \sigma(W_s m_k^t + b_s), \\ o &= \sigma(W_o m_k^t + b_o), \end{aligned} \quad (2)$$

$$\begin{aligned} u &= \tanh(W_u m_k^t + b_u), \\ \hat{l}, \hat{r}, \hat{f}, \hat{d} &= \text{softmax}(l, r, f, d) \\ c_k^t &= \hat{l} \odot c_l^{t-1} + \hat{f} \odot c_k^{t-1} + \hat{r} \\ &\quad + \odot c_r^{t-1} + \hat{d} \odot c_g^{t-1}, \\ \hat{c}_k^t &= i \odot u + (1 - i) \odot c_k^t, \\ h_k^t &= o \odot \tanh(\hat{c}_k^t), \end{aligned} \quad (3)$$

where σ denotes the sigmoid activation function, $[h_k^{t-1}, h_l^{t-1}, h_r^{t-1}, h_g^{t-1}, x_k]$ is the vector concatenation of $h_k^{t-1}, h_l^{t-1}, h_r^{t-1}, h_g^{t-1}$ and x_k . $\hat{l}, \hat{r}, \hat{f}, \hat{d}$ are gates controlling information from the left context cell, the current cell, the right cell and the global cell respectively. i, o are input and output gates which control information from the current input state u_k^t and the current cell state \hat{c}_k^t , respectively. $W_i, W_l, W_r, W_f, W_s, W_o, W_u, b_i, b_l, b_r, b_f, b_s, b_o$ and b_u are model parameters.

Table 1. Per-layer complexity, parallelizability and maximum path lengths for different layer types. n is the sequence length, k is the kernel size of convolutions.

Layer Type	Complexity	Parallelizable	Maximum path lengths
RNNs	$\mathcal{O}(n)$	False	$\mathcal{O}(n)$
CNNs	$\mathcal{O}(n)$	True	$\mathcal{O}(\log_k(n))$
SANs	$\mathcal{O}(n^2)$	True	$\mathcal{O}(1)$
S-LSTM	$\mathcal{O}(n)$	True	$\mathcal{O}(1)$

Global State Transition. As shown in Fig. 2(a), the global state s_g^t is updated based on previous hidden states concerning all words and the global node as well as the corresponding cell states:

$$\begin{aligned}
 \tilde{h}_g &= \text{avg}(h_1^t, h_2^t, \dots, h_n^t), & \mathcal{F} &= \{f_g, f_1, \dots, f_n\}, \\
 f_g &= \sigma(W_g h_g^{t-1} + V_g \tilde{h}_g + b_g), & \hat{f}_g, \hat{f}_1, \dots, \hat{f}_n &= \text{softmax}(\mathcal{F}; \tau) \\
 f_k &= \sigma(W_f h_g^{t-1} + V_f h_k^{t-1} + b_f), \quad (4) & c_g^t &= \hat{f}_g \odot c_g^{t-1} + \sum_{k=1}^n \hat{f}_k \odot c_k^{t-1}, \\
 \forall k &\in [1, n] & o_g &= \sigma(W_o h_g^{t-1} + V_o \tilde{h}_g + b_o), \\
 & & h_g^t &= o_g \odot \tanh(c_g^t),
 \end{aligned} \tag{5}$$

where `avg` is the average pooling function, τ is a temperature, which is applied to logits to affect the final probabilities from the softmax function. f_g, f_k are forget gates which aggregate information from the previous global and word cell states, respectively. $W_g, V_g, W_f, V_f, W_o, V_o, b_g, b_f$ and b_o are model parameters. It should be noted that the transition function of global state and word states use independent model parameters.

2.2 Comparison with RNNs, CNNs and Transformer

As shown in Table 1, we compare S-LSTM with RNNs, CNNs and Transformers by considering three aspects:

- The computational complexity per layer;
- Computational parallelism for sequence encoding;
- The maximum path length of the input sentence.

The first two items jointly determine the computational and memory cost of a system. The third item assesses the capacity of learning long-range dependencies.

RNNs. Standard RNNs passes information from one end to the other along the sentence. As a result, RNNs lack parallelizability and the time complexity of RNNs scales with the input sentence length. In contrast, S-LSTM takes a whole sentence as input, updating word states simultaneously. Therefore, S-LSTM is **more time-efficient** than standard RNNs. In addition, RNNs are proven to be limited in modeling long

sequences [38]. S-LSTM alleviate this issue by using a global node, which reduces the maximum path length of arbitrary words to $\mathcal{O}(1)$, compared to RNNs with $\mathcal{O}(n)$. Theoretically, S-LSTM can be ***more accurate*** when dealing with longer inputs.

Transformer and CNNs. The main difference between the S-LSTM layer and a standard Transformer layer can be two folds: 1) S-LSTM is based on gating and recurrent mechanism while Transformer uses attention mechanism. 2) As shown in Fig. 2(b), Transformer updates word hidden states based on the whole context, resulting in a complexity of $\mathcal{O}(n^2)$. In contrast, S-LSTM use a local context together with a sentence-level feature, which significantly ***reduces the time complexity*** from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$. Similar to CNNs, S-LSTM updates word states using larger context with increasing time steps. However, S-LSTM additionally considers sentence-level context, which makes S-LSTM more powerful for modeling important long-term dependencies.

2.3 LSTM Decoder

Our system adopts a LSTM decoder, which takes the final encoder hidden states as input and generates a target sentence in an auto-regressive manner. The decoder is composed of a stack of L identical layers, each of which consisting of two sub-modules: 1) the first one is a LSTM layer and 2) the second one is an attention layer which collects information from encoder. Specifically, we initialize the hidden and cell states of LSTM decoder using final state of the global node:

$$\bar{h}_0^l, \bar{c}_0^l = h_g^T, c_g^T, \forall l \in [1, L]. \quad (6)$$

At each time step t , the l -th decoder layer takes output hidden states of the preceding layer as input and produce new hidden states as:

$$\begin{aligned} \tilde{h}_t^l, \tilde{c}_t^l &= \text{LSTM}(\bar{h}_{t-1}^l, \bar{c}_{t-1}^l, \bar{h}_t^{l-1}), \\ \bar{h}_t^l &= \text{Attn}(\tilde{h}_t^l, M), \end{aligned} \quad (7)$$

where Attn refers an attention function, $M = \{h_1^T, h_2^T, \dots, h_n^T\}$ represents the final encoder hidden states.

The final-layer decoder hidden states are used to predict the probability of a target word:

$$P(y_t | y_{<t}, X) = \text{softmax}(W_d \bar{h}_t^L), \quad (8)$$

where $y_{<t} = [y_0, y_1, \dots, y_{t-1}]$ and W_d is a parameter matrix.

Following recent NMT systems [40], we equip our method with absolute position embeddings and feed forward layers to improve model capacity.

2.4 Training

Given a translation sentence pair $\langle X, Y \rangle$ where $Y = \{y_0, y_1, \dots, y_m\}$, our model is trained to minimize the cross-entropy loss:

$$\ell = - \sum_{t=1}^m \log p(y_t | y_{<t}, X; \Theta), \quad (9)$$

where Θ is the set of model parameters.

Table 2. Number of sentences of each dataset.

Setting	NIST ZH → EN	WMT16 EN ↔ RO	IWSLT16 EN ↔ De	IWSLT17 EN ↔ FR
Train	1.9M	606k	160k	236k
Dev	1.7k	2.0k	7.3k	9.5k
Test	1.4k/1.8k/1.1k	2.0k	6.8k	2.6k

3 Experiments

Dataset. We conduct experiments on NIST Chinese to English (NIST ZH → EN) translation tasks as well as WMT16 English-to/from-Romanian (WMT16 EN ↔ RO), IWSLT14 English-to/from-German (IWSLT14 EN ↔ DE) and IWSLT17 English-to/from-French (IWSLT17 EN ↔ FR) datasets. For NIST ZH → EN, we use parts of the bitext provided within NIST’12 OpenMT.¹ We choose NIST MT06 as the validation set, and MT04, MT05, MT08 as the test sets. For WMT16 EN ↔ RO, we use the same data for training and pre-processing as Sennrich *et al.* [29] but remove back-translated sentences. We use newstest2016 for evaluation. For IWSLT14 EN ↔ DE, we use *fairseq* script *prepare-iwslt14.sh* to split the train/dev dataset and merge the multiple testsets dev2010, dev2012, tst2010-tst2012 for testing. For IWSLT17 EN ↔ FR, we merge dev2010, tst2010-2015 for validation and tst2016, 2017 for evaluation. Table 2 lists the statistics of above datasets.

Experimental Settings. We implement our model following Transformer-base. The hidden state size is set as 512, and feed-forward layer size is 2048 for NIST12 ZH → EN as well as WMT16 EN ↔ RO, and 1024 for other datasets. We use a dropout of 0.1 on NIST ZH → EN and 0.3 on the other datasets. We select the context window size from [1, 2, 3] and the softmax temperature from [1, 5, 10], based on validation loss. We preprocess sentences using byte pair encoding ([30]; BPE), jointly learned from the concatenation of the parallel training dataset only. We use the Adam optimizer [17] with the same learning rate schedule strategy as Vaswani *et al.* [40] with 4k warmup steps. The learning rate linearly increases from $1e^{-7}$ to $7e^{-4}/5e^{-4}$ for NIST ZH→EN and the other datasets, respectively. Each mini-batch consists of 4,096 source and target tokens respectively. ALL experiments run on *fairseq*² with 4 RTX 2080TI GPUs.

Evaluation. We use multi-bleu.perl to evaluate our model for a fair comparison with previous systems. For NIST ZH → EN, we use beam size 4 and case-insensitive BLEU scores. For other datasets, we use beam size 5 and report case-sensitive BLEU scores.

3.1 Main Results

Translation Quality. Table 3 shows the main results on the NIST ZH → EN dataset. We compare our method with previous models based on BiLSTM, convolution [11] and

¹ LDC2000T46, LDC2000T47, LDC2000T50, LDC2003E14, LDC2005T10, LDC2002E18, LDC2007T09, LDC2004T08.

² <https://github.com/pytorch/fairseq/>.

Table 3. BLEU score NIST ZH → EN dataset.

Model	MT04	MT05	MT08	Average
BiLSTM	42.1	35.8	28.4	35.4
ConvS2S [11]	49.7	44.3	39.0	44.3
Transformer [40]	53.0	47.8	43.5	48.1
S-LSTM	52.3	47.9	43.6	47.9

Table 4. BLEU scores and Inference speedup (over Transformer) on benchmark datasets. “ \dagger ” indicates results are based on both parallel and back-translated corpus. “ \ddagger ” means results are based on our own implementations.

Model	WMT16		IWSLT14		IWSLT17		Average	Speedup
	EN → RO	RO → EN	EN → DE	DE → EN	EN → FR	FR → EN		
GRU [29]	23.9	27.8	—	—	—	—	—	$0.7 \times \ddagger$
ConvS2S [11]	30.2 \dagger	—	—	—	—	—	—	$0.8 \times$
Transformer [40]	32.4	32.1	28.6	34.8	38.7	38.9	34.3	$1.0 \times$
S-LSTM	32.2	32.0	28.2	34.4	38.2	38.5	33.9	1.6 \times

self-attention [40]. Among previous systems, Transformer obtains the best results on all testsets. Compared with Transformer, S-LSTM gives 0.7 lower BLEU score on MT04 but slightly better results on MT05 and MT08 testset. On average, S-LSTM gives a BLEU score of 47.9, which is 0.2 lower than Transformer, but 12.5 and 3.6 points higher than BiLSTM and ConvS2S, respectively.

Table 4 shows the BLEU scores on WMT16 EN \leftrightarrow RO, IWSLT14 EN \leftrightarrow DE and IWSLT17 EN \leftrightarrow FR. Transformer gives the best results on all datasets, even outperforming ConvS2S which uses 2M additional back-translation data for training. S-LSTM gives highly competitive BLEU scores compared to Transformer. In addition, S-LSTM obtains BLEU scores of 32.2 and 32.0 on EN \rightarrow RO and RO \rightarrow EN, respectively, which are significantly better than GRU and ConcS2S system.

In total, the results on above 4 benchmarks indicate that the S-LSTM architecture is better than previous RNN and convolution based methods, and can be a competitive alternative system in both large and small datasets.

Translation Speed. We also measure the translation speed, which is important in practical scenarios. As shown in the last column of Table 4, BiLSTM gives the slowest speed because BiLSTM encoder requires sequential operations. ConvS2S is faster than BiLSTM but slower than Transformer. The reason is that ConvS2S requires deeper encoder and decoder for long-range information exchange, which slows down the translation speed. S-LSTM gives the fastest speed, being 60% faster than Transformer. This is consistent with our theoretical analysis in Sect. 2.2, showing the efficiency of S-LSTM.

4 Analysis

We conduct an ablation study to assess the effectiveness of each component in S-LSTM, and how does the number of recurrent step affects model performance.

4.1 Ablation Study

Table 5. Ablation Study on WMT16 RO → EN.

Model	BLEU
SLSTM	32.0
- global node	30.8
- global initialized decoder	31.6
- temperature	31.7

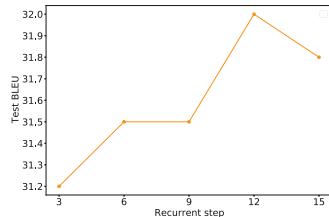


Fig. 3. Performance against model depth.

Table 5 shows BLEU scores on WMT16 RO → EN regarding different model structures. Specifically, we consider three configurations: 1) removing the global node; 2) initializing the LSTM decoder states with zeros rather than sentence-level state and 3) removing the temperature parameter in Eq. 5. First, after removing the sentence-level global node, S-LSTM loses sentence-level information and deteriorates into a local model. This results in a performance decrease of 1.2, showing that the global node is indispensable for S-LSTM. Second, using the sentence-level state to initialize the decoder leads to improvements. The test BLEU score increase from 31.6 to 32.0. Finally, the temperature parameter also has positive impact on model performance.

4.2 Effect of Recurrent Steps

We study the effect of recurrent steps on WMT16 RO → EN. As shown in Fig. 3, the test BLEU becomes better with growing recurrent steps, increasing from 31.2 to 32.0. This shows the effectiveness of recurrent information exchange in S-LSTM. When the recurrent step is 12, our system reaches the peak performance, and the performance starts to decrease with even more steps. A possible explanation for this is that too many recurrent steps leads to overfitting, which can hurt the model performance.

5 Related Work

Our work is related to prior research on sequence-to-sequence learning and efficient sequence encoding.

5.1 Seq2seq Modeling

Seq2seq models, which follow an encoder-decoder paradigm [6, 15, 35], have shown dominant performance in various sequence-to-sequence tasks such as machine translation (MT) and text summarization. Bahdanau *et al.* [1] introduces attention mechanism into seq2seq framework and first surpass the performance of phrase-based MT using RNN-based encoder and decoder. The RNN-based NMT approach (RNMT) quickly becomes the standard paradigm for seq2seq tasks, followed by Wu *et al.* [43], which achieves promising performance on multiple benchmark datasets. The major drawback of RNMT models is failure for parallel training due to the inherently sequential nature of RNN. To this end, ConvS2S [11] apply convolutional neural networks as basic blocks for both encoder and decoder, allowing to fully parallel training. The ConvS2S is shown to outperform RNMT models on both translation quality and training speed. However, CNN models are weak in learning distant dependencies as only local features are captured in each layer [3, 40]. Vaswani *et al.* [40] propose Transformer, which addresses deficiencies in both RNN-based models and CNN-based models: (1) it avoids sequential dependencies and thus allows for parallel computing during training and (2) its self-attention layer enables each position to connect to all other positions in a sequence. Compared with Transformer, the proposed S-LSTM model maintains the ability of parallel training and long-dependency modeling, while efficiently updating word states based on a local context as well as a sentence-level node.

5.2 Efficient Sequence Encoding

There have been many prior efforts on designing efficient sequential encoders. For example, Cho *et al.* [6] introduce gated recurrent unit (GRU), which has less gates than LSTM and merges the hidden and cell states. Szegedy *et al.* [36] and Gao *et al.* [10] propose Grouped RNNs and CNNs, respectively, which reduce the number of parameters as well as the computation complexity by group size. Transformer XL [7] splits a input sequence into segments and uses a segment-level recurrence mechanism to avoid long-range information exchange. Sparse-Transformer [5] uses pre-defined mask patterns to compute subsets of the attention matrix. Star-Transformer [12] exploits a star-shaped structure which reduces token connections from quadratic to linear. Roy *et al.* [27] extend Sparse-Transformer by using learnable patterns to compute sub-attention matrix. Reformer [19] replaces standard dot-product attention by one that uses locality-sensitive hashing. Linformer [42] approximates standard attention matrix by a low-rank one. Our work shares the same goal with above approaches. In particular, we focus on a recurrent architecture and reduce the computation complexity to $\mathcal{O}(n)$.

6 Conclusion

We investigated a recurrent neural network structure for sequence to sequence modeling, using S-LSTM for the encoder and LSTM for the auto-regressive decoder. Results on four machine translation datasets show that our system obtains 1.6 translation speed up compared with Transformer, while giving comparable results. Our work shows that recurrent neural networks can be a useful alternative to Transformer in neural sequence modeling.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: ICLR (2015)
2. Beltagy, I., Peters, M.E., Cohan, A.: Longformer: the long-document transformer. [arXiv:2004.05150](https://arxiv.org/abs/2004.05150) (2020)
3. Bengio, Y., Frasconi, P., Schmidhuber, J., Elvezia, C.: Gradient flow in recurrent nets: the difficulty of learning long-term dependencies(2001)
4. Chen, M., et al.: The best of both worlds: combining recent advances in neural machine translation. In: ACL (2018)
5. Child, R., Gray, S., Radford, A., Sutskever, I.: Generating long sequences with sparse transformers. [arXiv:1904.10509](https://arxiv.org/abs/1904.10509) (2019)
6. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Moschitti, A., Pang, B., Daelemans, W. (eds.) EMNLP (2014)
7. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q.V., Salakhutdinov, R.: Transformer-XL: attentive language models beyond a fixed-length context. In: ACL (2019)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (2019)
9. Dyer, C., Ballesteros, M., Ling, W., Matthews, A., Smith, N.A.: Transition-based dependency parsing with stack long short-term memory. [arXiv:1505.08075](https://arxiv.org/abs/1505.08075) (2015)
10. Gao, F., Wu, L., Zhao, L., Qin, T., Cheng, X., Liu, T.Y.: Efficient sequence learning with group recurrent networks. In: NAACL (2018)
11. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.: Convolutional sequence to sequence learning. In: ICML (2017)
12. Guo, Q., Qiu, X., Liu, P., Shao, Y., Xue, X., Zhang, Z.: Star-transformer. In: NAACL (2019)
13. Hassan, A., Mahmood, A.: Deep learning for sentence classification. In: LISAT (2017)
14. Kalchbrenner, N., Blunsom, P.: Recurrent continuous translation models. In: EMNLP (2013)
15. Kalchbrenner, N., Espeholt, L., Simonyan, K., van den Oord, A., Graves, A., Kavukcuoglu, K.: Neural machine translation in linear time. arXiv (2016)
16. Kim, Y.: Convolutional neural networks for sentence classification. In: EMNLP (2014)
17. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: ICLR (2015)
18. Kipf, T., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)
19. Kitaev, N., Kaiser, L., Levskaya, A.: Reformer: the efficient transformer. In: ICLR (2020)
20. Kübler, S., McDonald, R., Nivre, J.: Dependency parsing. Synthesis Lectures on Human Language Technologies (2009)
21. Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: ACL (2020)
22. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: EMNLP (2015)
23. Nguyen, N., Guo, Y.: Comparisons of sequence labeling algorithms and extensions. In: ICML (2007)
24. Oluwatobi, O., Mueller, E.T.: DLGNet: a transformer-based model for dialogue response generation. [arXiv: Computation and Language](https://arxiv.org/abs/Computation and Language) (2020)
25. Pickering, M.J., Van Gompel, R.P.: Syntactic parsing. In: Handbook of Psycholinguistics (2006)
26. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019)
27. Roy, A., Saffar, M., Vaswani, A., Grangier, D.: Efficient content-based sparse attention with routing transformers. TACL **9**, 53–68 (2021)

28. Schuster, M., Paliwal, K.: Bidirectional recurrent neural networks. *IEEE Trans. Sig. Process.* **45**, 2673–2681 (1997)
29. Sennrich, R., Haddow, B., Birch, A.: Edinburgh neural machine translation systems for WMT 16. In: WMT (2016)
30. Sennrich, R., Haddow, B., Birch, A.: Improving neural machine translation models with monolingual data. In: ACL (2016)
31. Shaw, P., Uszkoreit, J., Vaswani, A.: Self-attention with relative position representations. In: NAACL-HLT (2018)
32. Shen, T., Zhou, T., Long, G., Jiang, J., Pan, S., Zhang, C.: DiSAN: directional self-attention network for RNN/CNN-free language understanding. In: AAAI (2018)
33. Socher, R., et al.: Recursive deep models for semantic compositionality over a sentiment treebank. In: EMNLP (2013)
34. Song, L., Zhang, Y., Wang, Z., Gildea, D.: A graph-to-sequence model for AMR-to-text generation. In: ACL (2018)
35. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: NIPS (2014)
36. Szegedy, C., et al.: Going deeper with convolutions. In: CVPR (2015)
37. Tang, D., Qin, B., Feng, X., Liu, T.: Effective LSTMs for target-dependent sentiment classification. In: COLING (2016)
38. Tang, G., Müller, M., Gonzales, A.R., Sennrich, R.: Why self-attention? A targeted evaluation of neural machine translation architectures. In: EMNLP (2018)
39. Tian, Y., Song, Y., Xia, F., Zhang, T.: Improving constituency parsing with span attention. In: EMNLP (2020)
40. Vaswani, A., et al.: Attention is all you need. In: NIPS (2017)
41. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio', P., Bengio, Y.: Graph attention networks. In: ICLR (2018)
42. Wang, S., Li, B.Z., Khabsa, M., Fang, H., Ma, H.: Linformer: self-attention with linear complexity. [arXiv:2006.04768](https://arxiv.org/abs/2006.04768) (2020)
43. Wu, Y., et al.: Google's neural machine translation system: bridging the gap between human and machine translation. ArXiv (2016)
44. Xu, H., Liu, B., Shu, L., Yu, P.S.: BERT post-training for review reading comprehension and aspect-based sentiment analysis. In: NAACL (2019)
45. Zhang, J., et al.: Improving the transformer translation model with document-level context. In: EMNLP (2018)
46. Zhang, Y., et al.: DialoGPT: large-scale generative pre-training for conversational response generation. In: ACL (2020)
47. Zhang, Y., Liu, Q., Song, L.: Sentence-state LSTM for text representation. In: ACL (2018)



Guwen-UNILM: Machine Translation Between Ancient and Modern Chinese Based on Pre-Trained Models

Zinong Yang^{1(✉)}, Ke-jia Chen^{1,2(✉)}, and Jingqiang Chen^{1,2}

¹ Nanjing University of Posts and Telecommunications, Nanjing, China

² Jiangsu Key Lab of Big Data Security and Intelligent Processing, Nanjing, China
`{chenkj, cjq}@njupt.edu.cn`

Abstract. Ancient Chinese literatures are not only the unique cultural heritage of China but also the treasures of world civilization. Nevertheless, it has become quite difficult for modern people to comprehend or even create ancient works with the evolution of language in the long history. Translation is therefore playing a key role in bridging the two eras. This paper is to develop an automatic translation method between ancient and modern Chinese literature. To start with, an open sourced sentence level parallel corpus of ancient-modern Chinese is established since there is no available parallel corpus open for use. As the seq2seq-based machine translation models do not work well on this task, the pre-trained model UNILM is then applied in our method considering the monolingual characteristics of this task. Furthermore, the ancient Chinese pre-trained model - Guwen-BERT is utilized to further improve the performance of the method. The quality of translation is evaluated by both Human Evaluation and two automatic metrics: a) case-sensitive BLEU scores and b) Imagery Conservation (I.C), which is first developed in this paper. The experimental results under all metrics show that our method can generate higher quality of translation.

Keywords: Guwen-UNILM · Ancient-Modern Chinese MT · Pre-trained models

1 Introduction

Ancient Chinese literature, especially proses and poems, are not only the unique cultural heritage of China but also the common treasures of the world civilization. Nevertheless, it has become quite difficult for modern people to comprehend or even create these ancient works with the evolution of language in long history. Several examples shown in Fig. 1 indicate that the ancient Chinese is significantly different from modern Chinese though they share some common characters.

However, the translation of ancient Chinese texts usually requires the participation of experts in the relevant field and involves lots of time and labor even for historians and linguists. It becomes the luxury and scarce resources for ordinary people in modern society to learn and enjoy the value of traditional Chinese culture from piles of retained literatures in history. Moreover, the published translations by experts are almost all ancient-to-modern. There are few reverse modern-to-ancient translations. It would be very interesting for modern people to create their own works in ancient expression as

Ancient Chinese	Modern Chinese	English Version
胜者有力 自胜者强	能战胜别人的人只能 被称为有力量的人。 能战胜自己的人才可 以是精神上的强者。	<i>He who conquers others is merely strong physically.</i> <i>He who conquers himself is truly invincible mentally.</i>
人生自古谁无死 留取丹心照汗青	人生自古以来有谁能 够长生不死? 我要留一片爱国的丹 心映照史册。	<i>Who can avert his death since time immemorial?</i> <i>Let my heart remain true to shine in the annals.</i>

Fig. 1. Examples of ancient-modern Chinese parallel texts. Intuitively, ancient Chinese language tend to be more abbreviated.

an aesthetic exercise. The linguists can also benefit from the reverse translation to find the clues for their research.

Machine Translation (MT) greatly relieves the human efforts in conversion between two different languages and the quality of translation has also made a significant improvement with the research on Neural Machine Translation (NMT), represented by seq2seq [15] framework and Transformer [16] using self-attention mechanism. While these models work well for bilingual translation, it is far from satisfactory when applied to the monolingual task like translation between ancient and modern Chinese.

In summary, there is a strong need to develop a specific machine translation application in the context of ancient Chinese.

The first challenge is the lack of ancient-modern Chinese parallel texts. The success of NMT in bilingual translation largely relies on the huge number of available parallel corpus while there are only few published ancient-to-modern translations with high quality¹ and the translators devoted to this area are also limited.

The second challenge comes from the language itself. The versatility of Chinese characters in performing diverse functions in syntactical structure makes the ancient expression concise but confusing. Besides, metaphor and imagery are widely used in ancient Chinese to evoke sensory and emotional experiences, which is an advantage in aesthetics but increases ambiguity in delivering message.

Accordingly, this paper made the following contributions by developing a method named *Guwen-UNILM*, where *Guwen* means ancient Chinese.

First, we collected a large number of ancient Chinese texts and their corresponding translated modern texts from several official ancient Chinese literature websites that collect the authentic translation works. However, these translations are article-level parallel texts, which are too long and complicated for model training. For simplicity, we then manually establish two fine-grained sentence-to-sentence parallel corpora on this basis.²

Second, the UNILM [2] model initially proposed for natural language generation tasks (e.g., generative question answering, abstractive summarization), is applied for the first time on the translation task, as inspired by the characteristic that ancient Chinese is usually concise and seems to be the *summary* of its corresponding modern ver-

¹ Three Hundred Tang Poems, Mao Zedong Selected Poems and One Hundred 100 Sung Proses.

² Our corpora and code are available at: <https://www.github.com/cloudyskyy/Guwen-UNILM>.

sion. Based on BERT architecture, UNILM can only be trained on monolingual corpora and hence suitable for monolingual machine translation. Moreover, UNILM can utilize BERT parameters pre-trained on large monolingual corpora as its initial parameters, thereby reducing the impact of insufficient parallel data.

Third, Guwen-BERT, the BERT model specifically pre-trained on a monolingual corpus built from 15,694 ancient Chinese books containing 1.7 billion Chinese tokens, is utilized to initialize the parameters of UNILM. Pre-training on ancient Chinese texts in advance can help UNILM better adapt to the translation task and speed up the convergence of the model.

Forth, an automatic evaluation metric: Imagery Conservation (I.C) is specially proposed to evaluate the translation between ancient and modern Chinese.

Finally, our method achieves a significant improvement over the state-of-art on both automatic evaluation metrics and human evaluation.

2 Related Work

The development of Machine Translation (MT) has shifted from the early Statistical Machine Translation (SMT) [5,11] to Neural Machine Translation (NMT) with the prevalence of deep learning in recent years. Great breakthroughs have been made with the RNN-based [15] or CNN-based [6] NMT approaches to improve both the quality and capability of the MT. The attention mechanism [10,15] further enhances the seq2seq framework and Transformer becomes one of the most outstanding NMT methods by using self-attention mechanism [17].

Although the NMT based methods can manage bilingual translation well in many areas with great success, their performance in the monolingual context such as translation between ancient and modern Chinese [20] are yet to be improved due to the shortage of available parallel corpus that can be used for model training and the fact that the Chinese characters in ancient expression can perform multiple syntactic role at various positions in a sentence with different meaning, which could confuse the translation model. As the result, the pre-training is required to better learn the language representation.

Pre-trained models (PTMs) led by BERT [1] have achieved unprecedented success in the field of Natural Language Processing (NLP). They can learn universal language representation that will benefit downstream NLP tasks by modeling the masked pieces in a sentence with plenty of corpora. Moreover, these models can be reused to minimize the effort of training a new model for a new task. PTMs are frequently used in Natural Language Understanding (NLU) tasks such as extractive question answering [18], named entity recognition [14] and passage ranking [12] and recently have been extended to NLG domain.

UNILM (Unified Language Model Pre-training for Natural Language Understanding and Generation), developed by designing a unique self-attention mask matrix and a set of novel cloze tasks, can be used to perform NLG task such as generative question answering, abstractive summarization and question generation. Like other PTMs, UNILM can only be applied on monolingual tasks, which makes it suitable for our translation task between ancient and modern Chinese. Moreover, UNILM can directly use the weights of BERT pre-trained on a large number of monolingual corpora as initial parameters, thereby alleviating the problem resulted from insufficient parallel text.

3 The Guwen-UNILM Framework

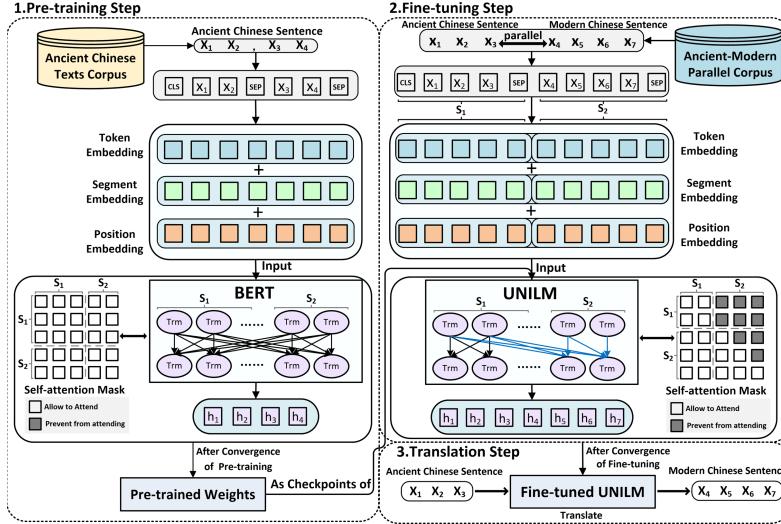


Fig. 2. Overview of the Guwen-UNILM framework. We take the ancient-to-modern translation direction as an example in this figure. Vice versa. The framework mainly includes three steps: pre-training step, fine-tuning step and translation step. The detailed information of our framework is given in Sect. 3.

The framework of Guwen-UNILM as shown in Fig. 2 consists of three parts: pre-training, fine-tuning and translation.

In the pre-training step, the BERT model is trained by plentiful ancient Chinese texts until convergence; in the fine-tuning step, the parameters of the pre-trained model are used as the checkpoints for UNILM to run further seq2seq training, with the help of self-attention masks. At the end of seq2seq learning, the fine-tuned UNILM is applied to translate the test data.

3.1 Pre-training Step

The whole pre-training procedure is shown in Fig. 3.

First, the ancient Chinese monolingual corpus³ is built from 15,694 ancient Chinese literatures (including poems, proses, documentations, fictions, etc.), containing 1.7 billion Chinese characters. The data need pre-processing first to convert traditional Chinese characters and variant Chinese characters⁴ into simplified Chinese characters for consistence. Each Chinese character is then treated as one token, and the 23,287 characters that occur most frequently are selected as the vocabulary for pre-training of Guwen-BERT.

³ The corpus is built from the data provided by Daizhige Ancient Literature at <http://www.daizhige.org>.

⁴ These characters were once used in ancient China but now have been simplified.

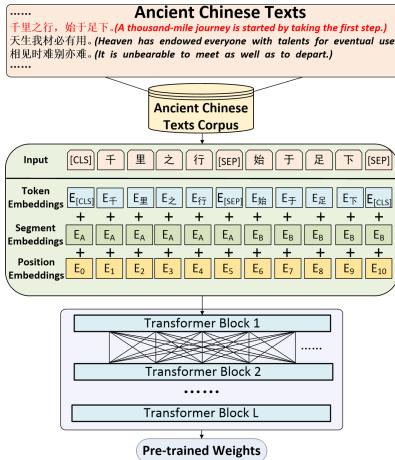


Fig. 3. Overview of the Guwen-BERT pre-training.

Specifically, the roBERTa [9] pre-training approach is adopted in our paper. The word sequence represented by x is embedded into the input vector $\{\mathbf{x}\}_{i=1}^{|x|}$, which is the sum of the token embeddings, segment embeddings and position embeddings of x . $\{\mathbf{x}\}_{i=1}^{|x|}$ is then fed to an L -layer Transformer $\mathbf{H}^l = \text{Transformer}_l(\mathbf{H}^{l-1})$, $l \in [1, L]$ to learn the contextual representation $\mathbf{H}^l = [\mathbf{h}_1^l, \dots, \mathbf{h}_{|x|}^l]$ of each layer l . In each layer, the self-attention head \mathbf{A}_l is calculated to aggregate the output of the previous layer.

$$\left\{ \begin{array}{l} \mathbf{Q} = \mathbf{H}^{l-1} \mathbf{W}_l^Q \\ \mathbf{K} = \mathbf{H}^{l-1} \mathbf{W}_l^K \\ \mathbf{V} = \mathbf{H}^{l-1} \mathbf{W}_l^V \end{array} \right. \quad (1a)$$

$$\mathbf{A}_l = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}_l \quad (2)$$

where the matrices $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{|x| \times d_k}$ refer to queries, keys and values, respectively. They are obtained by multiplying the $l-1$ th layer's contextual representation $\mathbf{H}^{l-1} \in \mathbb{R}^{|x| \times d_h}$ by the linear projection matrices $\mathbf{W}_l^Q, \mathbf{W}_l^K, \mathbf{W}_l^V \in \mathbb{R}^{d_h \times d_k}$.

3.2 Fine-Tuning Step

The fine-tuning procedure of Guwen-UNILM is shown in Fig. 4.

Parallel Corpus Establishment. In order to achieve ancient Chinese translation, the first thing to do is to establish an ancient-modern Chinese parallel corpus for training. From several official Chinese literature websites,⁵ we collected around 5 thousand parallel documents containing 3 million characters. However, these parallel documents are too long for model training, some of which even contains thousands of characters. For simplicity, we manually established a more fine-grained sentence-level parallel corpus on these texts, which is being continuously enlarged and polished up. The corpus is currently open-sourced and the detailed information can be available in Sect. 4.1.

⁵ <https://www.gushiwen.org/>, <http://www.ewenyan.com/>.

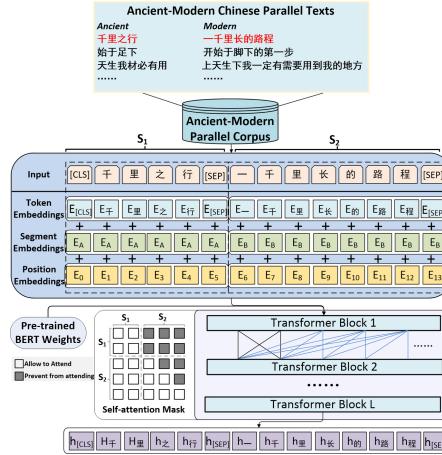


Fig. 4. Overview of the Guwen-UNILM fine-tuning.

Seq2seq Learning of Guwen-UNILM. Assume that we are translating ancient Chinese into modern Chinese, each line of the parallel corpus consists of two parts: the ancient Chinese sequence x_a and the modern Chinese sequence x_m . We take the concatenated word sequence $x = x_a + x_m$ as the input sequence of Guwen-UNILM. The input representation vector of Guwen-UNILM is $\{\mathbf{x}\}_{i=1}^{|x|} = \{\mathbf{x}_a\}_{j=1}^{|x_a|} + \{\mathbf{x}_m\}_{k=1}^{|x_m|}$, which is the sum of token embeddings, segment embeddings, and position embeddings. The parameters of pre-trained BERT are used as checkpoints of Guwen-UNILM. The input vector $\{\mathbf{x}\}_{i=1}^{|x|}$ is fed into the multi-layer Transformer $\mathbf{H}^l = \text{Transformer}_l(\mathbf{H}^{l-1})$, $l \in [1, L]$. In UNILM, the calculation of the attention head \mathbf{A}_l is similar to that in BERT, except for the introduction of the self-attention mask matrix \mathbf{M} .

$$\left\{ \begin{array}{l} \mathbf{Q} = \mathbf{H}^{l-1} \mathbf{W}_l^Q \\ \mathbf{K} = \mathbf{H}^{l-1} \mathbf{W}_l^K \end{array} \right. \quad (3a)$$

$$\left\{ \begin{array}{l} \mathbf{V} = \mathbf{H}^{l-1} \mathbf{W}_l^V \end{array} \right. \quad (3b)$$

$$\left\{ \begin{array}{l} \mathbf{M}_{ij} = \begin{cases} 0 & \text{allow to attend} \\ -\infty & \text{prevent from attending} \end{cases} \end{array} \right. \quad (3c)$$

$$\mathbf{M}_{ij} = \begin{cases} 0 & \text{allow to attend} \\ -\infty & \text{prevent from attending} \end{cases} \quad (4)$$

$$\mathbf{A}_l = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}} + \mathbf{M}\right)\mathbf{V}_l \quad (5)$$

The mask matrix \mathbf{M} here determines the visibility between tokens. As is shown in Fig. 4, the tokens in the first word sequence (ancient Chinese) can attend to each other, while the tokens in the second word sequence (modern Chinese) can only attend to the leftward tokens. For example, given an ancient Chinese sequence x_1x_2 and its target parallel modern Chinese sequence $x_3x_4x_5$, the input tokens of the model is “[CLS] x_1 x_2 [SEP] x_3 x_4 x_5 [SEP]”. Here, [CLS], x_1 , x_2 , and the first [SEP] can attend to each other. However, x_3 can only attend to the four tokens leftwards. This is different from BERT, as BERT is a bidirectional language model where all the tokens in a sequence are visible to each other.

Translation and Loss Function. In order to output the translation, the end of the multi-layer Transformer is connected to a linear classifier (e.g. a softmax classifier) to generate a probability distribution over the vocabulary, thereby generating translation. In the above instance, given a pair of parallel texts: $x_a = x_1x_2$ and $x_m = x_3x_4x_5$. The input representation of x_a is “[CLS] x_1 x_2 [SEP]”, and then appended with a “[MASK]” token.

$$[\text{CLS}] x_1 x_2 [\text{SEP}] \text{[MASK]}$$

As mentioned above, the tokens of x_a can attend to each other, but cannot attend to the [MASK] token, while the [MASK] token can attend to all the leftward tokens. The six tokens is then encoded by UNILM as $\mathbf{h}_{[\text{CLS}]}$, \mathbf{h}_{x_1} , \mathbf{h}_{x_2} , $\mathbf{h}_{[\text{SEP}]}$, $\mathbf{h}_{[\text{MASK}]}$. The representation vector $\mathbf{h}_{[\text{MASK}]}$ is then fed to a softmax classifier to achieve a probability distribution over the vocabulary V .

$$S = W\mathbf{h}_{[\text{MASK}]} + b \quad (6)$$

$$P(t|\mathbf{h}_{[\text{MASK}]}) = \frac{e^{S_i}}{\sum_j^{|V|} e^{S_j}} \quad (7)$$

where $S \in \mathbb{R}^{1 \times |V|}$ is the score vector on the vocabulary. W and b are trainable parameters of linear transformation, and t is the predicted token of the classifier. The token with the maximum probability is selected as the prediction of model.

The predicted token is then appended to the input sequence to replace the [MASK] token, and the decoding continues until the [SEP] token emerges. In our realization, the decoding process is enhanced by beam search [4]. The categorical cross entropy loss \mathcal{L} of the whole sentence is defined as:

$$\mathcal{L} = - \sum_{i=1}^{|x_m|} y_i \log(\hat{y}_i) \quad (8)$$

where x_m is the true target sentence, y_i is a token in x_m , and \hat{y}_i is the predicted token generated by the classifier corresponding to y_i .

4 Experiment

Our method Guwen-UNILM is tested with different datasets and training settings, and the performance of Guwen-UNILM is analyzed with that of comparative methods.

4.1 Datasets

Three corpora (Daizhige, Literature and History) are used in the experiment. The Daizhige corpus is a monolingual ancient Chinese corpus for pre-training BERT. Literature and History are both parallel corpora used to fine-tune UNILM. The purpose of selecting two corpora is to verify the compatibility of the model under different text style, as the former is indirect and ambiguous while the latter is realistic and repetitive. Therefore, the translation performance of two different text styles might vary.

The statistics of datasets are shown in Table 1.

Table 1. Statistics of corpora.

Corpora	Type	Train/Valid./Test
Daizhige	Monolingual	—
Literature	Parallel	30k/2k/2k
History	Parallel	30k/2k/2k

4.2 Experimental Setup

In the experiment, both the ancient-to-modern translation and the modern-to-ancient translation are conducted. Each parallel corpus is split into training set, validation set and testing set (see Table 1).

Unlike English, Chinese words can consist of one to multiple characters. Therefore, we conducted experiments using both single character and word as tokens to see if and how the translation quality is affected. The segmentation of modern and ancient sentences is accomplished with Jieba⁶ and Jiayan Toolkit,⁷ respectively.

The Guwen-UNILM model is constructed by using the bert4keras⁸ toolkit. The number of Transformer layers of UNILM L is set to 12. The learning rate of the Adam optimizer is $1e^{-5}$, where $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The maximum sequence length l of the input (e.g., the total length of two sentences) is set to 128. In the decoding stage, the beam size of beam search is 4.

4.3 Comparative Models

The following baseline models are compared to Guwen-UNILM: (1) LSTM and Bi-LSTM with attention mechanism (2) GRU and Bi-GRU with attention mechanism (3) Transformer (4) BERT-UNILM (5) roBERTa-UNILM. In our experiments, LSTM, Bi-LSTM, GRU, Bi-GRU and Transformer are built by OpenNMT toolkit [7].

As mentioned above, we also studied on the influence of word segmentation on these five models. The model labeled with “‡” indicates that the input sentences of this model are pre-processed by word segmentation tools. In BERT-UNILM and roBERTa-UNILM, the modern Chinese texts are used to pre-train the original BERT and roBERTa respectively.

4.4 Evaluation Metrics

Both automatic evaluation metrics and Human Evaluation are used to access the performance of models. The automatic evaluation metrics include BLEU [13] and Imagery Conservation (I.C). Human Evaluation includes fluency, semantic consistency, and meaningfulness of the texts.

BLEU is a widely used machine translation metric that measures the degree of overlap between the generated sentence and referred sentence. Specifically, we use case-sensitive BLEU score to measure the quality of our generated sentences. Different from traditional machine translation tasks, in our task (especially in the translation of ancient

⁶ <https://github.com/fxsjy/jieba>.

⁷ <https://github.com/jiaeyan/Jiayan>.

⁸ <https://github.com/bojone/bert4keras>.

literary works), sometimes although the BLUE score of the generated sentence is low, it is still evaluated by human experts as a high-quality translation.

Imagery Conservation is the automatic evaluation metric specially designed for our task. From the experiments, it is found that even if the translated sentence is significantly different from the original sentence, the character and word that are used to express *imagery* (“意象” in Chinese) should be the same. These words are often objective entities (usually nouns such as flower, grass, light, tears, etc.) in the texts that help evoke sensory or emotional experiences, e.g. Therefore, the Imagery Conservation metric is defined as the number of imagery nouns that appear in both ancient and modern Chinese sentences.

Human Evaluation is usually believed more reliable and credible than the automatic evaluation metrics, given the translation task is more organic than mechanic. The evaluation criterion includes three dimensions with each of a metric score from 1 to 5. *Fluency* metric evaluates whether the generated text is grammatically fluent. *Consistency* metric evaluates how well the translated sentence conserve the content of the original sentence. *Meaningfulness* metric is to judge whether the words in the generated sentence has actual meanings. The final human evaluation score is calculated by averaging score given by all human experts.

4.5 Results and Discussion

The case-sensitive BLEU scores, I.C scores and Human Evaluation of all methods on the Literature corpus and the History corpus are compared in Table 2, Table 3, and Table 4, respectively.

Results on Literature Corpus. Three methods using UNILM (B-UNILM, R-UNILM and Guwen-UNILM) are significantly superior to other benchmark methods in BLEU score for both modern-to-ancient translation and ancient-to-modern translation. However, it seems that most models work better for modern-to-ancient translation than ancient-to-modern translation in terms of BLEU score. The possible reason is that the former translation is similar to sentence summary, while the latter translation is similar to sentence expansion, in which the more selective vocabulary leads to greater differences from the reference translation. On the I.C metric, Guwen-UNILM conserves 86.3% (modern-to-ancient) and 70.0% (ancient-to-modern) of the imagery words that appear in the text (100% on groundtruth), performing best among all methods. On both metrics, Guwen-UNILM performs better than B-UNILM and R-UNILM, which verifies the effectiveness of adding the pre-trained model on ancient texts into the translation task and also explains why RoBERTa was chosen for pre-training.

Results on History Corpus. On the History corpus, Guwen-UNILM still achieves the state-of-art results in BLEU score and I.C score. It is also noted that the BLEU scores of all models on the History corpus are significantly higher than that on the Literature corpus in the two translation directions. The first possible reason is that there exist plentiful objective facts in the historical records such as names of places, people, time and event facts will stay the same in the translated sentences. The second possible reason is that texts in history corpus are more rigid and consistent, while those in the Literature corpus are more romantic and imaginative.

Table 2. BLEU scores on the Literature corpus. “‡” indicates that the input sentences of this model are pre-processed by the word segmentation tool. B-UNILM, R-UNILM and G-UNILM are short for BERT-UNILM, roBERTa-UNILM and Guwen-UNILM respectively. (BT) indicates the back translation (BT) is applied to the model.

Model	Modern to Ancient (Literature)					Ancient to Modern (Literature)				
	BLEU	BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEU	BLEU-1	BLEU-2	BLEU-3	BLEU-4
LSTM‡	1.11	9.2	1.8	0.4	0.4	1.28	13.4	3.2	0.7	0.2
Bi-LSTM‡	1.26	15.7	4.0	0.6	0.2	1.38	14.2	3.6	0.8	0.2
GRU‡	1.09	13.6	2.9	0.4	0.2	1.40	15.1	3.9	0.9	0.3
Bi-GRU‡	1.49	15.2	3.3	0.7	0.4	1.51	15.4	3.7	0.9	0.3
Transformer‡	1.56	15.4	4.2	0.8	0.5	1.58	14.2	3.7	0.9	0.4
LSTM	2.17	26.0	5.1	0.9	0.4	1.98	23.0	4.8	0.9	0.2
Bi-LSTM	2.90	29.1	6.8	1.3	0.4	2.61	25.4	6.0	1.3	0.4
GRU	2.75	28.2	5.7	1.1	0.5	2.78	26.7	6.4	1.5	0.4
Bi-GRU	3.03	30.4	6.2	1.3	0.6	3.11	26.4	6.6	1.6	0.5
Transformer	5.43	31.8	9.5	3.1	1.8	3.17	28.2	7.8	2.1	0.7
B-UNILM	6.91	39.1	13.1	3.6	1.2	5.01	36.7	11.3	3.0	1.3
R-UNILM	7.35	39.4	13.0	3.7	1.6	5.19	41.0	12.9	3.7	1.6
G-UNILM	11.14	43.5	18.3	7.3	3.8	6.49	41.3	14.9	4.9	2.1
G-UNILM(BT)	11.67	45.3	19.4	8.2	4.4	7.01	42.5	15.7	5.4	2.5
Model	Modern to Ancient (History)					Ancient to Modern (History)				
	BLEU	BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEU	BLEU-1	BLEU-2	BLEU-3	BLEU-4
LSTM‡	15.82	48.5	23.3	14.2	9.2	11.44	35.8	14.3	7.9	4.7
Bi-LSTM‡	18.06	52.6	26.7	16.6	11.0	13.95	41.2	18.6	10.9	6.8
GRU‡	12.40	49.1	21.5	12.3	7.6	10.56	36.4	14.1	7.5	4.3
Bi-GRU‡	15.19	51.2	24.5	14.6	9.4	12.19	42.2	18.3	10.3	6.2
Transformer‡	12.22	48.9	21.4	12.4	7.9	12.02	43.3	18.9	10.5	6.3
LSTM	26.21	63.4	35.9	23.6	16.4	23.59	60.9	34.2	21.5	14.2
Bi-LSTM	31.46	66.2	40.0	27.6	20.0	24.16	60.8	34.1	21.9	14.9
GRU	26.42	62.1	35.0	22.7	15.7	23.46	61.7	34.1	21.6	14.6
Bi-GRU	31.89	66.1	40.0	27.5	19.8	24.29	62.1	35.7	21.9	15.4
Transformer	31.52	65.2	39.5	27.2	19.6	24.25	61.3	34.8	21.8	15.2
B-UNILM	36.18	67.2	42.4	30.0	22.0	27.52	70.0	42.2	28.5	20.1
R-UNILM	36.26	67.3	42.7	30.2	22.2	27.41	70.2	42.3	28.5	20.2
G-UNILM	40.26	71.5	48.3	35.7	27.2	30.08	69.8	42.5	29.4	21.1
G-UNILM(BT)	40.73	72.7	49.7	36.5	27.9	30.52	70.9	43.0	29.8	21.7

Influence of Word Segmentation. Unexpectedly, the use of word segmentation undermines the performance of the models. The BLEU scores of these models are all under 2.0 on Literature, and under 20.0 on History. One possible reason is that word segmentation causes a significant expansion of vocabulary due to the combination of characters. The vocabulary size is around 4,000 before word segmentation, but becomes over 20,000 after word segmentation, causing the problem of data sparseness and model overfitting. Another consequence of word segmentation is the increase number of out of vocabulary (OOV) [19] words. As a result, it is more difficult for the model to predict the correct word in the inference stage. The experimental results in this paper are

Table 3. Imagery Conservation (I.C) scores.

Model	Literature		History	
	m2a	a2m	m2a	a2m
LSTM	58.8	46.7	70.3	73.8
Bi-LSTM	63.2	49.9	71.9	76.5
GRU	60.3	53.7	70.1	76.1
Bi-GRU	61.7	54.9	71.1	76.3
Transformer	68.3	57.6	72.3	77.9
B-UNILM	85.4	68.1	87.1	83.1
R-UNILM	85.1	67.7	84.1	84.4
G-UNILM	86.3	70.0	89.9	86.7

consistent with the conclusion drawn by Li et al. [8], that is, word segmentation is not suitable for Chinese NLP tasks in most cases.

Influence of Back Translation. Back Translation(BT) [3] is applied to the Guwen-UNILM model as a method of data augmentation. Besides the original parallel sentences in the corpora, we created 100k additional parallel sentences by back translation. In Table 2 the results show that BT can improves the original Guwen-UNILM by around 0.5 on BLEU scores on both corpora.

Table 4. Human evaluation scores.

Model	Flu.	Cons.	Mean.	Total
Bi-LSTM	2.56	2.92	2.41	7.89
Bi-GRU	2.67	2.94	2.54	8.15
Transformer	3.02	2.97	2.89	8.88
B-UNILM	3.31	3.40	3.26	9.97
R-UNILM	3.38	3.44	3.27	10.09
G-UNILM	3.45	3.42	3.30	10.17

Human Evaluation Results. We invited five human experts with master’s degree on Chinese linguistics to evaluate the translations generated by 9 different models. For each model, 100 ancient-to-modern translations and 100 modern-to-ancient translations generated by the model are randomly selected for evaluation. In total, we received $5 \times 9 \times 200 = 9000$ evaluation records. The results of human evaluation in Table 4 indicate that the sentences generated by Guwen-UNILM have the highest readability and the model gets the highest score of 10.17 in total.

Table 5. The accuracy of human discrimination test.

Model	Literature		History	
	a2m	m2a	a2m	m2a
Bi-GRU	83.4	79.2	64.8	62.7
Transformer	80.3	76.5	62.5	60.1
G-UNILM	69.8	66.2	57.9	54.2

Human Discrimination Test. Finally, the five experts did the human discrimination test. On both corpora, 200 translations (100 ancient sentences and 100 modern sentences) generated by 3 comparative models (Bi-GRU, Transformer and Guwen-UNILM) were randomly selected and paired with their corresponding ground-truth translations. The experts are requested to distinguish which one of the two sentences is written by human. The evaluation criterion is the accuracy of discrimination, which is the ratio of correctly distinguished sentence pairs to the total number of sentence pairs. As shown in Table 5, discrimination accuracy on Literature corpus is obviously higher than that on History corpus, which indicates that the literature texts translated by machine are more easily recognized by the experts. Among the three models, the translations generated by Guwen-UNILM is more like human translations on both corpora.

5 Conclusion

In this paper, we study a novel task on translation between ancient and modern Chinese and propose a monolingual translation framework based on pre-trained models. The Guwen-BERT is used to adapt to the characteristic of ancient Chinese and then combined with the UNILM framework to complete the translation. For this task, we established two fine-grained parallel corpora. Experiments show that the proposed method significantly outperforms the traditional machine translation methods.

References

- Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) (2018)
- Dong, L., et al.: Unified language model pre-training for natural language understanding and generation. In: Advances in Neural Information Processing Systems, pp. 13063–13075 (2019)
- Edunov, S., Ott, M., Auli, M., Grangier, D.: Understanding back-translation at scale. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 489–500 (2018)
- Freitag, M., Al-Onaizan, Y.: Beam search strategies for neural machine translation. arXiv preprint [arXiv:1702.01806](https://arxiv.org/abs/1702.01806) (2017)
- Galley, M., Manning, C.D.: A simple and effective hierarchical phrase reordering model. In: Conference on Empirical Methods in Natural Language Processing (2008)
- Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. arXiv preprint [arXiv:1705.03122](https://arxiv.org/abs/1705.03122) (2017)

7. Klein, G., Kim, Y., Deng, Y., Senellart, J., Rush, A.: OpenNMT: open-source toolkit for neural machine translation. In: Proceedings of ACL 2017, System Demonstrations (2017)
8. Li, X., Meng, Y., Sun, X., Han, Q., Yuan, A., Li, J.: Is word segmentation necessary for deep learning of Chinese representations? arXiv preprint [arXiv:1905.05526](https://arxiv.org/abs/1905.05526) (2019)
9. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. arXiv preprint [arXiv:1907.11692](https://arxiv.org/abs/1907.11692) (2019)
10. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. arXiv preprint [arXiv:1508.04025](https://arxiv.org/abs/1508.04025) (2015)
11. Marcu, D., Och, F.J.: Statistical phrase-based translation. In: Proceedings of the HLT-NAACL (2003)
12. Nogueira, R., Cho, K.: Passage re-ranking with BERT. arXiv preprint [arXiv:1901.04085](https://arxiv.org/abs/1901.04085) (2019)
13. 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, ACL 2002, pp. 311–318. Association for Computational Linguistics, USA (2002). <https://doi.org/10.3115/1073083.1073135>
14. Souza, F., Nogueira, R., Lotufo, R.: Portuguese named entity recognition using BERT-CRF. arXiv preprint [arXiv:1909.10649](https://arxiv.org/abs/1909.10649) (2019)
15. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, pp. 3104–3112 (2014)
16. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30, pp. 5998–6008 (2017)
17. Vaswani, A., et al.: Attention is all you need. arXiv (2017)
18. Yang, W., et al.: End-to-end open-domain question answering with BERTserini. arXiv preprint [arXiv:1902.01718](https://arxiv.org/abs/1902.01718) (2019)
19. Young, S.R.: Detecting misrecognition and out-of-vocabulary words. In: Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 2, pp. II-21. IEEE (1994)
20. Zhang, Z., Li, W., Su, Q.: Automatic translating between ancient Chinese and contemporary Chinese with limited aligned corpora. In: Tang, J., Kan, M.-Y., Zhao, D., Li, S., Zan, H. (eds.) NLPCC 2019. LNCS (LNAI), vol. 11839, pp. 157–167. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32236-6_13



Adaptive Transformer for Multilingual Neural Machine Translation

Junpeng Liu, Kaiyu Huang, Jiuyi Li, Huan Liu, and Degen Huang^(✉)

School of Computer Science and Technology, Dalian University of Technology,
Dalian, China
`{liujunpeng.nlp,kaiyuhuang,lee.91,liuhuan4221}@mail.dlut.edu.cn,`
`huangdg@dlut.edu.cn`

Abstract. Multilingual neural machine translation (MNMT) with a single encoder-decoder model has attracted much interest due to its simple deployment and low training cost. However, the all-shared translation model often yields degraded performance due to the modeling capacity limitations and language diversity. Moreover, it has been revealed in recent studies that the shared parameters lead to negative language interference although they may also facilitate knowledge transfer across languages. In this work, we propose an adaptive architecture for multilingual modeling, which divides the parameters in MNMT sub-layers into shared and language-specific ones. We train the model to learn and balance the shared and unique features with different degrees of parameter sharing. We evaluate our model on one-to-many and many-to-one translation tasks. Experiments on IWSLT dataset show that our proposed model remarkably outperforms the multilingual baseline model and achieves comparable or even better performance compared with the bilingual model.

Keywords: Multilingual neural machine translation · Adaptive transformer · Language-specific modeling

1 Introduction

Multilingual neural machine translation (MNMT) leverages a single encoder-decoder model for translations in multiple language pairs [1, 10, 11, 13]. MNMT is appealing since it greatly simplifies the large-scale deployment of translation models and reduces the maintenance cost [6, 11]. Moreover, the multilingual model tends to yield better translations for low-resource and zero-shot language pairs because the exposure to various languages facilitates the potential knowledge transfer among languages [2, 9, 27].

However, a big challenge to this highly compact model is the curse of multilinguality, which degrades the performance across languages due to the language interference [5]. It has been revealed that adding language-specific capacity into the multilingual model can mitigate the negative interference [23]. To this end,

previous works [4, 8, 16, 22] has mostly focused on model architecture design by reorganizing parameter sharing in the MNMT model, aiming to achieve a better transfer-interference trade-off. Recent language adapters attract much attention for their lightweight architectures and prominent improvements in multilingual translation, which are inserted in-between two different sub-layers in MNMT model [3, 24, 25]. Although those studies differ in model architectures, they adopt similar parameter-sharing strategies which share either all or no parameters of a MNMT sub-component (e.g. attention or feed-forward sub-layer) across languages, leaving the impacts of inner parameters under-studied.

In this work, we explore more fine-grained parameter sharing strategies with an adaptive model, which adds language-specific capacity into each sub-layer in MNMT. We divide the weight matrices in each sub-layer into two parts: shared and language-specific parameters. Each language maintains those two types of parameters so that its specific characteristics can be captured and incorporated into each sub-layer besides the shared information. The adaptive model is trained to seek a balance between sharing and not sharing by altering the dimension of the two parameters.

We evaluate our proposed method on IWSLT dataset, with models building on the Transformer architecture [19]. Following [24], we use target-specific and source-specific modeling for one-to-many (O2M) and many-to-one (M2O) translation, respectively. Experimental results demonstrate that our method outperforms the standard multilingual baseline model and achieves comparable or better translation quality compared with the bilingual model. To sum up, the contributions of this work are as follows:

- We propose a fine-grained parameter sharing strategy, which adds language-specific capacity to Transformer sub-layers and trains the model to balance the shared and unique features for a better transfer-interference trade-off.
- We apply the proposed strategies to attention and feed-forward sub-layer, and obtain remarkable improvements on translation quality. Extensive experiments show that our parameter sharing strategies in different sub-layers are complementary and the combined model outperforms the separate ones.
- We optimize the degree of parameter sharing in different sub-layers and achieve best the translation performance on O2M/M2O translation task with the improvements of 1.41/0.93 BLEU over the multilingual baseline model.

2 Related Work

Our work is related to language-specific modeling for MNMT in general. Early researches in MNMT focused on the use of shared sub-components (i.e. encoder, decoder or attention mechanism) to encourage knowledge transfer, which included using a shared encoder for all source languages on one-to-many translation [7], sharing the attention mechanism across multiple languages in many-to-many translation scenario [8] and employing a shared decoder for all target languages on many-to-one translation [26]. However, the need of using

an additional sub-component for each language makes those networks complicated and expensive to train. Johnson et al. [11] developed a universal MNMT system which shares all sub-components across all languages by introducing an artificial token to the source text to indicate the target language. Remarkably, this paradigm has greatly simplified the deployment of MNMT by eliminating the use of separate sub-components for each new language pair. However, it is extremely burdensome to process all translations within a single encoder-decoder model and thus results in underperformance compared to the bilingual model. Therefore, subsequent studies turn to explore language-specific modeling to seek a balance between sharing and not sharing, ranging from introducing language-dependant positional embedding and representation [21], redesigning parameter sharing [4, 16, 20, 22], separately modeling languages for different language clusters [18], employing adapters for fine-tuning [3], devising language-aware normalization and transformation [25], and dynamically scheduling language-specific capacity for each token in MNMT [24].

Our work continues in this direction but differs from those aforementioned methods. We propose a more fine-grained parameter sharing strategy which directly introduces language-specific parameters to different sub-components in MNMT model.

3 Background

In this section, we briefly review the multilingual translation approach and the Transformer model. To perform multilingual translation, Johnson et al. [11] reuse the standard bilingual translation model by adding a pretending token to each source sentence to specify the target language. Specifically, given a source sentence $X' = \{x_1, x_2, \dots, x_m\}$ and the language token $lang$, the source input for MNMT is changed to $X = \{lang, x_1, x_2, \dots, x_m\}$. Following Zhang et al. [24], the language token $lang$ is altered to denote the target language in O2M translation but source language in M2O translation in this paper.

Transformer stacks several identical layers in both the encoder and the decoder, which consist of multi-head attention and feed-forward sub-layers. Multi-head attention employs h attention heads to jointly attend to information from different representation sub-spaces and concatenates the results from all the attention heads. Each head keeps a set of query (Q), key (K) and value (V) for all the input tokens. For a given token x_i ($x_i \in \mathbb{R}^{d_m}$), a head assigns its attention to a sequence of tokens using query-key compatibility function between linearly transformed input tokens and gets the attention score α_{ij} with a softmax function:

$$e_{ij} = \frac{(h_i^q)(h_j^k)^T}{\sqrt{d_m}} = \frac{(x_i W_q)(x_j W_k)^T}{\sqrt{d_m}} \quad (1)$$

$$\alpha_{ij} = \frac{\exp e_{ij}}{\sum_{k=1}^n \exp e_{ik}} \quad (2)$$

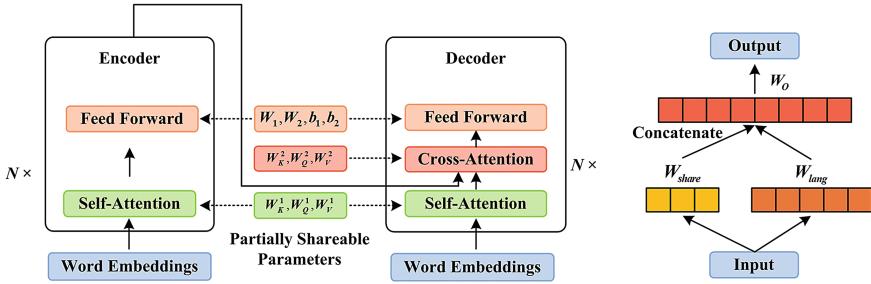


Fig. 1. The illustration of proposed adaptive model for multilingual translation based on partial parameter sharing strategies in different transformer sub-layers. **Left:** Transformer model and partially shareable parameters. **Right:** Description of partially sharing strategy.

where d_m is model hidden size. The final output z_i ($z_i \in \mathbb{R}^{d_m}$) is computed as the weighted sum of linearly transformed value vectors of all input tokens.

$$z_i = \sum_{j=1}^n \alpha_{ij} h_j^v = \sum_{j=1}^n \alpha_{ij} (x_j W_v) \quad (3)$$

$W_k, W_q, W_v \in \mathbb{R}^{d_m \times d_m}$ are linear projection parameters of the multi-head attention sub-layer. Those parameters are unique per head and layer.

The output of multi-head attention sub-layer z_i is then fed into the feed-forward network, which consists of two linear transformations with a ReLU activation in between:

$$w_i = \text{FFN}(z_i) = \max(0, z_i W_1 + b_1) W_2 + b_2 \quad (4)$$

where $W_1 \in \mathbb{R}^{d_m \times d_f}$, $W_2 \in \mathbb{R}^{d_f \times d_m}$, $b_1 \in \mathbb{R}^{d_f}$ and $b_2 \in \mathbb{R}^{d_m}$ are trainable parameters, and d_f is the feed-forward hidden dimension with $d_f > d_m$.

4 Proposed Method

In this paper, we propose an adaptive transformer model to improve multilingual translation quality by incorporating language-specific parameters into the vanilla transformer-based MNMT model. We start with our general strategies for redesigning the parameter sharing in the MNMT model, and then introduce the implementations in different Transformer sub-layers including multi-head attention and feed-forward network.

4.1 Adaptive Transformer

In the universal multilingual translation model, all the sub-layer parameters are shared across languages. It has been shown that the language signals from the

language embeddings alone are insufficient [2]. Therefore, the sub-layers should embody some language-specific information to be more flexible to handle diverse languages. Partially inspired by the idea of using a fixed mix of shared and unique hidden units [21], we incorporate language-specific information into Transformer sub-layers by dividing each linear projection parameter in Fig. 1 into two parts: shared and language-specific parameters. On the one hand, the shared parameters are capable of learning the commonality of languages, which can be beneficial to the potential knowledge transfer across languages. On the other hand, the language-specific parameters are able to increase the modeling capacity for each language to capture its unique characteristics.

On each of these shared and language-specific parameters, we perform linear projections in parallel, yielding the shared and the language-specific representations. They are then concatenated and once again projected, resulting in the final output as illustrated in Fig. 1. By controlling the dimension of the projection parameters W_{share} and W_{lang} , we can alter the degree of parameter sharing and achieve a better transfer-interference trade-off. Based on this parameter sharing mechanism, we reformulate the attention and feed-forward modules in Transformer, respectively.

4.2 Adaptive Attention Layer

Instead of performing a single attention function, Transformer employs multi-head attention mechanism which splits the d_m -dimensional representation into h tensors with the same dimension. However, it is shown by previous studies that some attention heads are redundant [14]. We propose an adaptive attention model which replaces some shared attention parameters with language-specific ones, not only reducing the redundancy in the multi-head attention model but improving the ability to learn the difference between languages. Given an input token x_i , the adaptive attention model first divides the original linear projection into shared and specific parts to get their corresponding representations via separate linear transformations:

$$h_i^{share} = x_i W^{share} \quad (5)$$

$$h_i^{lang} = x_i W^{lang} \quad (6)$$

where $W^{share} \in \mathbb{R}^{d_m \times \alpha d_m}$ is the weight matrix shared across languages, while $W^{lang} \in \mathbb{R}^{d_m \times (1-\alpha)d_m}$ is only used for modeling language $lang$ which endows MNMT with language-specific capacity. $\alpha \in (0, 1)$ is the coefficient which decides the rate of shared parameters.

The adaptive attention model then concatenates the shared representation h_i^{share} and language-specific representation h_i^{lang} and outputs the mix representation through another parameterized linear transformation:

$$h_i = [h_i^{share}; h_i^{lang}] W_o \quad (7)$$

where $[; ;]$ represents concatenation, $W_o \in \mathbb{R}^{d_m \times d_m}$ is the final linear projection parameter. Compared to separately initializing attention heads for shared

or varied representations, our adaptive attention model fuses those two representations before splitting into different heads so that each head is capable of capturing the shared and language-specific characteristics. We apply these formulas to the calculation of h_i^q , h_i^k and h_i^v .

4.3 Adaptive Feed-Forward Layer

The feed-forward network transforms input representations by first increasing the hidden dimension and then projecting it back to the original input dimension. We argue that the intermediate hidden state also has redundancy because it is over-parameterized with a dimension larger than the input dimension. Therefore, we propose to make part of the intermediate hidden state language-specific.

For an input token $z_i \in \mathbb{R}^{d_m}$, we first employ two independent feed-forward networks to separately perform transformations, yielding the shared and language-specific representations:

$$w_i^{share} = \text{FFN}_{share}(z_i) = \max(0, z_i W_1^{share} + b_1^{share}) W_2^{share} + b_2^{share} \quad (8)$$

$$w_i^{lang} = \text{FFN}_{lang}(z_i) = \max(0, z_i W_1^{lang} + b_1^{lang}) W_2^{lang} + b_2^{lang} \quad (9)$$

where $W_1^{share} \in \mathbb{R}^{d_m \times \beta d_f}$, $b_1^{share} \in \mathbb{R}^{\beta d_f}$, $W_2^{share} \in \mathbb{R}^{\beta d_f \times \beta d_m}$, $b_2^{share} \in \mathbb{R}^{\beta d_m}$ are shared projection parameters, while $W_1^{lang} \in \mathbb{R}^{d_m \times (1-\beta)d_f}$, $b_1^{lang} \in \mathbb{R}^{(1-\beta)d_f}$, $W_2^{lang} \in \mathbb{R}^{(1-\beta)d_f \times (1-\beta)d_m}$, $b_2^{lang} \in \mathbb{R}^{(1-\beta)d_m}$ are language-specific ones. $\beta \in (0, 1)$ is a scaling parameter similar to α , which can be tuned based on translation task, thus allowing us to control the degree of parameter sharing. Then we combine the two representations and convert their concatenation back to the original dimension.

$$w_i = [w_i^{share}; w_i^{lang}] W_o \quad (10)$$

where $W_o \in \mathbb{R}^{d_m \times d_m}$ is a linear transformation matrix.

5 Experiments

5.1 Dataset

We conduct our experiments on IWSLT dataset for both O2M and M2O translation tasks. The corpus is collected from IWSLT evaluation campaign¹ from year 2011 to 2018, which consists of 16 languages \leftrightarrow English translation pairs. Table 1 shows the statistics of the train/validation/test set. We up-sample each language in the training set to be roughly the same. All the sentences are first tokenized with moses tokenizer² and then segmented into sub-words using byte pair encoding (BPE) algorithm [17]. We learn a joint vocabulary by performing BPE with merge operations of 64K. We randomly shuffle the training set to mix the instances of different language pairs.

¹ <https://wit3.fbk.eu/>.

² <https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl>.

Table 1. The statistics of train, validation and test data for IWSLT dataset.

Language pair	Train	Valid	Test	Language pair	Train	Valid	Test
En-Ar	223K	887	1569	En-Nl	167K	887	1569
En-Cs	114K	480	1511	En-Pl	176K	767	1564
En-De	196K	887	1565	En-Ro	181K	887	1567
En-Es	180K	887	1570	En-Ru	177K	887	1568
En-Fr	219K	887	1664	En-Sl	17K	1144	1411
En-He	184K	888	1568	En-Tr	154K	887	1568
En-It	181K	887	1529	En-Vi	129K	768	1342
En-Ja	221K	871	1549	En-Zh	208K	887	1570

5.2 Model Configurations

We build our models based on Transformer [19] and adopt the same model configurations as Tan [18]. Specifically, the layer number is 2, the model hidden size d_m is 256, and the feed-forward hidden size d_f is 1024. The size of language embeddings is also set to 256. For training, we optimize all parameters with Adam optimizer [12] ($\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-9}$) with label smoothing of 0.1 and learning rate is scheduled as the inverse square root of training step with a warm-up step of 4K. We apply dropout to attention weights and residual layers with a rate of 0.1. Each training batch contains roughly 8192 tokens and the training sequence length is limited to 100. We set the maximum training step to 150K. During inference, we use beam search decoding with beam size of 5 and length penalty of 0.6. The BLEU score is measured by the de-tokenized case-sensitive SacreBLEU [15].³ All our experiments are conducted on a single Nvidia GeForce RTX 3090 GPU.

5.3 Main Results

Evaluation results on O2M translation task and M2O translation task are presented in Tables 2 and 3, respectively. We have the following observations: First, the Multilingual Baseline model slightly outperforms the Bilingual Baseline model due to the remarkable improvements on the low-resource language pair (+8.02 BLEU for En → Sl and +12.06 BLEU for Sl → En), showing that the low-resource languages benefit more knowledge transfer than the high-resource languages in the MNMT model. However, the Multilingual Baseline model yields degraded translations on over half of the language pairs for both O2M and M2O translation tasks, demonstrating that the negative language interference occurs in the all-shared translation model. Second, for O2M translation task, Adaptive Transformer model achieves the best performance which outperforms the Multilingual Baseline model by 1.19 BLEU score on average and succeeds on 15

³ Signature: BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.4.14.

Table 2. BLEU scores of English → 16 languages on the IWSLT dataset. The BLEU scores in **bold** represent the best performance across all methods. Δ represents the improvements of our Adaptive Transformer model over the Multilingual Baseline model.

Language	Bilingual baseline	Multilingual baseline	Adaptive attention	Adaptive feed-forward	Adaptive transformer	Δ
En → Ar	12.49	10.12	10.96	11.64	11.75	+1.63
En → Cs	13.76	14.72	15.29	15.94	15.89	+1.17
En → De	25.29	23.10	24.71	25.43	25.74	+2.64
En → Es	34.99	33.73	35.02	35.63	35.39	+1.66
En → Fr	30.05	32.21	33.65	34.12	34.42	+2.21
En → He	22.52	19.86	21.21	21.81	21.45	+1.59
En → It	26.19	25.73	26.28	26.77	26.86	+1.13
En → Ja	8.88	8.77	9.27	9.55	9.77	+1.00
En → Nl	27.57	26.34	26.77	27.77	28.15	+1.81
En → Pl	10.33	10.77	11.62	11.72	11.61	+0.84
En → Ro	22.82	22.78	23.28	23.86	24.18	+1.40
En → Ru	12.78	14.22	14.80	15.21	15.02	+0.80
En → Sl	4.79	12.81	11.10	12.02	11.04	-1.77
En → Tr	9.41	9.53	9.90	10.22	10.66	+1.13
En → Vi	25.94	27.79	28.73	28.96	28.64	+0.85
En → Zh	16.36	15.59	16.24	16.36	16.53	+0.94
AVG.	19.01	19.25	19.93	20.44	20.44	+1.19

language pairs. Furthermore, compared with the Bilingual Baseline model, Adaptive Transformer model also performs better in most cases (14 out of 16 cases). Third, as for M2O translation task, Adaptive Transformer model also scores the best average BLEU score with 0.93 BLEU gains over the Multilingual Baseline model. It outperforms the Multilingual Baseline model and Bilingual Baseline model on 14 and 10 language pairs, respectively. Fourth, O2M benefits more from the adaptive architecture than M2O because O2M has to translate multiple languages in the decoder, thus demanding more language-specific capacity to handle diverse typological features. By contrast, M2O shares the same target language so that information can be easily transferred. Fifth, it is worth noting that Adaptive Transformer model yields inferior performance to the Multilingual Baseline model on low-resource translation in both translation scenarios ($Sl \leftrightarrow En$). We leave the analysis of this problem to Sect. 5.6.

5.4 Ablation Study

We compare and discuss the adaptive architecture in attention and feed-forward sub-layer separately. As shown in Tables 2 and 3, the adaptive architecture in

Table 3. BLEU scores of 16 languages → English on the IWSLT dataset. The BLEU scores in **bold** represent the best performance across all methods. Δ represents the improvements of our Adaptive Transformer model over the Multilingual Baseline model.

Language	Bilingual baseline	Multilingual baseline	Adaptive attention	Adaptive feed-forward	Adaptive transformer	Δ
Ar → En	26.07	23.63	25.17	24.24	25.28	+1.65
Cs → En	20.00	21.38	22.16	22.12	22.12	+0.74
De → En	28.06	26.30	27.36	27.15	27.56	+1.26
Es → En	38.33	35.97	37.32	37.14	37.91	+1.94
Fr → En	30.00	29.78	30.87	30.96	31.58	+1.80
He → En	30.28	27.58	28.86	28.58	29.23	+1.65
It → En	28.47	27.51	28.66	27.97	28.56	+1.05
Ja → En	8.81	8.08	7.76	8.12	8.06	-0.02
Nl → En	31.76	29.69	31.15	31.38	31.67	+1.98
Pl → En	15.70	16.43	16.72	16.62	17.49	+0.43
Ro → En	29.97	28.35	29.73	29.19	30.41	+1.66
Ru → En	16.50	17.68	18.02	17.54	18.55	+0.43
Sl → En	7.14	19.20	17.56	17.36	16.04	-3.16
Tr → En	16.90	16.21	17.21	17.00	17.80	+1.59
Vi → En	22.29	23.10	24.00	23.84	24.23	+1.13
Zh → En	12.90	12.57	13.00	13.26	13.35	+0.78
AVG.	22.70	22.72	23.46	23.28	23.65	+0.93

each sub-layer alone yields better translation quality in comparison to the Multilingual Baseline model in both translation directions, indicating that the language interference exists in all the sub-layers and language-specific capacity is necessary. Specifically, for O2M translation, the Adaptive Feed-forward model outperforms the Adaptive Attention model and achieves even the same accuracy as the Adaptive Transformer model with respect to the average BLEU score. However, it performs worse than the Adaptive Transformer model on 10 out of 16 languages. As for M2O translation, the Adaptive Transformer model apparently outperforms other models and the gain brought by the Adaptive Attention model is smaller compared with the Adaptive Feed-forward model.

5.5 Analysis on Shared Rate

In this section, we study the question that how many parameters should be shared and how many parameters should be language-specific in different sub-layers. We conduct a series of experiments to investigate different settings. The results are presented in Fig. 2. We can observe that the Adaptive Transformer model achieves best performance when we share 50% and 75% of the linear

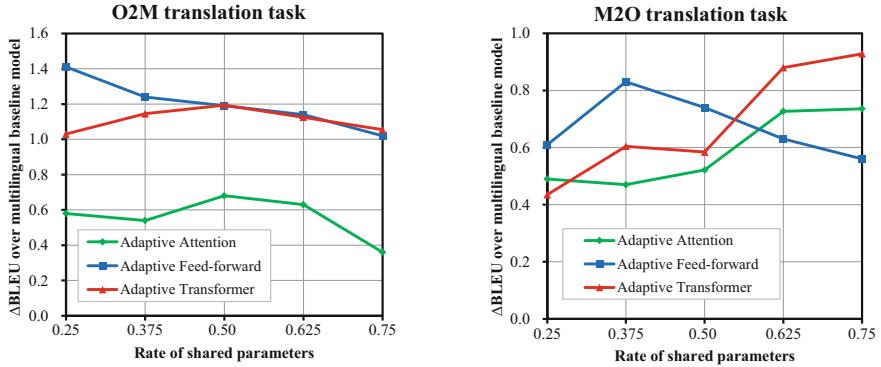


Fig. 2. The comparison of different shared rates for Adaptive Attention, Feed-forward and Transformer model under O2M and M2O translation scenarios.

projection parameters for O2M and M2O translation tasks, respectively. For the Adaptive Attention model, the best performance comes when we share one-half of parameters for O2M translation while three quarters of parameters for M2O translation. As for the Adaptive Feed-forward model, it obtains the best translation results when we share only 25% and 37.5% of matrix parameters on O2M and M2O translation, respectively. In particular, the Adaptive Feed-forward model also achieves the highest improvement across all the three adaptive models on O2M translation, which outperforms the Multilingual Baseline model by 1.40 BLEU score. Comparing the two translation tasks, we find that O2M has to employ more language-specific parameters in order to obtain the best performance than M2O, suggesting that O2M needs more language-specific capacity to handle the translation of languages with diverse features.

5.6 Analysis on Low-Resource Language

As shown in Tables 2 and 3, we find that the low-resource language translation ($\text{Sl} \leftrightarrow \text{En}$) degrades when we incorporate language-specific parameters into Transformer sub-layers (-1.77 BLEU for $\text{En} \rightarrow \text{Sl}$ and -3.16 BLEU for $\text{Sl} \rightarrow \text{En}$ with Adaptive Transformer model). In this section, we conduct experiments to figure out how language-specific capacity affects the translation quality of low-resource language pairs. Figure 3 reports the results. In general, the three adaptive models underperform the Multilingual Baseline models on both O2M and M2O translation tasks under almost all parameter settings. The only exception is the Adaptive Feed-forward model, which yields better translation accuracy than the Multilingual Baseline model in O2M task with the shared rate of 75%. As we increase the rate of shared parameters, the performance gap between the adaptive models and the multilingual baselines gradually decreases. We conjecture that the introduction of language-specific parameters discourages the knowledge transfer from high-resource languages to low-resource ones, leading to the worse performance.

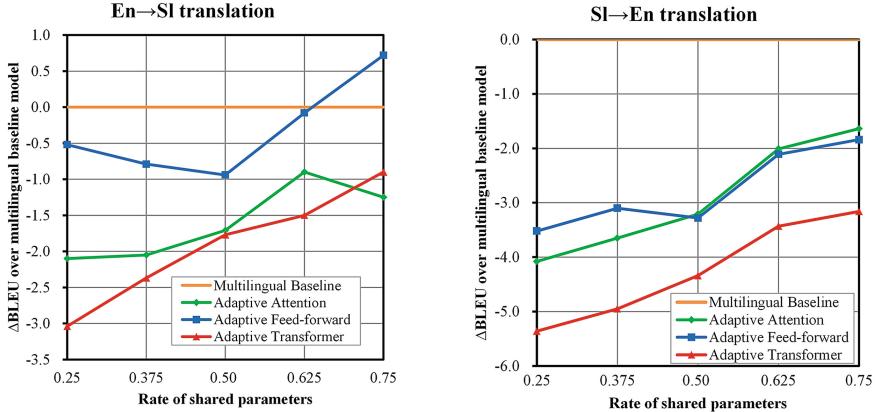


Fig. 3. The translation results of low-resource language pair ($\text{Sl} \leftrightarrow \text{En}$) for different models when varying the shared rates.

6 Conclusion and Future Work

In this work, we propose an adaptive transformer to tackle the language inference in MNMT. We divide the parameters in attention and feed-forward sub-layers into shared and language-specific ones to capture the shared features across all the languages and unique characteristics for each language. We optimize the degree of parameter sharing in order to achieve a better transfer-interference trade-off. Experiments demonstrate that our method can outperform the multilingual baseline model on both O2M and M2O translations, and achieves comparable or even better accuracy compared with the bilingual translation model.

For future work, we will extend our method to massively multilingual translation tasks. We will also explore new strategies to enhance the freedom of MNMT in performing flexible parameter sharing.

Acknowledgments. The research work has been supported by the National Key Research and Development Program of China No. 2020AAA0108004 and the Natural Science Foundation of China under Grant No. U1936109.

References

1. Aharoni, R., Johnson, M., Firat, O.: Massively multilingual neural machine translation. In: ACL 2019 (2019)
2. Arivazhagan, N., et al.: Massively multilingual neural machine translation in the wild: findings and challenges. arXiv preprint [arXiv:1907.05019](https://arxiv.org/abs/1907.05019) (2019)
3. Bapna, A., Firat, O.: Simple, scalable adaptation for neural machine translation. In: EMNLP-IJCNLP 2019 (2019)
4. Blackwood, G., Ballesteros, M., Ward, T.: Multilingual neural machine translation with task-specific attention. In: COLING 2018 (2018)

5. Conneau, A., et al.: Unsupervised cross-lingual representation learning at scale. In: ACL 2020 (2020)
6. Dabre, R., Chu, C., Kunchukuttan, A.: A survey of multilingual neural machine translation. ACM Comput. Surv. (CSUR) **53**(5), 1–38 (2020)
7. Dong, D., Wu, H., He, W., Yu, D., Wang, H.: Multi-task learning for multiple language translation. In: ACL-IJCNLP 2015 (2015)
8. Firat, O., Cho, K., Bengio, Y.: Multi-way, multilingual neural machine translation with a shared attention mechanism. In: NAACL 2016 (2016)
9. Firat, O., Sankaran, B., Al-onaizan, Y., Yarman Vural, F.T., Cho, K.: Zero-resource translation with multi-lingual neural machine translation. In: EMNLP 2016 (2016)
10. Ha, T.L., Niehues, J., Waibel, A.: Toward multilingual neural machine translation with universal encoder and decoder. In: Proceedings of IWSLT 2016 (2016)
11. Johnson, M., et al.: Google’s multilingual neural machine translation system: enabling zero-shot translation. Trans. Assoc. Comput. Linguist. **5**, 339–351 (2017)
12. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
13. Lu, Y., Keung, P., Ladhak, F., Bhardwaj, V., Zhang, S., Sun, J.: A neural interlingua for multilingual machine translation. In: Proceedings of the Third Conference on Machine Translation: Research Papers (2018). <https://doi.org/10.18653/v1/W18-6309>
14. Michel, P., Levy, O., Neubig, G.: Are sixteen heads really better than one? arXiv preprint [arXiv:1905.10650](https://arxiv.org/abs/1905.10650) (2019)
15. Post, M.: A call for clarity in reporting BLEU scores. In: Proceedings of the Third Conference on Machine Translation: Research Papers (2018). <https://doi.org/10.18653/v1/W18-6319>
16. Sachan, D., Neubig, G.: Parameter sharing methods for multilingual self-attentional translation models. In: Proceedings of the Third Conference on Machine Translation: Research Papers (2018). <https://doi.org/10.18653/v1/W18-6327>
17. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: ACL 2016 (2016)
18. Tan, X., Chen, J., He, D., Xia, Y., Qin, T., Liu, T.Y.: Multilingual neural machine translation with language clustering. In: EMNLP-IJCNLP 2019 (2019)
19. Vaswani, A., et al.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 6000–6010 (2017)
20. Vázquez, R., Raganato, A., Tiedemann, J., Creutz, M.: Multilingual NMT with a language-independent attention bridge. In: Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019) (2019)
21. Wang, Y., Zhang, J., Zhai, F., Xu, J., Zong, C.: Three strategies to improve one-to-many multilingual translation. In: EMNLP 2018 (2018)
22. Wang, Y., Zhou, L., Zhang, J., Zhai, F., Xu, J., Zong, C.: A compact and language-sensitive multilingual translation method. In: ACL 2019 (2019)
23. Wang, Z., Lipton, Z.C., Tsvetkov, Y.: On negative interference in multilingual models: findings and a meta-learning treatment. In: EMNLP 2020 (2020)
24. Zhang, B., Bapna, A., Sennrich, R., Firat, O.: Share or not? Learning to schedule language-specific capacity for multilingual translation. In: ICLR 2021 (2021)
25. Zhang, B., Williams, P., Titov, I., Sennrich, R.: Improving massively multilingual neural machine translation and zero-shot translation. In: ACL 2020 (2020)
26. Zoph, B., Knight, K.: Multi-source neural translation. In: NAACL 2016 (2016)
27. Zoph, B., Yuret, D., May, J., Knight, K.: Transfer learning for low-resource neural machine translation. In: EMNLP 2016 (2016)



Improving Non-autoregressive Machine Translation with Soft-Masking

Shuheng Wang¹, Shumin Shi^{2(✉)}, and Heyan Huang²

¹ School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, Jiangsu, China
wsh@njust.edu.cn

² School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China
{bjssm, hhy63}@bit.edu.cn

Abstract. In recent years, non-autoregressive machine translation has achieved great success due to its promising inference speedup. Non-autoregressive machine translation reduces the decoding latency by generating the target words in single-pass. However, there is a considerable gap in the accuracy between non-autoregressive machine translation and autoregressive machine translation. Because it removes the dependencies between the target words, non-autoregressive machine translation tends to generate repetitive words or wrong words, and these repetitive or wrong words lead to low performance. In this paper, we introduce a soft-masking method to alleviate this issue. Specifically, we introduce an autoregressive discriminator, which will output the probabilities hinting which embeddings are correct. Then according to the probabilities, we add mask on the copied representations, which enables the model to consider which words are easy to be predicted. We evaluated our method on three benchmarks, including WMT14 EN → DE, WMT16 EN → RO, and IWSLT14 DE → EN. The experimental results demonstrate that our method can outperform the baseline by a large margin with a bit of speed sacrifice.

Keywords: Non-autoregressive · Machine translation · Soft-masking

1 Introduction

Neural machine translation (NMT) has achieved tremendous success in recent years [1, 2, 18]. Generally, NMT models utilize the encoder-decoder framework [1]. Given an input sentence in source language, the encoder maps the sentence into the hidden representations, and the decoder generates the target words in an autoregressive manner from left to right [1, 18]. With this autoregressive decoding strategy, when NMT models generate the current target word, the previous predicted words must be fed into the decoder. In this way, the decoder can learn the target dependencies well, and the accuracy of NMT models has achieved human parity. Despite their success, NMT models suffer from the high

decoding latency, and this has been the bottleneck to apply NMT models in the actual scenario. The main reason for low decoding speed is the autoregressive decoding strategy.

To reduce the decoding latency, non-autoregressive machine translation (NAT) has been introduced in recent years [4]. By removing the dependencies between target words, NAT models [4, 16, 19] can generate the target words simultaneously and break the bottleneck of the autoregressive NMT (AT) models. Instead of feeding previous generated target words into the decoder, NAT models use the copied source representations as the input. In this way, the decoding process is no longer dominated by autoregressive decoding strategy, which enables parallel computation of the decoding process in NAT models.

Although NAT models significantly accelerate the inference process, the accuracy of NAT models falls behind AT models. It is mainly caused by multimodality problem [4]. NAT models remove the dependencies between the target words, and thus the target words may be chosen from multiple feasible translations, leading to repetitive or wrong words. Several iterative refinement methods [3, 8, 20] have been introduced to capture the target dependencies and keep parallel decoding. However, it has been proven that iterative NAT models may lose the advantage compared with careful layer allocation of AT models [6]. The opposite is that fully NAT models still keep the speed advantage. Several works [14, 19] have been proposed to improve the performance of fully NAT models. However, how to better improve the performance of fully NAT models calls for more exploration.

In this work, we propose a novel NAT model, which utilizes a simple autoregressive discriminator to capture the target sequence information and to determine which input embeddings are similar to its corresponding target embeddings. Specifically, we introduce a single-layer autoregressive discriminator before the decoder blocks. And according to the probabilities predicted by the discriminator, we add mask on the input embeddings. The mask will let the decoder know which copied embeddings may be correct or not. Meantime, with mask, the noise is also introduced for correct copied embeddings, which potentially improve the robustness of NAT models.

To evaluate the performance of our method, we conduct experiments on three benchmark tasks, including WMT14 EN → DE, WMT16 EN → RO, and IWSLT14 DE → EN. And experimental results demonstrate that our proposed method outperforms the baseline and narrows the gap between NAT model and AT model.

2 Background

NMT is proposed to generate the sentence $Y = \{y_1, y_2, \dots, y_m\}$ in target language based on the sentence $X = \{x_1, x_2, \dots, x_n\}$ in source language. According to different decoding strategies, NMT can be divided into two classes: non-autoregressive machine translation (NAT) and autoregressive machine translation (AT).

2.1 Autoregressive Machine Translation

At present, autoregressive decoding strategy is still a major method in NMT. For example, given a source sentence X , an AT model learns the distribution of the sentence Y by modeling the target sentence as a chain of conditional probabilities of words:

$$P_{AT} = \prod_{t=1}^M p(y_t | y_{<t}, X; \theta) \quad (1)$$

$y_{<t}$ denotes the previous predicted target words. And $y_{<t}$ provides a part of target information for the decoder. With the part of target information, the decoder can predict the next word better. However, it also limits the possibility of parallel decoding, which has been the bottleneck to apply NMT in real-world translation.

2.2 Non-autoregressive Machine Translation

To break the bottleneck and make parallel decoding possible, a non-autoregressive decoding strategy has been introduced [4]. Instead of modeling the distribution of target sentence as a chain of conditional probabilities, NAT removes the conditional dependencies between the target words, which can significantly speed up inference stage [4, 16, 19]. A NAT model models the distribution of target sentence as follows:

$$P_{NAT} = \prod_{t=1}^M p(y_t | X; \theta) \quad (2)$$

As denoted by Eq. 2, the generation of target word y_t is only dependent on the source sentence. Thus, the model can generate all target words by applying argmax to every time step, which makes parallel decoding possible. However, this method of acceleration presents a new problem, i.e. *multi-modality problem* [4]. Because the generation of target words is independent, it usually leads to duplicated or wrong words. These duplicated and wrong words significantly reduce the performance of NAT models. In this work, we aim to make the model aware which words are difficult to predict.

3 Method

This section will describe our proposed method in detail, which aims to alleviate the multi-modality problem in NAT models. The architecture of our proposal is shown in Fig. 1. As shown in Fig. 1, our proposed model consists of three parts: encoder, decoder, and discriminator.

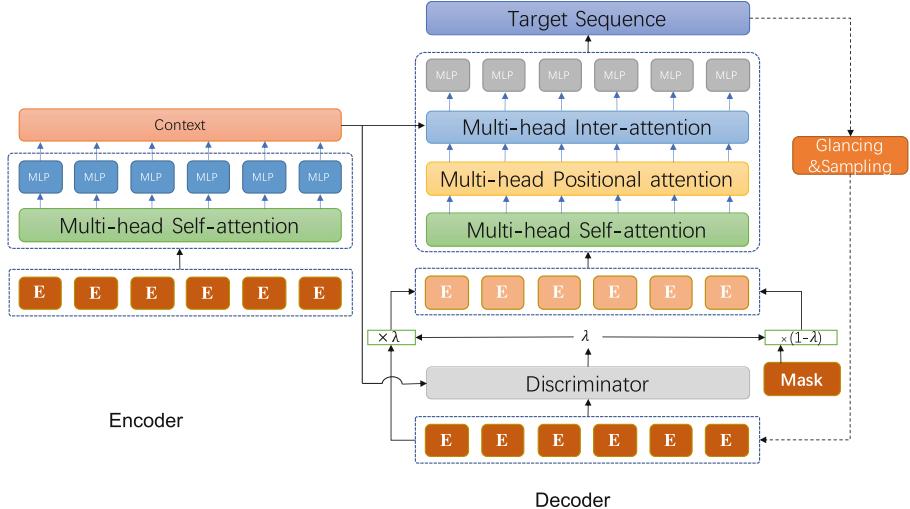


Fig. 1. The architecture of the proposed model. The decoder input is formed by a weighted addition of copied embedding and the embedding of “MASK” according to the probabilities predicted by the discriminator. “MLP” in the model denotes the feed-forward network. “E” denotes the embedding of word.

3.1 Encoder

Following the previous works [4, 14, 18], we utilize a stack of identical Transformer blocks as the encoder. Given the sentence X in source language, the encoder maps it into contextual representations $H = \{h_1, h_2, \dots, h_n\}$. Then the representations H are fed into the decoder.

3.2 Decoder

The decoder in our model also consists of 6 identical Transformer blocks. And to use the representations of source sentence, an inter-attention module is introduced. Moreover, we use a bidirectional mask to remove the dependencies between the target words. The non-autoregressive property of the decoder mainly lies on such bidirectional mask. However, compared to autoregressive decoder using time-shift target words as decoder inputs, the first issue, how long the target sentence will be, is faced by non-autoregressive decoder. In this work, we use the mean of the source representations H to predict the length of the target sentence. And following the previous works [4], we use *Uniform Copy* to form the inputs.

3.3 Discriminator

As described in the previous section, the decoder inputs are copied from the source representations. So the decoder inputs to vanilla decoder can not reflect

the semantic information of the target words. The target words are entirely learned by the inter-attention module. In this case, we introduce a discriminator¹ to determine how different the decoder inputs are from the target words. And if the copied embeddings are identical with the target embeddings, the decoder can use such copied embeddings as part of the target information.

Before the decoder inputs are fed into the decoder, we firstly feed the decoder inputs into the discriminator. And the discriminator in our model will predict the similarity λ , which denotes the differences between the target words and the inputs. Thus, this process can be written as follows:

$$P_d = \prod_{t=1}^M p(\lambda_{\tilde{y}_t} | X, \tilde{y}_{<t}), \quad (3)$$

where, $\tilde{y}_{<t}$ denotes the decoder inputs copied from the source before the position t . When the copied embedding from source is identical to the embedding of target word at position t , the similarity λ will be equal to 1.

So, in our model, the similarity predicted by the discriminator denotes how different the embedding copied from the source is from the embedding of the target word at position t . And based on the similarity, we add the embedding of a special token “[MASK]” on the copied embeddings according to Eq. 4. These additional embeddings will make the decoder aware which words are easy to translate. We name this method Soft-Masking.

$$\mathbb{E} = \lambda_{\tilde{y}_t} * \mathbb{E}(\tilde{y}_t) + (1 - \lambda_{\tilde{y}_t}) * \mathbb{E}([MASK]) \quad (4)$$

3.4 Glancing Training

We introduce a discriminator to detect the difference between the copied embeddings and the target embeddings. While it is unknown which embeddings are correct, the discriminator cannot learn that correctly without the right signals. So, we utilize a *reference glancing* technique [14] to train our model.

With glancing, our model performs the two-pass decoding during training. At the first decoding pass, the translation is generated by the decoder. Then, we compare the predicted translation with the reference and sample target words from the reference according to how well this translation is predicted. Next, we replace the copied embeddings with the embeddings of sampled target words at corresponding positions for the second pass. And then, we predict the remaining words using maximum likelihood estimation. Thus, by glancing training, we have the right signals to guide the learning of the discriminator. Formally, given a sentence X and its corresponding target sentence Y , the training objective of our decoder is:

¹ The discriminator is made up of a single Transformer decoder block. And the hidden size of the discriminator is consistent with NAT decoder.

$$\mathcal{L}_{decoder} = - \sum_{y_t \in \{Y \setminus \text{GS}(Y, \hat{Y})\}} \log p(y_t | \text{GS}(Y, \hat{Y}), X; \theta). \quad (5)$$

And \hat{Y} is the translation predicted by the first decoding pass. And $\text{GS}(Y, \hat{Y})$ is a set of sampled words. In this work, the sampling strategy is borrowed from [14]. For discriminator, we set the λ of sampled target words as 1, and others as 0. With these signals, the discriminator can learn which embeddings are identical with its target embeddings correctly. So the training objective of our discriminator is:

$$\mathcal{L}_{discriminator} = - \sum_{t=1}^M \log P_d \quad (6)$$

Overall, the whole model is trained by minimizing the total loss:

$$\mathcal{L} = \mathcal{L}_{decoder} + \mathcal{L}_{discriminator} \quad (7)$$

During inference stage, we first form the decoder inputs by copying the source embeddings, and then we add embedding of “[MASK]” on decoder inputs according to the similarities predicted by discriminator. Then, we feed the new decoder inputs to the decoder and the decoder performs vanilla decoding pass.

4 Experiments

4.1 Experiment Settings

Dataset. To validate the effectiveness of our model, we conduct experiments on three translation datasets: WMT14 EN → DE, WMT16 EN → RO, and IWSLT14 DE → EN. These three datasets consist of 4.5M, 610k and 160k bilingual sentence pairs. And these three datasets are tokenized using the script provided by Moses.² And then we segment each word in these datasets into sub-word units with Byte-Pair Encoding [15]. For WMT14 EN → DE task, we use newstest-2013 and newstest-2014 as the validation and test sets respectively. For WMT16 EN → RO task, we use newsdev-2016 and newstest-2016 as the validation and test sets, respectively. And for IWSLT14 DE → EN, dev2010, dev2012, tst2010, tst2011, tst2012 are concatenated as the test set.

Distillation. As described in [4], distillation is crucial for NAT models. In this work, we follow the previous works [4, 8] and use sequence-level distillation for all datasets. For WMT14 EN → DE, WMT16 EN → RO, and IWSLT14 DE → EN, we use Transformer-based [18] the teacher model to distill all datasets. And then we train our models on the distilled corpus.

Model Configuration. We implement our proposed model based on open-source toolkit Fairseq [12]. And for a fair comparison, we follow most of hyperparameters provided in [4, 8, 14]. For IWSLT14 DE → EN task, we utilize a small

² <https://github.com/moses-smt/mosesdecoder>.

configuration: $N_{layer} = 5$, $d_{model} = 256$, and $n_{head} = 4$. And for WMT tasks, we follow the configurations provided in Transformer [18]. We use Adam optimizer [7] with $\beta = (0.9, 0.98)$ to optimize our model.

Training and Inference. We train our models with 8/1 Nvidia Tesla V100 GPUs on WMT datasets and IWSLT dataset respectively. And we train the model with batches of 64k/8k tokens for WMT and IWSLT datasets respectively. Meanwhile, given the effectiveness of noise parallel decoding (NPD) [4], we also utilize NPD to select the best translation. As for evaluation, we adapt widely-used BLEU [13] as the evaluation metric.

Baselines. In the experiments, we compare our model with several baselines: **NAT** is a vanilla non-autoregressive machine translation model proposed by [4]. **Hint-NAT** [9] utilizes the alignment information extracted from an autoregressive machine translation model. **TCL-NAT** [10] transfers the knowledge from AT model to NAT model with curriculum learning. **Flowseq** [11] models the generated flow as latent variables. **DCRF-NAT** [17] utilizes an approximation of CRF for non-autoregressive machine translation. **GLAT** [14] adapts GLM to improve the performance of non-autoregressive machine translation. **I-NAT** [8] improves vanilla non-autoregressive machine translation by introducing iterative refinement. **Mask-Predict** [3] incorporates a masked language model into non-autoregressive machine translation.

4.2 Main Results

The main results of our models are listed in Table 1. Clearly, compared with the baselines, our model significantly improves the performance and outperforms the single-pass NAT baselines by a large margin. Furthermore, the results show that it is effective to introduce discriminator to detect how different copied embeddings are from the embeddings of the target words.

Compared with vanilla NAT model, our model achieves significant improvements on WMT14 EN → DE and WMT16 EN → RO. Even compared with strong baseline, GLAT, our model also achieves the better results, which denotes the introduction of soft-masking can help the decoder learn the distribution of target sentence well. What is noteworthy is that iteration-based models achieve better performances compared with fully NAT models. However, these improvements are based on the sacrifice of speed advantage. It was noted by [6] that with careful allocation of layers in AT models, AT models can also have the same-level speedup. In contrast, our model still maintains the decoding advantage of fully NAT models.

From the last group of Table 1, we also notice that noise parallel decoding is crucial to improve the performance of fully NAT models. Especially on IWSLT14 DE → EN, reranking 4 candidates obtains a gain of 1 BELU score, and reranking 7 candidates gains 1.5 BLEU improvement. Although reranking with an autoregressive model increases the decoding latency, it still gets a faster decoding speed than Transformer.

Table 1. The performance of BLEU scores on WMT14 EN → DE, WMT16 EN → RO and IWSLT14 DE → EN. “/” denotes that the results are not reported. “K” denotes the number of decoding iterations.

Model	WMT14 EN-DE	WMT16 EN-RO	IWSLT14 DE-EN	Speedup
Transformer	27.30	34.16	32.99	1×
Single-pass NAT models				
NAT	17.69	26.22	/	15.6×
NAT (NPD100)	19.17	29.79	24.21	2.4×
Hint-NAT	21.11	/	/	25.55×
TCL-NAT	21.94	/	28.16	27.6×
Flowseq	21.45	29.34	27.55	/
DCRF-NAT (NPD9)	26.07	/	29.99	9.63×
GLAT	25.21	31.19	/	15.3×
GLAT (NPD7)	26.55	32.87	/	7.9×
Iterative-based NAT models				
I-NAT (K = 1)	13.91	24.45	/	/
I-NAT (K = 10)	21.61	29.32	23.94	1.5×
Mask-Predict (K = 4)	25.94	32.53	30.42	9.79×
Mask-Predict (K = 10)	27.03	33.08	31.71	3.77×
Ours	25.73	31.83	29.67	15.03×
+NPD4	26.32	32.52	30.65	8.14×
+NPD7	26.83	33.03	31.15	7.8×

4.3 Decoding Speed

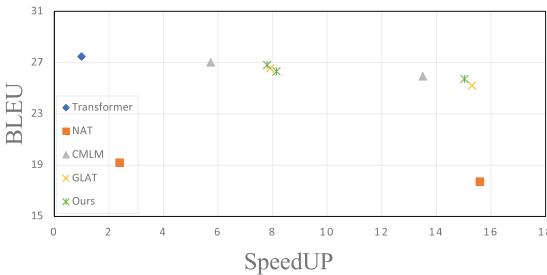


Fig. 2. The BLEU scores and Speedup on WMT14 EN → DE translation dataset.

To evaluate the decoding speedup of our model, we follow the previous works [4, 8], and calculate the average per sentence decoding time on WMT14 EN → DE test set with batch size as 1. And we report the absolute speedup in Table 1. Meantime, for a more intuitive comparison, we present the scatter plot in Fig. 2, which displays the trend of speedup and BLEU scores.

Obviously, we can observe that the trade-off between BLEU score and speedup of our model outperforms the competitors, which location is on the top-right position of Fig. 2. And we notice that the iterative decoding model, CMLM, achieves better accuracy. However, the speedup of CMLM is not obvious compared to AT model Transformer. And from Table 1, we can see that our model obtains comparable performance with Transformer, while achieving a $7.8\times$ speedup. Although compared with other fully NAT models, the speedup of our model is not the best. With controlled speedup, our model achieves better accuracy.

5 More Analysis

Effect of Sentence Length. To evaluate the effect of our model on different sentence lengths, we conduct experiments on the IWSLT14 DE \rightarrow EN test set and divide the sentences into different buckets according to the lengths of references. And we show the results in Fig. 3.

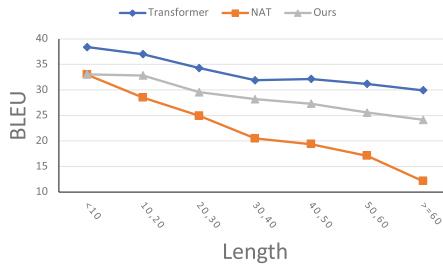


Fig. 3. The performance with respect to different sentence lengths.

From Fig. 3, we can observe that our model can improve the performances on different lengths of sentences. Especially, when the length of sentence is greater than 50, the BLEU scores of vanilla NAT model drop quickly, while our model and AT model have relatively stable performance. In addition, when the length of sentence is less than 10, the performance of our model and vanilla NAT model is at the same level. The conclusion can be drawn that our introduced soft-masking can effectively improve the performance of NAT model on long sentences.

Effectiveness of the Adaptive Masking. To validate the effectiveness of the adaptive soft-masking according to the discriminator, we also introduced a fixed soft-masking strategy for comparison. We manually set the λ in Eq. 4 as a fixed value, and conduct experiments on IWSLT14 DE \rightarrow EN test set. We report the results in Table 2. From Table 2, we can find that if we add mask on the embeddings according to a fixed value, the performance of NAT

Table 2. The performance of soft-masking strategies on IWSLT14 DE → EN test set.

Soft-Masking	λ	BLEU
Fixed	1.0	25.88
	0.9	25.84
	0.7	25.72
	0.5	25.85
Adaptive	/	29.67

model drops. This is because due to the fixed value, the embeddings of every word are added by the same ratio of mask. And this same ratio of mask is the noise for the model, which decreases the model’s performance. Correspondingly, the soft-masking introduced by us is dynamically regulated according to the input embeddings, which potentially denote the distance between the input embedding and the target embedding. In addition, when the λ is set as 0.5, the model achieves same performance as the model with $\lambda = 0.9$. This is because the noise introduced by mask is so much that the decoder sees the copied embeddings as placeholders, and the generation of the target words mainly lies on the inter-attention module. The opposite is that the mask introduced by our model denotes which embeddings are correct, and these correct embeddings provide the decoder part of target information.

The Effect of Repetitive Words. As described in [4, 5, 19], repetitive words are the main cause of performance degradation in NAT models. In this work, we suppose the autoregressive discriminator introduced by us can assist the model to learn sequential information, which can effectively reduce the number of repetitive words [20]. So, we conduct experiments to validate the effect of our model on reducing the number of repetitive words. And we show the results in Table 3. From the results shown in Table 3, the number of repetitive words generated by our model is significantly less than vanilla NAT model. And compared with NAT-Reg [19], our model can also alleviate this issue without explicit regularization. Even compared with iteration-based model CMLM, our model can achieve a comparable result, which shows the effect of our model. And these results prove that the discriminator does help the decoder learn target dependencies.

Table 3. The comparison on the number of per-sentence repetitive tokens on the validation set of the IWSLT14 De → En task.

NAT	NAT-Reg	CMLM	Ours
2.30	0.90	0.48	0.68

6 Related Works

Since [4] introduced non-autoregressive Transformer to speed up the inference stage of neural machine translation, a series of works have been proposed to improve the performance of NAT models [3,4,8]. Especially, [14] introduced a glancing-based model GLAT, which significantly improves the performance of NAT models and achieves comparable results with AT models. And in this work, we borrow the principle of GLAT, glancing, to train our model.

In addition, our model is related to [20], which introduced a discriminator to infuse the sequential information. The critical difference is that they introduced discriminator to determine which words generated by the decoder are correct, and we introduce discriminator to determine the similarity between the target embedding and copied embedding. While the method proposed by [20] can only be applied on iteration-based NAT models, our method is designed for fully NAT models.

7 Conclusion

In this work, we introduce a soft masking model to improve the performance of NAT model. In our model, we utilize an autoregressive discriminator to determine the difference between the target embedding and copied embedding, and add mask on the embedding according to the output of the discriminator. Experiments show that our proposed model can significantly improve the performance of non-autoregressive machine translation.

Acknowledgements. We would like to thank the anonymous reviewers for their insightful comments. And our work is supported by the National Science Foundation of China (61732005, 61671064).

References

- Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint [arXiv:1409.0473](https://arxiv.org/abs/1409.0473) (2014)
- Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. In: ICML (2017)
- Ghazvininejad, M., Levy, O., Liu, Y., Zettlemoyer, L.: Mask-predict: parallel decoding of conditional masked language models. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 6114–6123 (2019)
- Gu, J., Bradbury, J., Xiong, C., Li, V.O., Socher, R.: Non-autoregressive neural machine translation. arXiv preprint [arXiv:1711.02281](https://arxiv.org/abs/1711.02281) (2017)
- Guo, J., Xu, L., Chen, E.: Jointly masked sequence-to-sequence model for non-autoregressive neural machine translation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 376–385 (2020)

6. Kasai, J., Pappas, N., Peng, H., Cross, J., Smith, N.A.: Deep encoder, shallow decoder: reevaluating the speed-quality tradeoff in machine translation. arXiv preprint [arXiv:2006.10369](https://arxiv.org/abs/2006.10369) (2020)
7. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
8. Lee, J., Mansimov, E., Cho, K.: Deterministic non-autoregressive neural sequence modeling by iterative refinement. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 1173–1182 (2018)
9. Li, Z., He, D., Tian, F., Qin, T., Wang, L., Liu, T.Y.: Hint-based training for non-autoregressive translation (2018)
10. Liu, J., et al.: Task-level curriculum learning for non-autoregressive neural machine translation. arXiv preprint [arXiv:2007.08772](https://arxiv.org/abs/2007.08772) (2020)
11. Ma, X., Zhou, C., Li, X., Neubig, G., Hovy, E.: FlowSeq: non-autoregressive conditional sequence generation with generative flow. arXiv preprint [arXiv:1909.02480](https://arxiv.org/abs/1909.02480) (2019)
12. Ott, M., et al.: fairseq: a fast, extensible toolkit for sequence modeling. In: Proceedings of NAACL-HLT 2019: Demonstrations (2019)
13. 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 of the Association for Computational Linguistics, pp. 311–318 (2002)
14. Qian, L., et al.: Glancing transformer for non-autoregressive neural machine translation. arXiv preprint [arXiv:2008.07905](https://arxiv.org/abs/2008.07905) (2020)
15. 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 (Volume 1: Long Papers), pp. 1715–1725 (2016)
16. Shao, C., Zhang, J., Feng, Y., Meng, F., Zhou, J.: Minimizing the bag-of-n-grams difference for non-autoregressive neural machine translation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 198–205 (2020)
17. Sun, Z., Li, Z., Wang, H., He, D., Lin, Z., Deng, Z.: Fast structured decoding for sequence models. In: Advances in Neural Information Processing Systems, vol. 32, pp. 3016–3026 (2019)
18. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
19. Wang, Y., Tian, F., He, D., Qin, T., Zhai, C., Liu, T.Y.: Non-autoregressive machine translation with auxiliary regularization. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 5377–5384 (2019)
20. Xie, P., Cui, Z., Chen, X., Hu, X., Cui, J., Wang, B.: Infusing sequential information into conditional masked translation model with self-review mechanism. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 15–25 (2020)

Machine Learning for NLP



AutoNLU: Architecture Search for Sentence and Cross-sentence Attention Modeling with Re-designed Search Space

Wei Zhu^(✉)

East China Normal University, Shanghai, China
52205901018@stu.ecu.edu.cn

Abstract. The rise of BERT style pre-trained models has significantly improved natural language understanding (NLU) tasks. However, for industrial usage, we still have to rely on more traditional models for efficiency. Thus, in this paper, we present AutoNLU, which is designed for modeling sentence representation and cross-sentence attention in an automatic network architecture search (NAS) manner. We have two main contributions. First, we design a novel and comprehensive search space that consists of encoder operations and aggregator operations, and important design choices. Second, aiming for sentence-pair tasks, we use NAS to automatically model how the representations of two sentences interact with and attend to each other. A reinforcement learning (RL) based search algorithm is enhanced by cross operation and cross layer parameter sharing for efficient and reliable search. Model training is done by distilling knowledge from BERT models. By experimenting on SST-2, RTE, Sci-Tail and CoNLL 2003, we verify that our learned models are better at learning from BERT teachers than other baseline models. Ablation studies on Sci-Tail show that our search space design is valid, and our proposed strategies are helpful for improving the search results (The source code will be made public available.).

Keywords: Natural language understanding · Neural architecture search · Reinforcement learning

1 Introduction

Neural architecture search (NAS) has recently attracted intensive attention. On one hand, promising methodological innovation for NAS have been developed, e.g. the seminal gradient-based NAS approach DARTS [13], etc. On the other hand, NAS has helped to discover better network architectures for a variety of vision tasks, ranging from image classification [3], semantic segmentation [12], object detection [6], to super-resolution [1], etc. For natural language understanding (NLU) tasks, NAS is relatively less studied. Except for the general methodology-wise innovations NASNet [24], ENAS [15] and DARTS [13] which address searching for new RNN cells on language modeling (LM) tasks, there

are few studies tailored to the NLU task. One such an example is the evolved transformer [17], which uses the evolution-based NAS algorithm to search for better transformer architecture for machine translation. However, the methodologies or search spaces in the above literature can not be directly applied to NLU tasks like text classification (CLS) and natural language inference (NLI), due to the following reasons.

- A typical neural network architecture for NLU includes an encoder which encodes the embedded text inputs, and an aggregator that aggregates the encoded inputs to a fix-length vector to make a prediction [7]. ENAS [24] and DARTS [13] restricts themselves with RNN encoders, and [17] does not consider searching for aggregators.
- For NLU tasks with multiple inputs, cross sentence attention (cross-attn) is proven beneficial [4, 10, 19]. However, no previous work has shown how to apply NAS into cross-attention modeling.
- For NLU tasks there are many design choices like whether to freeze embedding, number of layers, etc., which is not addressed in the above literature.

This article attempts to address the above issues. We first define a comprehensive search space designed to better constitute the backbone of a neural network model for NLU tasks, i.e., the encoder search space and aggregator search space. Second, other import design choices, like whether to freeze word embedding, whether to use positional embedding, how many layers the network needs, and how the intermediate hidden states contributes to the later layer, are import but time-consuming to tune. We also include these design choices into the search space. Third, for tasks with multiple sentence inputs, like natural language inference (NLI), how the sentences attend to each other, and how to combine the cross-attention modules together with self-encoding operations are important, which to our knowledge, have not been studied in literature. To automatically model the cross-attention between sentences, we decompose this issue into steps: i) which attention function to use [2, 19], other than the dot product attention [21]; ii) the position setting choice, i.e., before the first self-encoding sub-layers, or in between, or after. iii) whether to accompany the cross-attention operation with a self-encoding operation, and if so, which one to use. The above three aspects will be decided automatically in our pipeline.

To improve the search stability and performance, we propose a series of strategies on top of RL based search algorithm. First, parameters are shared across different operation. We design specific parameter sharing strategies to accommodate the usage of depth-wise separable convolutions and multi-head attentions. Second, parameters are shared across different layers, which are especially import for block-wise search space like ours. Third, to obtain reliable reward signal, the shared parameters are warmed up in the beginning epochs of the training. To improve performances, we also use knowledge distillation (KD), and consider the BERT Large model as the teacher model.

Our work contributes the field by the following aspects:

- We re-define the search space for neural architecture search in NLU tasks, by not only including encoder and aggregator operations, but also including many import design choices into the search space.
- We propose to model cross-sentence attention automatically via network architecture search, which has shown its usefulness in our experiments, and to our knowledge has not been done in the literature.
- Our approach learns novel models whose performances are much closer to the teacher model BERT Large than the baseline models.

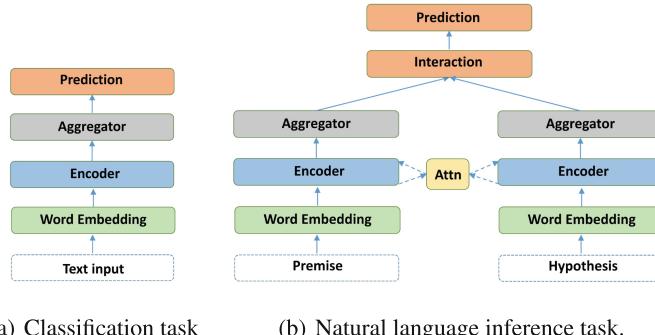


Fig. 1. Meta architectures for CLS and NLI tasks.

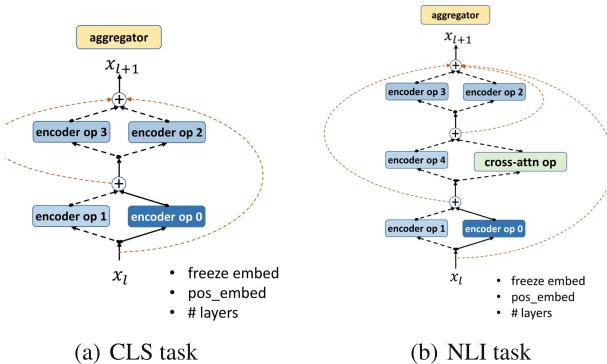


Fig. 2. Sub-layer architectures of encoders for different tasks.

2 Search Space Design

In this section, we elaborate on the search space design and formally introduce our search space.

2.1 Meta-architectures

As pointed out by [7] and depicted in Fig. 1(a), a neural network for sentence representation in a text classification task usually consists of the following components: i) an embedding layer, which maps words into low dimensional vectors; ii) encoding layer(s), which integrates and extracts higher-level features; iii) an aggregator layer, which aggregates the information to a single fix-length vector. Thus, our search space needs to define which encoder and aggregator operations are considered. The meta architecture (see Fig. 1(b)) for a NLI task is similar. However, as is shown in ESIM [4], MwAN [19], doing cross-attn can significant improve the performance of NLI tasks. Previous literature manually designs how the cross-sent attention module is combined with other encoder operations. In this work, we try to make the design of cross-attn module automatic.

We now define the sub-layer level structure inside an encoder for CLS and NLI tasks, as is depicted in Fig. 2(a) and Fig. 2(b). A sub-layer of an encoder consists of at most two encoder operations, which encode the same input and then the results are summed and fed into the next module. For a sub-layer that attends to the other sent, it can consists at most one self-encoding operation. We call the this type of sub-layer as cross-attn sub-layer, and that only encodes itself as a self-encoding sub-layer. An entire encoder layer consists at most two self-encoding sub-layers and at most one cross-attn sub-layer. In this work, we will automatically determine the following: i) how many encoder operations to use, i.e., how many self-encoding sub-layers are needed, and which encoder operations to use; ii) where to put the cross-attn sub-layer, it can be before the self-encoding sub-layers, or in the middle, or after; iii) which cross-attn operation to use. iv) is there a self-encoding operation in the cross-attn sub-layer.

2.2 Encoder Operations

In [13, 15], the objective is to discover new variants of RNNs, so their search space is a collection of linear or non-linear maps, such as, *tanh* and *sigmoid*. In this work, we will define the encoder space at a higher granularity, allowing to build a richer encoder search space. Recent years have witnessed the architecture of transformers [21] becoming ubiquitous. In this work, similar to [17], we include the multi-head attention layer, into the search space. The point-wise feed-forward layer contains conv1d and residual connection, so we will not include it as a basic operation.

Now we formally give out the encoder operation search space (Henceforth, ENCODER_SPACE), which consists of the following operations:

- Special zero operation, denoted as **null**;
- Skip connection, denoted as **identity**;
- 1-d depth-wise separable convolutions, with kernel size k , where $k = 1, 3, 5$, denoted as **sep.conv.k**;
- Two RNNs, which are denoted as **lstm** and **gru**;
- Multi-head self-attention, with number of heads $k = 1, 2, 4, 8$, denoted as **mha.k**.

2.3 Aggregator Search Space

There are several different aggregation operations. The most common two are the max pooling and the average pooling. Self-attention technique is also used for aggregation. We also include dynamic routing [7] into our aggregator operation space. It involves two hyper-parameters that affect the final performance: i) the number of capsules; and ii) the number of iterations. Therefore, we design the aggregator search space (henceforth, AGGREGATOR-SPACE) as:

- Max pooling, denoted as **max-pool**;
- Average pooling, denoted as **avg-pool**;
- Self-attention pooling, denoted as **self-attn-pool**;
- Dynamic routing, where the number of capsules is $nc = 2, 4$ and the number of iterations is $ni = 2, 3$, denoted as **dr_cap_nc_iter_ni**.

2.4 Design Choices

There are many design choices for building a neural network architecture. For the embedding layer, whether to fine-tune the word embedding is task specific [19]. In addition, whether to use positional embedding, and if so, which positional embedding to use needs to be determined.

For the encoder layers, first we need to decide how many self-encoding sub-layers are required, where the search space is $num_ses = 0.5, 1, 1.5, 2$. Here, 0.5 means that a sub-layer only requires one self-encoding operation. Correspondingly, one need to select a combination of encoder operations to fill in the sub-layers.

For the cross-attn sub-layer, firstly one has to decide whether cross-attn is needed at all. Second, which attention mechanism to select. In addition to the dot product attention used in MHA (denote as **dot**), we incorporate the four attention functions in [19], which are referred to as **p_dot**,¹ **concat**, **add**, **minus**.

Third, where to put the cross-attn sub-layer needs to be determined. We denote its position as $ca_pos = 0, 1, 2$, which corresponds to being before the self-encoding sub-layers, or in between, or after, respectively. In addition, whether a self-encoding operation will be needed in the cross-attn sub-layer (denoted as $num_ses_ca = 0, 1$) and if true, which operation to use also need to be determined.

Skip connection is of central importance to the optimization of a neural network [8]. Thus, we add another hyper-parameter in, i.e., How many intermediate outputs (counting backward from the last output inside an encoder layer) of sub-layers will be used to be the input of the next encoder layer. Here the choices are $num_outputs = 1, 2, 3, 4$. Note that transformer-style residual connection

¹ Note that the dot attention in [19] (denoted as **p_dot** by us) is not the same with the dot product attention in MHA, where the former is point-wise product ('A * B' in PyTorch), and the latter is (batch-wise) matrix multiplication ('torch.bmm(A, B)').

is included in the solution set of our search space, since we include **identity** operation in the ENCODER_SPACE.

Now, we are ready to define the whole search space:

- Number of self-encoding sub-layers = 0.5, 1, 1.5, 2;
- Operations used in self-encoding sub-layers = SELF_ENCODER_OP_SPACE;
- Number of self-encoding operation in the cross-attn sub-layer = 0, 1;
- Whether to do cross-attn = True, False;
- Attention function in the cross-attn operation = **dot**, **p-dot**, **add**, **concat**, **minus**;
- Position of cross-attn operation = 0, 1, 2;
- The number of intermediate outputs summed as the input of the next layer = 1, 2, 3, 4;
- Which positional embedding to use = learned_pos, sinusoid_pos, null, where null means not to use positional embedding;
- Whether to add positional embedding at the intermediate encoder layers = True, False;
- Whether to freeze word embedding layer = True, False;
- Number of encoder layers = 1, 2, 3, 4, 6;
- Aggregator operation type = AGGREGATOR_SPACE.

3 Architecture Search

AutoNLU uses a reinforcement learning algorithm with weight sharing to perform architecture searches. Our algorithm is similar to ENAS [16], but contains changes to improve robustness and scalability of the search process.

3.1 Search Algorithm

A controller, which is a LSTM network, is used to generate new architectures. At each step of the controller, a categorical decision is made which controls a certain aspects of the network architecture. For example, a single categorical decision might control whether we use a convolution (of certain kernel size) or a LSTM operation at a particular position in the encoder layer. An architecture is an assignment of values to these categorical decisions.

During a search, we learn a policy π , a probability distribution from which we can sample high quality architectures from the controller. Formally, π is defined as a collection of independent multi-nominal variables, one for each of the decisions in our search space. We also learn a set of shared weights W , which are used to efficiently estimate the quality of candidate architectures without training each of them till convergence.

Learning for the policy and the shared parameters is conducted in an interleaving fashion. At each step, we first sample a network architecture $\alpha \sim \pi$. Next, we use the shared weights to estimate the performance $r(\alpha)$ of the sampled architecture using a single or a few batches of examples from the validation set. $r(\alpha)$

is regarded as the reward signal to update the policy π using REINFORCE [23]. Finally, we update the shared model weights W by computing gradient updated w.r.t. the architecture α on a single or multiple batches of examples from the training set. The above process is repeated over and over until the search completes. At the end of the search, one will re-rank the architectures encountered during search by their rewards using the current shared weights, and select the top- k models.

3.2 Child Model Training

We adopt the BERT Large model as the teacher model, and our child model is trained by approximating the teacher model’s behaviours. The workflow of knowledge distillation is as follows. Firstly, we train the teacher model till convergence. Denote X as a input sequence, c as one of the classes (i.e., the sentiment is negative or positive), and we denote the teacher model’s predicted probability that X is labeled as c as $\mathcal{P}_T(c|X)$. The child model’s prediction is denoted as $\mathcal{P}_C(c|X)$. Without distillation, the training objective of the child model is the cross entropy loss, i.e.,

$$-\sum_c \mathcal{I}(X, c) \log(\mathcal{P}_C(c|X)), \quad (1)$$

where $\mathcal{I}(X, c)$ is the binary indicator (0 or 1) indicating whether class label c is the correct classification for X . That is, we want our child model to fit on the distribution of the training dataset. In comparison, with a teacher model, our objective becomes

$$-\sum_c \mathcal{P}_T(c|X) \log(\mathcal{P}_C(c|X)). \quad (2)$$

Table 1. Overview of used datasets in experiments.

Dataset	Task	Train #	Dev #	Test #	Label #	Metrics
SST-2	CLS	77k	872	1.8k	2	Acc
RTE	NLI	2.5k	277	3.0k	2	Acc
SciTail	NLI	23.5k	1.3K	2.1k	2	Acc
CoNLL2003	NER	14k	3.2K	3.4k	4	F1

Note that Eq. 2 differs from the cross entropy loss in Eq. 1 in that the former uses the soft targets $\mathcal{P}_T(c|X)$ provided by powerful teacher model while the latter uses the hard correct target $\mathcal{I}(X, c)$.

3.3 Improving Weight Sharing

To reduce the cost of an architecture search, NAS algorithms that are based on weight sharing [3, 13, 16] always train a large network – a super-net – with many redundant operations, most of which will be removed at the end of the search. There are two issues not fully investigated in the previous literature: (1) Previous literature treat each operation as completely separate, ignoring the weights that can be shared; (2) operations in different blocks are separate, thus linearly increasing the number of shared parameters, which may result in unreliable reward signals. Now we introduce a series of more aggressive weight sharing strategies to resolve the above problems.

Cross-Operation Parameter Sharing. (COPS) Instead of using totally separate sets of weights for each choice of operation in the encoder search space at each sub-layer, we aggressively re-use the weights that can be shared: (1) for multi-head attention modules, including that in the self-encoding operation and in the cross attention operation, we set the head size by dividing the hidden size with #heads, so that the query, key, value matrix can be shared across multi-head attention operations with different #heads; (2) each 1-d depth-wise separable conv can share the same point-wise conv (i.e., the conv with kernel size 1). (3) for aggregation layer, we share the fully-connected layer in the dynamic routing aggregators across different numbers of iterations and numbers of capsules.

Cross-Layer Parameter Sharing. In this paper, we employ cross-layer parameter sharing (CLPS). During search, we let all the encoder layers share the same shared parameters, thus significantly reducing the quantities of shared parameters during search, and as a result, helping to accelerate convergence and yielding better architectures. During the architecture evaluation stage, we train the yielded models both with CLPS and without, and pick the one with better dev performance.

3.4 Search Warm-Up

When the search begins, the shared weights are all randomly initialized, thus rewards obtained using them is unreliable and may mislead the controller. Thus for the beginning epochs of the search, we do not update the parameters of the controller. In addition, considering the search space involves multi-head attention which requires warm up, thus, we also warm up the learning rate for the shared weights.

Table 2. performances on the benchmark datasets. ‘–’ denotes the numbers are not reported in the original paper otherwise from the reference.

Model	SST-2	RTE	SciTail	CoNLL2003
BiLSTM + Attn [22]	85.9	51.9	–	–
BiLSTM + Attn + Elmo [22]	90.2	50.4	–	–
DGEM [9]	–	–	70.8	–
BiLSTM	85.9	52.8	71.1 (72.6)	80.2 (82.8)
Transformer [21]	86.5 (87.1)	49.3 (49.9)	73.2 (73.6)	67.1 (68.5)
BiLSTM (distilling BERT-Large)	89.3 (99.9)	64.2 (65.1)	82.2 (83.5)	86.4 (87.5)
Transformer (distilling BERT-Large)	89.7 (90.6)	64.8 (65.4)	83.1 (83.9)	86.2 (87.4)
ESIM (distilling BERT-Large)	–	65.2 (65.9)	83.5 (84.7)	–
BERT-Large [5]	94.9 (95.8)	70.1 (70.6)	88.3 (89.2)	91.4 (92.8)
Random search	88.3 (89.3)	57.2 (59.8)	77.8 (80.3)	84.8 (86.9)
Search w/o distillation	88.6 (89.1)	53.5 (53.7)	75.6 (75.9)	82.4 (83.5)
$\text{AN}_{sst,0,t}$	90.8 (92.7)	–	–	–
$\text{AN}_{sst,2,u}$	91.5 (92.9)	–	–	–
$\text{AN}_{rte,0,t}$	–	66.1 (67.9)	–	–
$\text{AN}_{rte,4,t}$	–	67.5 (68.2)	–	–
$\text{AN}_{scitail,0,u}$	–	–	84.1 (84.8)	–
$\text{AN}_{scitail,4,t}$	–	–	85.2 (85.9)	–
$\text{AN}_{conll03,0,u}$	–	–	–	87.2 (87.9)
$\text{AN}_{conll03,3,u}$	–	–	–	87.7 (88.5)

4 Experiments and Discussion

In our experiments, for each search or evaluation, we assign 2 CPU cores, 5G memory and 1 T V100 GPU card.

4.1 Datasets

We conduct experiments for CLS, NLI and NER tasks, with 4 benchmark datasets whose statistics are shown in Table 1.

SST-2. Stanford Sentiment Treebank (SST) is a movie review dataset which has been parsed and further splitted to train/dev/test set by [18].

RTE. Recognizing Textual Entailment (RTE) comes from a series of annual textual entailment challenges and is included as part of the GLUE benchmark [22].

SciTail. This is a textual entailment dataset derived from a science question answering (SciQ) dataset [9].

CoNLL2003. This dataset consists of 200k training words which have been annotated as Person, Organization, Location, Miscellaneous, or Other (non-named entity).

4.2 Architecture Search Protocols

Experiments on each task consist of two stages, architecture search and architecture evaluation. In the search stage, we train the policy function with the shared parameters and select 5 candidate architectures. In the second stage, the architectures are trained from scratch for multiple times using the complete training set, with or without cross layer parameter sharing. After obtaining the model checkpoints with the best dev performances, the prediction on test are made.

The training of BERT-Large teacher model closely follow [5]. For the child model, the pre-trained embedding is from [14] (840B, 300d). During search phase, the interleaving optimization process runs for 100 epochs on train set. On each interleaving step, 5 batches of train samples is used to update the shared weights, and 2 batch of dev samples is used to obtain rewards. Search warm-up and learning rate warm-up are set to be 10% and 3% of the whole search phase. The training batch size is 32 and eval batch size is 128. Optimizers for shared parameters and controller are both Adam [11] with lr $1e-4$. Weight decay is set as $1e-4$ and gradient clipping is set to be 1.

To compare our methods with random search, for each task we randomly sample 10 different models and train them from scratch, and report the performance of the best model as the performance of random search.

Table 3. Comparison of GPU memory consumption and inference speed between the teacher model BERT Large and $\text{AN}_{scitail,4,t}$, when doing inference on Sci-Tail test dataset.

Model	GPU memory	Inference speed
BERT Large	7.8 GB	1.1 it/s
$\text{AN}_{scitail,4,t}$	0.84 GB	16.6 it/s

4.3 Results

In the tables referred in this section, we report the average and best (in the bracket) performances of multiple runs. For notation convenience, we denote the architecture obtained by searching on task x , ranked i -th in the search phase and training from scratch with or without CLPS b ('t' for true and 'u' for false) as $\text{AN}_{x,i,b}$.

Results on the 4 datasets are presented in Table 2, and best learned architectures are depicted in Fig. 3(a), 3(b), 3(c) and 3(d). First, we can see that performances improve significantly with the help of KD, both for the baseline models and architecture search. Second, we can see that the learned models

outperform the baselines significantly, both with KD and without. The results demonstrate NAS's ability to better fit the learning signals in a task specific level. Third, Table 3 shows that during inference with the same batch size 128, our learned model on Sci-Tail are 16 times faster and consume 8 times less GPU memories than BERT Large.

SST-2 selects a simple encoder layer with a single sub-layer and stack the encoder layer for 4 times. On the RTE task, the best model uses a complex single encoder layer which includes RNN, convolution and self-attention operations. However, as a NLI task, RTE chooses not to include cross attention. Cross attention is applied by AN_{scitail,4,t}, and the attention function used is **add**. The encoder for CoNLL03 is a combination of RNN and conv-1d with kernel size 1. This pattern of sub-layer also appears on the RTE task. For aggregators, 2 out of 3 tasks select dynamic routing, but with different hyper-params, and RTE select **avg_pool**, which validate the necessity of the aggregator search space.

4.4 Ablation on Our Strategies

In the previous section, we propose three strategies, cross operation parameter sharing (COPS), cross layer parameter sharing (CLPS) and search warm-up.

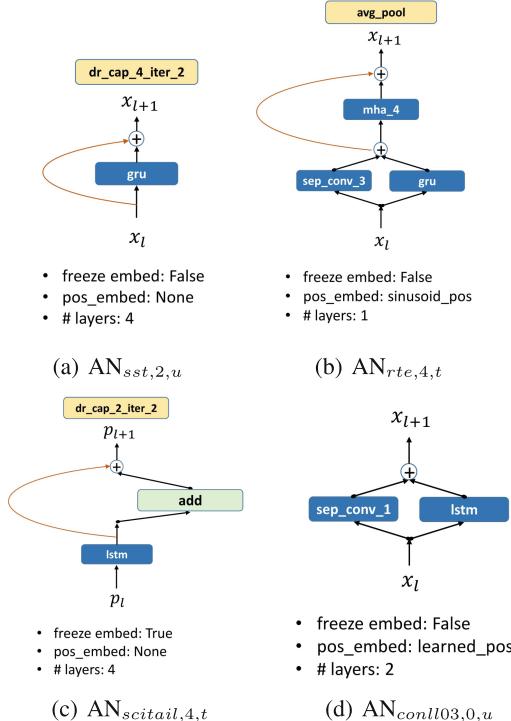


Fig. 3. Learned architectures on the four benchmark datasets.

We now study on how searching with or without our proposed strategies affects the final search performance. We conduct separate search runs, with different sets of strategies, under the same search settings, and use the performance of the best model obtained as the metric for a search run. We consider four scenarios: (1) search with our proposed strategies, which is denoted as **search**; (2) dropping the CLPS strategies, denoted as **search w/o CLPS**; (3) further dropping the COPS strategies, denoted as **search w/o CLPS or COPS**; (4) further search without search warm-up (**search w/o CLPS, COPS or warm-up**).

The ablation is conducted on Sci-Tail, and the results are reported on Table 4. As we can see, conducting search with CLSP will significantly stabilize the search process and provide more reliable search results. The intuition is quite straightforward. Doing CLSP significantly reduce the number of shared parameters during the search phase, thus making the reward signal during search more reliable. The results also shows COPS and search warm-up are beneficial for search.

Table 4. Results for ablations studies on the strategies we propose for search.

Strategies	Test ACC
Search	85.2 (85.9)
Search w/o CLPS	84.3 (84.8)
Search w/o CLPS or COPS	83.5 (84.2)
Search w/o CLPS, COPS or warm-up	83.0 (83.8)

Table 5. Results for ablations studies on the search space we construct for architecture search for NLU tasks.

Search space	Test ACC
baseline: BiLSTM + max_pool	82.2 (83.5)
SP_0	82.7 (83.3)
SP_1	83.4 (84.6)
SP_2	84.3 (85.0)
SP_{full}	85.2 (85.9)

4.5 Ablation on Our Search Space

We now conduct ablation study on our entire search space. We start with a simple baseline, BiLSTM with max pooling (single layer, no cross attention, no positional embedding, not to freeze word embedding). We gradually add parts of the search space: (1) add aggregator search space (SP_0); (2) add search options related to self-encoding operations (SP_1); (3) add search space for cross attention (SP_2); (4) further adding the rest of the design choices (SP_{full}). This ablation study is done on Sci-Tail, and results are shown in Table 5. As we gradually expand the search space, the test performances improves significantly, demonstrating the necessity of each components of our search space.

5 Conclusion and Future Work

In this work we experiment on modeling sentence representation and cross-sentence attention via NAS. First, we include encoder and aggregator operations in the search space, to accommodate the meta-architectures of NLU models, which is novel in the literature. Second, we are the first to experiment on modeling cross-sentence attention via NAS, by letting the controller decide which attention function to use, where to put the cross attention, and how it is combined with other encoder operations. Third, we also take many import design choices like whether to freeze embedding, how many layers of encoder, etc., into the search space, thus further reducing the human interventions in neural network design for NLU tasks. Fourth, we propose a series of strategies to ensure the search stability and performance. Experimental results show that our search process can obtain new models that are comparable or better than the existing network based baselines. In addition, ablation studies show that different parts of our search space are necessary and our proposed strategies indeed work for NAS.

In the future, we aim to further improve the search space for NLP tasks, as we have seen search space can play a significant role for vision tasks [20]. Meanwhile stable search and more evaluations on large-scale tasks are also of common interest to the community.

References

1. Ahn, N., Kang, B., Sohn, K.A.: Fast, accurate, and lightweight super-resolution with cascading residual network. In: ECCV (2018)
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv e-prints [arXiv:1409.0473](https://arxiv.org/abs/1409.0473), September 2014
3. Cai, H., Zhu, L., Han, S.: ProxylessNAS: direct neural architecture search on target task and hardware. arXiv preprint [arXiv:1812.00332](https://arxiv.org/abs/1812.00332) (2018)
4. Chen, Q., Zhu, X., Ling, Z., Wei, S., Jiang, H., Inkpen, D.: Enhanced LSTM for natural language inference. arXiv preprint [arXiv:1609.06038](https://arxiv.org/abs/1609.06038) (2016)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) (2018)
6. Ghiasi, G., Lin, T.Y., Le, Q.V.: NAS-FPN: learning scalable feature pyramid architecture for object detection. In: CVPR (2019)
7. Gong, J., Qiu, X., Wang, S., Huang, X.: Information aggregation via dynamic routing for sequence encoding. arXiv preprint [arXiv:1806.01501](https://arxiv.org/abs/1806.01501) (2018)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv e-prints [arXiv:1512.03385](https://arxiv.org/abs/1512.03385) (Dec 2015)
9. Khot, T., Sabharwal, A., Clark, P.: SciTaiL: a textual entailment dataset from science question answering. In: AAAI (2018)
10. Kim, S., Kang, I., Kwak, N.: Semantic sentence matching with densely-connected recurrent and co-attentive information. arXiv e-prints [arXiv:1805.11360](https://arxiv.org/abs/1805.11360), May 2018
11. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: ICML (2015)

12. Liu, C., et al.: Auto-DeepLab: hierarchical neural architecture search for semantic image segmentation. In: CVPR (2019)
13. Liu, H., Simonyan, K., Yang, Y.: DARTS: differentiable architecture search. arXiv preprint [arXiv:1806.09055](https://arxiv.org/abs/1806.09055) (2018)
14. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: EMNLP (2014)
15. Pham, H., Guan, M., Zoph, B., Le, Q., Dean, J.: Efficient neural architecture search via parameter sharing. arXiv preprint [arXiv:1802.03268](https://arxiv.org/abs/1802.03268) (2018)
16. Pham, H., Guan, M.Y., Zoph, B., Le, Q.V., Dean, J.: Efficient neural architecture search via parameter sharing. arXiv e-prints [arXiv:1802.03268](https://arxiv.org/abs/1802.03268) (Feb 2018)
17. So, D.R., Liang, C., Le, Q.V.: The evolved transformer. arXiv preprint [arXiv:1901.11117](https://arxiv.org/abs/1901.11117) (2019)
18. Socher, R., et al.: Recursive deep models for semantic compositionality over a sentiment treebank. In: EMNLP (2013)
19. Tan, C., Wei, F., Wang, W., Lv, W., Zhou, M.: Multiway attention networks for modeling sentence pairs. In: IJCAI, pp. 4411–4417 (2018)
20. Tan, M., et al.: MnasNet: platform-aware neural architecture search for mobile. arXiv e-prints [arXiv:1807.11626](https://arxiv.org/abs/1807.11626), July 2018
21. Vaswani, A., et al.: Attention is all you need. In: NIPS (2017)
22. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: GLUE: a multi-task benchmark and analysis platform for natural language understanding. arXiv preprint [arXiv:1804.07461](https://arxiv.org/abs/1804.07461) (2018)
23. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach. Learn. **8**(3–4), 229–256 (1992)
24. Zoph, B., Le, Q.: Neural architecture search with reinforcement learning. In: ICLR (2017)



AutoTrans: Automating Transformer Design via Reinforced Architecture Search

Wei Zhu^{2(✉)}, Xiaoling Wang¹, Yuan Ni², and Guotong Xie^{2,3,4}

¹ East China Normal University, Shanghai, China

² Ping An Healthcare Tech, Beijing, China

52205901018@stu.ecnu.edu.cn

³ Ping An Healthcare and Tech, Shanghai, China

⁴ Ping An International Smart City, Shenzhen, China

Abstract. Though the transformer architectures have shown dominance in many natural language understanding tasks, there are still unsolved issues for the training of transformer models, especially the need for a principled way of warm-up which has shown importance for stable training of a transformer, as well as whether the task at hand prefer to scale the attention product or not. In this paper, we empirically explore automating the design choices in the transformer model, i.e., how to set layer-norm, whether to scale, number of layers, number of heads, activation function, etc., so that one can obtain a transformer architecture that better suits the tasks at hand. RL is employed to navigate along search space, and special parameter sharing strategies are designed to accelerate the search. It is shown that sampling a proportion of training data per epoch during search help to improve the search quality. Experiments on the CoNLL03, Multi-30k and WMT-14 shows that the searched transformer model can outperform the standard transformers. In particular, we show that our learned model can be trained more robustly with large learning rates without warm-up.

Keywords: Transformer network · Neural architecture search · Reinforcement learning

1 Introduction

The transformer architecture [19] has achieved great success in not only machine translation but also many other natural language processing (NLP) tasks. Its popularity obtains further increase with the introduction of BERT [5]. Meanwhile, there are also many criticisms to the transformer. First, transformers often require a warm-up step to stabilize model training, which is directly affected by how the layer normalization is applied in the transformer architecture. In [19], layer-norm is applied after residual connection (post-LN transformer). [20] place layer-norm before multi-head attention and activation (prev-LN transformer).

Recently [22] gives a theoretical explanation of the advantages of the prev-LN transformer. However, these two strategies are the only existing combinations of layer-norms. Second, [23] points out that some tasks like named entity recognition (NER) may prefer not to scale the attention product. In addition, there are many design choices (or hyper-parameters) in the transformer architecture, e.g., the number of attention heads, layers [18], and the number of dimensions in the positional feed-forward module, etc. There exists little guidance on how to achieve an optimized transformer architecture for different tasks. Manual tuning or simple heuristic search is time consuming and computationally expensive without any grounded guarantee. Due to its wide applications, optimizing the transformer architectures for specific tasks can be of great importance.

Network Architecture Search (NAS) has achieved great success in image recognition. NAS has helped to discover better models for a variety of vision tasks, from image classification [3], semantic segmentation [11], to object detection [7], etc. Despite its success in vision, there are not enough work to study NAS for NLP. Some efforts have also been invested in searching for sequence models [14, 24]. In these cases, it has always been to find better RNN architectures. There are few studies in applying NAS into improving the standard transformer architectures. [17] employs evolution algorithms for search, but their focuses are to combine convolutions and multi-head attention operations. In addition, the search process they conduct requires enormous computations that is forbidding to most researchers or NLP practitioners. In this work, we propose a more efficient yet effective methodology to improve upon the standard transformer architectures.

When trying to improve upon the standard transformers, the above literature fall short in the following aspects.

- 1) For the settings of layer-norms, no previous literature have proven theoretically or empirically the existing two solutions are optimal.
- 2) Many other import design choices of transformers architectures like whether to scale, whether to put attention to encoder module before or after self attention, number of layers is not addressed by NAS literature.
- 3) Existing work on NAS for transformers is extremely time-consuming, making it in-practical.

In this work, we experiment on making the design choices in the transformer model automatically, i.e., how to set layer-norm, whether to scale, where to put encoder attention, number of heads, number of layers, activation function, etc., so that one can obtain a transformer architecture that better suits the tasks at hand. To navigate on our search space, we employ reinforcement learning (RL) strategy, or more specifically, ENAS by [14], and we design the specialized parameter sharing strategies for multi-head attention to help speeding up the search process. And we propose to sample a proportion of the training data at each search epoch, as a way of regularization and speed-up. Experiments on the CoNLL03, Multi-30k, WMT-14 dataset shows that the searched transformer models can outperform the standard transformer models significantly. And we

will show that this performance advantage is persistent across different learning rates. Note that since they are two phases in ENAS search, the top-ranked model at the search phase may not be the best one, but it can still reliably outperform the standard transformers.

The contributions of the paper can be summarized as:

- We develop a comprehensive search space to improve transformer architecture, especially for the positions of layer-norms.
- We develop efficient search for new transformer architectures (e.g. 1.5 GPU hours on Multi-30k), by employing RL strategy and designing specialized parameter sharing strategies for multi-head attention.
- The learned models outperforms the standard transformers, and this performance gain is robust under different learning rates.

2 Related Work

The field of neural architecture search (NAS) has attracted a lot of attentions in the recent years. The goal is to find automatic mechanisms for generating new neural architectures to replace conventional handcrafted ones, or automatically deciding optimal design choices or hyper-parameters instead of manually tuning [2]. Recently, it has been widely applied to computer vision tasks, such as image classification [3], semantic segmentation [11], object detection [7], super-resolution [1], etc. However, NAS is less well studied in the field of natural language understanding (NLU). Recent works [12, 14, 24] search new recurrent cells for the language modeling (LM) tasks. The evolved transformer [17] employs an evolution-based search algorithm, and the vanilla transformer as the initial population, it generates a better transformer architecture that consistently outperform the vanilla transformer on 4 benchmark machine translation tasks.

Our work also focuses on the transformer architecture, but the difference with the evolved transformer is clear. First, evolved transformer emphasize on combining convolution operation and multi-head attentions, while our work is to optimize the settings of layer-norms, whether to scale, where to place the encoder attentions in the decoder, number of layers, etc., such that the model can converge and generalize well without warm-up. Second, we employ a special designed parameter sharing strategies, and we propose to sample a proportion of training data per search step, such that the search time cost can significantly reduce.

Our work is also closely related to a line of work that try to modify and improve the transformer architecture. Sparse transformer [4] uses a sparse alternative of softmax to reduce the head size of self attention. Star Transformer [8] uses an intermediate center node to change the fully-connected attention to a more sparse one, thus making the model lighter and can perform better on a series of small or medium sized datasets. The recent Reformer [9] uses local sensitive hashing and reversible transformer to significantly reduce the time and space complexity of transformer architectures. Our work contributes to the literature by including many design choices that are ignored by literature into our search space, and making the search for better transformers automatic and efficient.

3 Search Space Design

Now we discuss our search space in detail. Since our goal here is to optimize the transformer architecture, we keep its main bone structure, as shown in Fig. 1.

First, Fig. 1 depicts the possible positions for layer-norms, where $LN - i$ ($i < 7$ for the encoder, and $i < 10$ for the decoder) in yellow boxes means a layer-norm can be put at position i .¹ The second aspect is whether to scale at the multi-head attention. [23] suggest not to scale results in sparser attention and thus help to improve the transformer’s performance on NER tasks. Third, the number of layers is also import as it can not only affect performance during training, but also how many GPU resources are needed for deploying the model. In addition, it is common to set the number of layers in encoder to be equal to that in the decoder [19], however in [18] the decoder has fewer layers than the encoder. In this paper, we let the search procedure to decide whether to have different numbers of layers in the encoder and decoder. The fourth design choice we include in our search space is whether the attention to encoder module is placed after the self-attention or before.

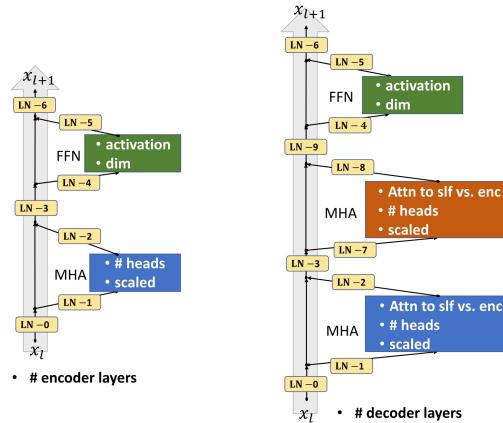


Fig. 1. The architecture for AutoTrans.

Similar to [17], the search space also include the number of heads, and the activation function used in the positional feed-forward layer, and the relative dimension for the intermediate hidden states. The activation function we consider is listed in Table 1.

¹ If layer-norm is put at position 0, then there is no need for layer-norm at position 1. This also holds for position 3 and 4, 9 and 4.

Table 1. Activation functions in the search space.

Name	Function
relu	$\max(x, 0)$
leaky_relu	$x \text{ if } x \geq 0 \text{ else } 1e-2 * x$
elu	$x \text{ if } x \geq 0 \text{ else } e^x - 1$
swish	$x * \text{sigmoid}(x)$
gelu	$0.5 * x * (1 + \text{erf}(x/\sqrt{2}))$
gelu_new	$0.5x(1 + \tanh(\sqrt{2/\pi}(x + 0.044715x^3)))$

Now we formally introduce our search space. For the encoder, the search space is as follows.

- Layer-norm at position i ($i < 7$) = True, False;
- Embedding layer-norm = True, False;
- Final output layer-norm = True, False;
- Number of self-attention heads = 1, 2, 4, 8, 16;
- Attention scaled = True, False;
- Activation = relu, leaky_relu, elu, swish, gelu, gelu_new;
- Relative dimension = 0.5, 1, 2, 4, 8²;
- Number of layers = 1, 2, 3, 4, 6, 9;

For the decoder, most of the search space is the same, except the following three items.

- Layer-norm at position i ($i < 10$) = True, False;
- Encoder attention after self-attention = True, False;

In total, our search space contains $1.5e+6$ number combinations of possible transformer encoder architectures and $1.2e+8$ for encoder-decoder architectures, which is quite a large search space. Next, we will show how to navigate through this enormous search space and obtain architectures that are better than standard transformers efficiently.

4 Architecture Search

We elaborate on the search algorithm that exploits the defined search space, the parameter sharing strategies and data sampling strategy controlling the search time within a GPU day.

² Here the relative dimension being equal to 0.5 means halving the hidden dimensions.

4.1 Search Algorithm

We employ a controller to do a guided exploitation in search space, which is similar to ENAS [14]. The controller is an LSTM network with 100 units, with parameters θ . The output hidden state is fed into a classifier to decide the action at each step. The shared parameters of the child models are denoted by ω , which will be discussed in detail in the next subsection.

The architecture search procedure consists of two interleaving phases. The first phase trains ω , the shared parameters of the child models, on a pass through the training data set. In ENAS [14], this step use the whole pass of the training data set. However, we argue that for each pass, randomly sampling a proportion of the training data not only saves time, but also provides extra regularization so that the parameters do not over-fit and the search can obtain better architectures. The second phase trains θ , the parameters of the controller, via optimizing the expected reward function using the REINFORCE algorithm [21]. The reward function is the negative of the perplexity on the valid set for the machine translation task, and F1 score for NER.

4.2 Deriving Architectures

We discuss how to derive novel architectures from a trained ENAS model. We first sample several models from the trained policy $\pi(m, \theta)$. For each sampled model, we compute its reward on the validation set. Then we take the top model(s) with the highest rewards to re-train from scratch. The number of top models to select is based on our computational resources, which will be shown in detail in the next section. We find that it is not guaranteed that the top-ranked model generated here will turn out to be the best model in certain tasks. But, selecting only a few top models to train from scratch makes the search procedure reasonably economical, and is shown to be able to improve the transformer architecture.

4.3 Cross-operation Parameter Sharing

In this work, the embeddings for the source language and the target language are shared for all the child models. For more efficient parameter sharing and ease of training, we constrain that the hidden dimension of each attention head is equal to the hidden size divided by the number of heads, so that multi-head attention block with different number of heads have parameters of the same size, and thus are possible to share the query, key and value matrices. Note that the encoder and decode does not share weights.

4.4 Cross-layer Parameter Sharing

Recently ALBERT [10] shows that cross-layer parameter sharing can provide regularization for training and stabilize the gradients, thus are beneficial for training deep models. In this paper, unless otherwise stated, during search the parameters in the previous transformer layer is shared to the next one. Cross-layer parameter sharing is beneficial in the scenario of NAS since it notably reduce the number of shared weights, thus greatly accelerating the search process.

5 Experiments and Results

For each search or evaluation, we assign 1 T V100 GPU card(s) for CoNLL03 and Multi-30k, and 8 for WMT-14.

5.1 Datasets

Table 2. Overview of used datasets in experiments.

Dataset	Task	Train	Dev	Test	Metrics
Multi-30k (en-de)	MT	30k	1.03k	1.0k	BLEU
WMT-14 (en-de)	MT	4.5M	39k	3k	BLEU
CoNLL2003	NER	14k	3.2K	3.4k	F1

We conduct experiments on two different tasks with 4 benchmark datasets, whose statistics are shown in Table 2. To verify the validity of our method on tasks of different scales, we select three machine translation tasks, Multi-30K [6] and the standard WMT-14 (English-German) dataset, which represent machine translation tasks of different sizes. We also experiment on CoNLL2003 [15], a benchmark NER datasets, to showcase that our method works under different NLP tasks.

Multi-30K. This task involves translating English sentences that describe an image into German.

WMT-14. It consists of about 4.5 million EN-DE sentence pairs. Sentences were encoded using byte-pair encoding [16], which has a shared source target vocabulary of about 37000 tokens. We use *newstest2013* for validation and *newstest2014* for test, which is in consistence with [19].

CoNLL2003. This dataset consists of 200k training words which have been annotated as Person, Organization, Location, Miscellaneous, or Other (non-named entity).

5.2 Architecture Search Protocols

During search phase, the interleaving optimization process is run 100 times. For each search epoch, a proportion r of the train data is passed to a child model, where $r = 0.05, 0.2, 0.5$ or 1 . For WMT-14 task, we only consider $r = 0.2$. For the CoNLL03 task, we use the pre-trained embedding from Glove [13] (840B, 300d), and the 300-d embedded input is reshaped to 512d with a separable convolution with kernel size 1. For the three MT tasks, the embedding is randomly initialized, and the dimensions for the embedding and for the hidden states are all set to 512.

Due to the resource limitation, for the WMT-14 dataset, we limit the number of layers to be less or equal to 6, so that the size and number of parameters of the new transformers will not be larger than the transformer base setting in [19], thus for comparison of performances, we will only compare with transformer base. The hidden dim for the controller is set to 100. After manually fine-tuning, the learning rate for the search is set at 1e-4 for CoNLL03, and 1e-3 for Multi-30k and WMT-14. The batch-size is set at 64 per GPU.

After the search phase, 30 model architectures are sampled from the trained controller, and they are ranked via their performance on the valid data when they are initialized using the shared parameters. Then the top-ranked 5 models (2 models for WMT-14 task) are trained from scratch to convergence on the whole training data of the task to formally evaluate their performances. The training is also repeated for n runs to calculate the fluctuation of performances. n is set in consideration of replication and our resource limitations. For CoNLL03 and Multi-30k, n is set to be 10. For WMT-14, n is set to be 5.

To compare our methods with random search, for task CoNLL03 and Multi-30k, we randomly samples 7 different models, since the GPU time for training 7 models is slightly larger than an entire search and evaluation process. Due to the same reason, we also randomly samples 3 different models for evaluation on the WMT-14 task. We train them from scratch, and report the performance of the best dag as the performance of this random search run. The results of random search will be the average of 5 such runs.

The key hyper-parameters for all the learned models and baseline transformers are learning rate, and for standard transformers the number of layers, number of heads, relative dimension are also considered. Learning rate is selected from 3e-3, 1e-3, 1e-4, and the other hyper-parameters are consistent with that in our search space. The optimal hyper-parameters are determined via exhaustive search over the search space.

5.3 Main Results

Results on CoNLL-03. First, we report the results on the CoNLL 2003 task in Table 3. The prev-LN transformers perform slightly better than the post-LN version, achieving around 67.89 F1 on the test set. Random search on our search space obtains worse average results, and the performances are quite volatile. Now we look at our learned models. The top-ranked dag $AT_{conll,0.2,0}$ by the search procedure when using only 20% of the training data is depicted in Fig. 2(a). It only uses layer-norm once at the beginning of the transformer block, and it uses a relative dim of 8 and stack two transformers layers. Note that it does not scale at the MHA. This model significantly outperforms the two versions of standard transformers by achieving 68.81 F1 score on the test set. The best model the search gives out is $AT_{conll,0.5,1}$, a 4-layer transformer discovered when using 50% of the training data for search. If the resources allows, training more top-rated dags can help to discover better architectures. The results show that our search method is able to discover new transformer architectures efficiently.

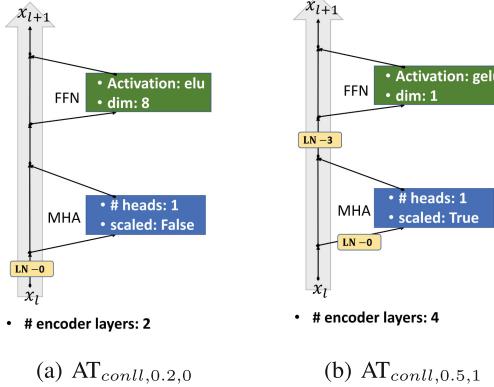


Fig. 2. New transformer architectures learned on CoNLL03.

Table 3. Results on the CoNLL2003 dataset.

Model	dev F1 (%)	test F1 (%)
prev-LN Transformer	75.77 ± 0.354	67.89 ± 0.204
post-LN Transformer	75.46 ± 0.217	67.11 ± 0.262
random search	74.89 ± 0.768	66.7 ± 0.645
$AT_{conll,0.2,0}$	78.15 ± 0.178	68.81 ± 0.189
$AT_{conll,0.5,1}$	78.64 ± 0.204	69.75 ± 0.197

Table 4. Results on the Multi-30k dataset.

Model	dev ppl	test BLEU
prev-LN Transformer	22.33 ± 0.123	29.47 ± 0.426
post-LN Transformer	19.64 ± 0.569	31.20 ± 0.089
random search	24.76 ± 1.987	28.08 ± 1.863
$AT_{multi30k,0.2,0}$	17.54 ± 0.090	33.55 ± 0.310
$AT_{multi30k,1,1}$	16.63 ± 0.143	34.56 ± 0.325

Results on Multi-30k. The results on the Multi-30k English-German translation task is reported on Table 4. Random search on our search space can not result in good performances and is quite unreliable in finding good architectures. The top-ranked architecture at the search phase when using 20% of the training data for search achieves average ppl of 17.63 on dev set and BLEU score of 33.55, which outperforms the two standard transformers. However, this is not the best architecture we obtained. When using all the train data per search epoch, the second best dag by the search phase is the best architecture we find. Figure 3(a) and 3(b) depict the two learned models. Note that in the two models: i) a layer-norm is placed right after the multi-head self attention, and after the

FFN module; ii) the decoder has more layers than the encoder layer; iii) the attn to encoder module is placed before the self attn in the decoder layer. The two learned models verify the necessity of our search space design.

Results on WMT-14. For WMT-14 EN-DE task, the results for prev-LN and post-LN transformers are referenced from [20] and [19]. As we can see in Fig. 4, the learned models AT_{wmt14,0.2,0} and AT_{wmt14,0.2,1} places encoder attention after self-attention. But now the best discovered models on WMT-14 tends to place layer-norm before the self-attention module, and before and after the feed-forward layer. Here we find that the decoder has less layers than the encoder. According to Table 5, both models outperform the standard transformers significantly, and performs comparably with the evolved transformer. Note that the evolved transformer includes branches in its architecture to allow more variable feature extraction, and the GPU time cost is 1000 times more than ours. The result shows that our approach is efficient and effective on machine translation tasks of various sizes.

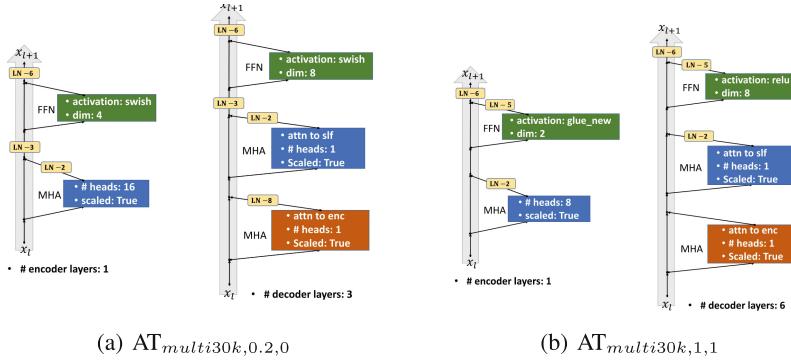


Fig. 3. New transformer architectures learned on Multi-30K.

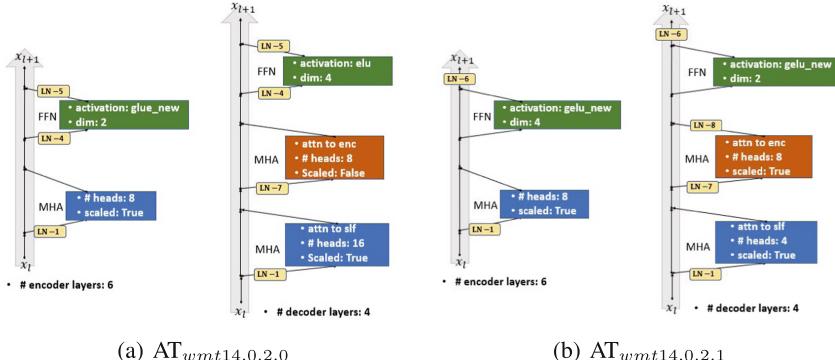


Fig. 4. New transformer architectures learned on WMT-14.

Table 5. Results on the WMT-14 dataset.

Model	dev ppl	test BLEU
prev-LN Transformer (base)	–	27.1
post-LN Transformer (base)	–	27.3
Evolved Transformer (base)	4.03	28.4
random search	4.56 ± 0.764	26.85 ± 0.698
AT _{wmt14,0.2,0}	4.18 ± 0.092	27.68 ± 0.164
AT _{wmt14,0.2,1}	4.09 ± 0.076	28.05 ± 0.115

5.4 Effects of Proportions of Training Data

We conduct a series of experiments on the Multi-30k dataset, trying to study the effects of using only a proportion of training data in a search epoch. Table 6 gives out the results for the top-ranked models for the proportion of training data being 5%, 20%, 50% and 100%, respectively. When using only 20% of the data for search, we can already learn a good architecture that outperforms the standard transformer. Using the whole training data for search can generate better performance, but the search cost is higher. Note that, when given only 5% of the training data per search epoch, the controller fails to obtain a better model. We believe the intuition behind this phenomenon is that when fed with not enough data, the reward signals the controller receives can not well represent a model’s performance, thus making it difficult to design good models for the dev set. However, our experiments shows that when the resources is limited, one can use a proper proportion of data for search (Table 7).

Table 6. Multi-30k: different proportions of train data for search.

Model	dev ppl	test BLEU
prev-LN Transformer	22.33 ± 0.123	29.47 ± 0.426
post-LN Transformer	19.64 ± 0.569	31.20 ± 0.089
AT _{multi30k,0.05,0}	21.85 ± 0.162	29.39 ± 0.395
AT _{multi30k,0.2,0}	17.54 ± 0.090	33.55 ± 0.310
AT _{multi30k,0.5,0}	17.95 ± 0.205	31.12 ± 0.363
AT _{multi30k,1,0}	17.27 ± 0.095	33.72 ± 0.232

Table 7. Multi-30k: different learning rates for model training.

Model	dev ppl	test BLEU
lr = 1e-3		
prev-LN Transformer	23.66 ± 0.686	29.41 ± 0.405
post-LN Transformer	19.64 ± 0.569	31.20 ± 0.089
AT _{multi30k,1,1}	16.63 ± 0.143	34.56 ± 0.325
lr = 3e-3		
prev-LN Transformer	22.33 ± 0.123	29.47 ± 0.426
post-LN Transformer	34.44 ± 9.31	22.62 ± 6.462
AT _{multi30k,1,1}	17.358 ± 0.290	33.253 ± 0.199
lr = 1e-4		
prev-LN Transformer	25.07 ± 0.239	28.97 ± 0.422
post-LN Transformer	22.25 ± 0.166	30.29 ± 0.202
AT _{multi30k,1,1}	18.50 ± 0.094	34.08 ± 0.481

5.5 Effects of Different Learning Rates on the Learned Architecture

We study how different learning rates affect the performances of our learned architecture AT_{multi30k,1,1}, which is obtained by setting the learning rate to be 1e-3 during search. The learning rate is set to be 3e-3, 1e-3, 1e-4, and as always, no warm-up is used for all models. The post-LN transformer is the most sensitive to learning rate, and prev-LN transformer is robust with different learning rate, but it does not result in good performance. Our learned model AT_{multi30k,1,1} is affected by learning rate, but it outperforms the two baselines significantly.

5.6 Effects of Learning Rate on Search

As shown in Table 8, learning rate affects the search results significantly. For different learning rates, the searched models given by using 20% of training

Table 8. Multi-30k: different learning rates to search.

Model	dev ppl	test BLEU
lr = 1e-3		
AT _{multi30k,0.2,0}	17.54 ± 0.094	33.55 ± 0.310
lr = 3e-3		
AT _{multi30k,0.2,0}	18.62 ± 0.115	33.08 ± 0.385
lr = 1e-4		
AT _{multi30k,0.2,0}	21.29 ± 0.079	32.88 ± 0.286

data are different, and the performance difference is significant. Thus, how to incorporate the learning rate into the search space or search for models that are robust to different learning rates is an important issue we would like to further investigate.

6 Conclusions and Discussions

In this work, we have investigated how neural architecture search can improve the standard transformer architectures efficiently. We focus on the design choices that are not well studied in literature, such as how to place layer-norms, number of layers, how to place encoder attention in the decoder, etc. By applying parameter sharing and training data sampling, we can obtain improved transformer models within a couple of hours on a single GPU. Our experiments on CoNLL03 and Multi-30K shows that our methodology works on different tasks of different sizes. In addition, our learned model can perform more robustly when trained with different learning rate.

There are possibilities for future work. First, how to make the transformer architectures more robust to different learning rate or minimize the effects of learning rate is an important direction. Second, although the top-ranked model during search is better than standard transformers, it may not be the best one. Thus, minimizing the gap between the search and training is a challenging and worth efforts.

References

1. Ahn, N., Kang, B., Sohn, K.A.: Fast, accurate, and lightweight super-resolution with cascading residual network. In: ECCV (2018)
2. Bergstra, J.S., Bardenet, R., Bengio, Y., Kégl, B.: Algorithms for hyper-parameter optimization. In: Advances in Neural Information Processing Systems, pp. 2546–2554 (2011)
3. Cai, H., Zhu, L., Han, S.: ProxylessNAS: direct neural architecture search on target task and hardware. arXiv preprint [arXiv:1812.00332](https://arxiv.org/abs/1812.00332) (2018)
4. Correia, G.M., Niculae, V., Martins, A.F.: Adaptively sparse transformers. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 2174–2184 (2019)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) (2018)
6. Elliott, D., Frank, S., Sima'an, K., Specia, L.: Multi30K: multilingual English-German image descriptions, pp. 70–74 (2016)
7. Ghiasi, G., Lin, T.Y., Le, Q.V.: NAS-FPN: learning scalable feature pyramid architecture for object detection. In: CVPR (2019)
8. Guo, Q., Qiu, X., Liu, P., Shao, Y., Xue, X., Zhang, Z.: Star-transformer. arXiv preprint [arXiv:1902.09113](https://arxiv.org/abs/1902.09113) (2019)
9. Kitaev, N., Kaiser, L., Levskaya, A.: Reformer: the efficient transformer. arXiv preprint [arXiv:2001.04451](https://arxiv.org/abs/2001.04451) (2020)

10. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: ALBERT: a lite BERT for self-supervised learning of language representations. arXiv preprint [arXiv:1909.11942](https://arxiv.org/abs/1909.11942) (2019)
11. Liu, C., et al.: Auto-DeepLab: hierarchical neural architecture search for semantic image segmentation. In: CVPR (2019)
12. Liu, H., Simonyan, K., Yang, Y.: DARTS: differentiable architecture search. arXiv preprint [arXiv:1806.09055](https://arxiv.org/abs/1806.09055) (2018)
13. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: EMNLP (2014)
14. Pham, H., Guan, M.Y., Zoph, B., Le, Q.V., Dean, J.: Efficient neural architecture search via parameter sharing. In: ICML (2018)
15. Sang, E.F., De Meulder, F.: Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. arXiv preprint [arXiv:cs/0306050](https://arxiv.org/abs/cs/0306050) (2003)
16. 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 (Volume 1: Long Papers), Berlin, Germany, pp. 1715–1725. Association for Computational Linguistics, August 2016. <https://doi.org/10.18653/v1/P16-1162>, <https://www.aclweb.org/anthology/P16-1162>
17. So, D.R., Liang, C., Le, Q.V.: The evolved transformer. arXiv preprint [arXiv:1901.11117](https://arxiv.org/abs/1901.11117) (2019)
18. Song, K., Tan, X., Qin, T., Lu, J., Liu, T.Y.: MASS: masked sequence to sequence pre-training for language generation. In: International Conference on Machine Learning, pp. 5926–5936 (2019)
19. Vaswani, A., et al.: Attention is all you need. In: NIPS (2017)
20. Wang, Q., et al.: Learning deep transformer models for machine translation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, pp. 1810–1822. Association for Computational Linguistics, July 2019. <https://doi.org/10.18653/v1/P19-1176>, <https://www.aclweb.org/anthology/P19-1176>
21. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach. Learn. **8**(3–4), 229–256 (1992)
22. Xiong, R., et al.: On layer normalization in the transformer architecture (2020). <https://openreview.net/forum?id=B1x8anVFPr>
23. Yan, H., Deng, B., Li, X., Qiu, X.: TENER: adapting transformer encoder for named entity recognition. arXiv e-prints [arXiv:1911.04474](https://arxiv.org/abs/1911.04474) (Nov 2019)
24. Zoph, B., Le, Q.: Neural architecture search with reinforcement learning. In: ICLR (2017)



A Word-Level Method for Generating Adversarial Examples Using Whole-Sentence Information

Yufei Liu^(✉), Dongmei Zhang, Chunhua Wu, and Wei Liu

Beijing University of Posts and Telecommunications, Beijing, China
`{liuyuf,zhangdm,wuchunhua,liuw}@bupt.edu.cn`

Abstract. Adversarial examples mislead the deep neural networks (DNNs) by adding slight human-imperceptible perturbations to the input, they reveal the vulnerability of DNNs and can be applied to improve the robustness of the model. Recent work generates adversarial examples by performing word-level substitutions. However, these methods can lead to contextually inappropriate or semantically deviant substitutions because they do not take full advantage of the whole-sentence information and are inefficient in searching. The aim of this study is to improve current methods to enhance the effectiveness of adversarial examples. This study proposes an adversarial example generation method based on an improved application of the masked language model exemplified by BERT. The method injects fuzzy target word information into BERT to predict substitutes by regularizing its token embedding, which empowers BERT to integrate whole-sentence information, and then searches for adversarial examples within the substitute space using beam search with the guidance of word importance. Exhaustive experiments show that it not only significantly outperforms state-of-the-art attack methods, but also has high application value as it can generate fluent and natural samples with minimal perturbation. The work indicates that the method proved to be both effective and efficient in generating adversarial examples.

Keywords: Adversarial example generation · Word substitution · BERT application

1 Introduction

The widespread use of deep neural networks (DNNs) has brought the artificial intelligence security concerns into the limelight. Recently, researchers found them vulnerable to adversarial examples, which are maliciously crafted by adding some human-imperceptible perturbations to the inputs, fooling the model to give false outputs with high confidence [23]. Adversarial examples limit the further development of DNNs, especially for many security-sensitive applications, such as spam detection and toxic comments detection systems [5, 9]. From another perspective, adversarial examples can be used to improve model robustness [7].

Compared to images, textual adversarial example generation remains a challenge because the discrete nature of text makes it difficult to generate perturbations that are truly imperceptible to humans. Therefore, the goal in text domain is to generate samples that meet three requirements: (1) to have the consistent semantics with the original sample; (2) to read fluently and naturally; (3) to make the target model predict incorrectly. To meet these goals, word-substitution method has been widely studied in textual adversarial example generation tasks.

The core of word substitution methods lies in the screening of substitutes with the aim of semantic consistency and linguistic fluency. There are currently two categories of methods: target-based methods and context-based methods. Target-based methods take synonyms of the target word as its candidates, using synonym repository [19], word embeddings [1, 11] and the sememe knowledge base [25]. This method is static and context-agnostic, leading to contextually inappropriate substitutions. On the other hand, context-based methods use language models to predict substitutes [6, 13, 14]. However, in both sequential and masked language models, the original word is not visible when predicting, as a result of which, semantic deviations always occur. The second stage is to search for a successful adversarial sample in the candidate space. However, the efficiency and effectiveness of existing models are still far from satisfying.

In this paper, a novel word-level method for generating adversarial examples based on whole-sentence information is proposed to address the limitations of substitution effectiveness and search efficiency. Compared with target-based and context-based methods, our method is able to integrate the target word and contextual information to produce globalized substitutions so as to ensure linguistic fluency and semantic consistency. Meanwhile, it has an efficient and effective search method. First, we use an improved application of BERT masked language model (BERT-MLM) to predict substitute words. Notably, based on the original BERT-MLM, a novel, simple and effective target word information injection method is proposed, which can effectively influence the prediction process to generate globalized substitutions. After that, a method based on beam search with the guidance of word importance is proposed, greatly reduces the probability of falling into local extrema and improves the search efficiency. Comprehensive experiments show our method outperforms the state-of-the-art methods overall. The main contributions of this work are summarized as follows:

- A simple and effective adversarial example generation method is proposed and successfully fool the target model in a black-box scenario.
- The shortcoming exposed by the mask language model (MLM) in generating adversarial examples is discussed: the word information injection problem. A fuzzy word information injection method is proposed to solve it, which enables the MLM to generate global perturbations. And a novel heuristic beam search method is also proposed.
- The algorithm is evaluated on three state-of-the-art models on four representative text classification datasets, and it achieved the state-of-the-art performance. Automatic and human evaluations show that it's able to generate valid, natural and semantically consistent adversarial examples.

2 Related Work

The textual adversarial attack models can be classified into character-level, word-level, and sentence-level categories according to the granularity of modification [2]. Character-level perturbations [4,5] are neglectful, but can be easily detected by spell checking [17,19]. Sentence-level models [10,20] make a larger perturbation and the changes in the semantics are difficult to control [20].

Current successful adversarial attacks usually use word substitution based models. Textfooler proposed by [11] selects candidate substitutes by finding TOP-K nearest neighbors of the target word in the word embedding space, and then performs a greedy search under the constraints of heuristic rules. They were the first to confirm the vulnerability of BERT to adversarial attacks. Textfooler is a target-based method, which is context-agnostic and often leads to unnatural output. Contextualized methods based on language models address this problem, and in particular, BERT-based methods show good results. [6,13,14] did similar work by using the “mask-then-infill” pattern to generate adversarial perturbations and using greedy search to search for successful adversarial examples. Context-based approaches are able to generate fluent and natural output, but fail to incorporate the control of semantic similarity in the substitute screening process, causing semantic deviations and even changes in the true label, which means the failure of the attack. In comparison, Our approach aims at generating global perturbations using whole-sentence information, making the substitutions both contextually appropriate and relevant to the original word. Besides, most current methods use greedy search in the second step [6,11,13,14,19], but the effectiveness is low. [1,25] made an attempt of genetic algorithm and particle swarm search algorithm, but they’re too costly to be used for practical applications. In contrast, our method balances the effectiveness and efficiency well.

3 Methodology

In this section, we describe our model in detail. It consists of two main steps: selecting candidate substitutes and searching for adversarial examples.

3.1 Selecting Candidate Substitutes

Pre-trained masked language models exemplified by BERT [3] emerged to refresh the performance of a large number of NLP tasks. For adversarial example generation tasks, the large corpus gives BERT the capability to generate fluent and natural contextual perturbations. However, the nature of the masked language model makes the replaced target word invisible to the model, which limits the performance of BERT on adversarial example generation tasks. So how to inject target word information into the prediction process? We refer to this problem as the “Word Information Injection Problem”. An intuitive idea is to discard the [MASK] token and feed the target word into the model directly when predicting, and then select the words with TOP-K predicted probability in the output as

candidates. So we firstly experimented on this idea. However, experiments show that for almost all examples, more than 99.999% of the predicted results for the target position fall into the target word itself and its inflections. Frustratingly, the inflections are usually ungrammatical for the context, and it's clearly unreliable to select substitutes from the tiny probabilities left.

Therefore, the balance of target word and contextual information needs to be considered in the word information injection problem. Inspired by [21, 24], we utilize the geometric properties of pre-trained word vectors. In [21, 24], they demonstrated that the length of the semantic vector can somehow reflect the strength of word meaning, and simply put, the shorter the length, the more ambiguous the word meaning. Therefore, We propose a novel, simple and effective method for fuzzy injection of target word information to help mask language models exemplified by BERT use balanced whole-sentence information while predicting. As shown in Fig. 1, the fuzzy injection operation is achieved by regularizing the semantic vector of the target word. The operation makes the injected semantic information of the target word ambiguous, thus forcing the model to predict using global information. Moreover, it can reduce the attention to w_i for the purpose of spreading the model's attention to the whole sentence.

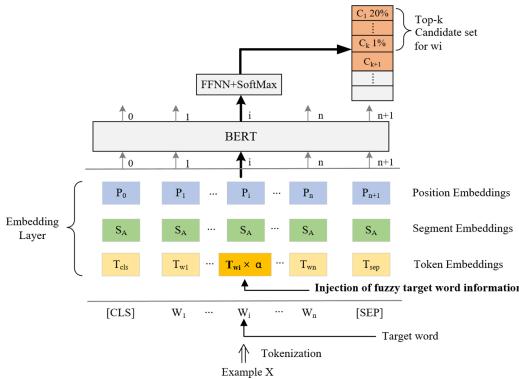


Fig. 1. Injecting fuzzy target word information when selecting word substitutes based on BERT-MLM.

Formally, for the target word w_i to be replaced in a sentence, let T_{w_i} denote its token embedding vector. We get T'_{w_i} by performing fuzzy operation of Eq. (1) on T_{w_i} , where α is a hyperparameter and $\alpha \in (0, 1)$. Each dimension of T_{w_i} is scaled down by α simultaneously. As shown in Fig. 1, the regularized T'_{w_i} is summed with the segment embedding and position embedding of w_i . Then, the summed word embedding is fed into BERT-MLM and the top-k predictions of the model for position i are used as candidate substitutes for w_i .

$$T'_{w_i} = \alpha \times T_{w_i} \quad (1)$$

Besides, we use the Glove vectors post-processed with counter-fitting method presented by [15] to filter out candidates that are antonyms of word w_i . Since BERT uses Bytes-Pair-Encoding (BPE) algorithm [22] for tokenization, for those sub-words, we use the whole word masking method for prediction.

3.2 Searching for Adversarial Examples

The substitutes of all words in the sentence constitute a candidate space, and this step aims to search for a successful adversarial example within this space.

Ranking by Word Importance. First, we ranked the importance of each word in the sample \mathcal{X} to find those words on which the classification model relies heavily, which are vulnerable words for the attack. In the black box scenario, we separately mask each word in the sentence with [MASK] token and use Eq. (2) to calculate the importance score I_{w_i} of word w_i .

$$I_{w_i} = p(C(\mathcal{X}) = y) - p(C(\mathcal{X}_{-w_i}) = y) \quad (2)$$

Where $\mathcal{X} = \{w_1, \dots, w_i, \dots, w_n\}$ and \mathcal{X}_{-w_i} denotes the new sentence after w_i is masked by the [MASK] token. $p(C(\mathcal{X}) = y)$ denotes the prediction score for ground truth label y by the target model C for input \mathcal{X} . The importance score list of all words in sentence \mathcal{X} is a guide for heuristic search.

Searching with Beam Search Algorithm. We design a search method based on beam search to balance the efficiency and effectiveness. The initial root node of the search tree is the original sample, and performing one word substitution yields its child node. We denote the current node by the symbol \mathcal{X}' . We select the top \mathcal{B} unmodified words in \mathcal{X}' with the highest importance score to form the vulnerable word list $\mathcal{V}_{\mathcal{X}'}$, where \mathcal{B} denotes beam size, and then calculate the score sco_{s_k} for each substitute word s_k in candidate list \mathcal{S}_{w_i} of each $w_i \in \mathcal{V}_{\mathcal{X}'}$.

$$sco_{s_k} = p(C(\mathcal{X}') = y) - p(C(\widetilde{\mathcal{X}'}) = y) \quad (3)$$

Where $\mathcal{X}' = \{w_1, \dots, w_i, \dots, w_n\}$ and $\widetilde{\mathcal{X}'} = \{w_1, \dots, s_k, \dots, w_n\}$, sco_{s_k} represent the confidence drop of the correct label, indicating the gain from substituting w_i with s_k . For each level of the search tree, TOP- \mathcal{B} words with the maximum gain are replaced independently to form the next level. The termination condition is that a predicted label has been changed, then the attack succeeds, or the similarity of all leaf nodes to the original sample can't satisfy the similarity threshold ℓ or there are no unsubstituted words, then the attack fails. Notably, ℓ is used to limit the semantic consistency¹.

¹ In the experiments, ℓ was set to 0.7 for the IMDB and YELP datasets, and 0.5 for the MR and AG datasets, which is differentiated according to the text length, and short texts are relatively sensitive to perturbations. In addition, the size of each S_{w_i} was set to 50, which means that a candidate list of 50 words was chosen for each word. And the hyperparameter α was set to 0.3, \mathcal{B} was 5 in the following experiments.

4 Experiments

4.1 Setup

Datasets and Target Models. To evaluate our adversarial example generation method, we applied our method to different text classification tasks. (1) IMDB²: Movie review dataset for the sentiment classification task. (2) AG’s News [26]: News topic classification dataset, containing: World, Business, Sports, and Science/Technology. (3) MR [16]: Movie review dataset for the sentiment classification task. (4) Yelp Polarity [26]: Binary review classification dataset. Additional information about the dataset is shown in Table 1. Following [11, 14], we experiment with 1k randomly selected test samples from each test set. Three representative and state-of-the-art target models are selected for attack: BERT [3], word-CNN [12] and word-LSTM [8]. For all target models, we achieved similar accuracy scores to the original implementation on the original test sets.

Table 1. Some details of the dataset and the original classification accuracy for each target model on the original test set.

Dataset	TrainSet	TestSet	AvgLen	ClassNum	Acc (BERT)	Acc (LSTM)	Acc (CNN)
IMDB	25000	25000	215	2	92.7%	89.8%	89.2%
AG	30000	1900	43	4	94.2%	91.3%	91.5%
MR	9000	1000	20	2	90.4%	80.7%	78%
YELP	560000	38000	156	2	97.6%	96.0%	94.0%

Evaluation Metrics

Automatic Evaluation. Following metrics are used to evaluate the effectiveness of the attacks and the generated adversarial examples. (1) Attack Success Rate (%Acc): It is equal to 1 minus the post-attack accuracy, and is a crucial metric for judging the performance of an attack algorithm. (2) Perturbation rate (%P): Percentage of modified words to the total number of words. The fewer perturbations represent the higher consistency between adversarial examples and original examples. (3) Perplexity (PPL): The average perplexity of the adversarial examples, calculated with the pre-trained language model GPT-2 [18]. (4) Grammatical errors (GErr): Average number of increased grammatical errors per sample, calculated with LanguageTool³. Low PPL and GEErr represent high linguistic fluency and naturalness.

Human Evaluation. Human evaluation is used to further evaluate the validity of the adversarial examples. We randomly selected 100 samples from the IMDB dataset (wordCNN) and MR dataset (BERT) and their adversarial samples for

² <https://datasets.imdbws.com/>.

³ <https://www.languagetool.org/>.

the experiments. The following metrics are included: (1) True label consistency: Human evaluators judge whether a pair of samples belongs to the same label. (2) Linguistic naturalness: Human evaluators rate the naturalness of examples on a Likert scale of 1–5, with higher scores being better. (3) Semantic consistency: Evaluators assess whether the semantics of the adversarial example and the original example are consistent and score on a scale of 1 to 5, with higher scores being more consistent. Each task was performed by 5 human evaluators with a university education who completed the experiment independently.

Baseline Models. Three latest state-of-the-art attack methods based on word substitutions are baseline models: Textfooler [11], Bert-Attack [14] and BAE-R [6]. The three baseline models were previously introduced in Sect. 2. Briefly, Textfooler is a target-based method, while BERT-ATTACK and BAE-R are context-based methods. The main difference between the latter two is that in the process of finding substitutes, BERT-ATTACK uses the method of preserving the target word with BERT, which is similar to the discarding [MASK] token experiment we mentioned in Sect. 3.1, while BAE-R uses the method of masking the target word consistent with the original BERT.

Table 2. Results of attack success rate of attack models on different classification tasks. The optimal results on each task are shown in **bold**.

Target model	Attack model	IMDB	AG	MR	YELP
BERT	Textfooler [11]	85.3%	70.5%	82.4%	92.1%
	BERT-ATTACK [14]	86.8%	85.6%	86.9%	97.8%
	BAE-R [6]	86.8%	85.1%	77.9%	96.1%
	Our approach	88.4%	93.1%	94.2%	99.4%
wordCNN	Textfooler	100%	97.6%	97.1%	98.6%
	BERT-ATTACK	100%	99.0%	97.4%	99.6%
	BAE-R	99.9%	97.6%	89.1%	98.1%
	Our approach	100%	99.9%	97.7%	99.8%
wordLSTM	Textfooler	99.4%	84.8%	96.9%	97.2%
	BERT-ATTACK	99.6%	89.3%	96.9%	98.5%
	BAE-R	99.2%	85.2%	92.9%	96.7%
	Our approach	99.7%	96.0%	98.1%	99.7%

4.2 Results

The evaluation results are shown in Table 2, 3, 4 and 5. Overall, our method consistently outperforms the baseline models on all datasets and all models.

Table 3. Automatic evaluation results of the quality of the generated adversarial examples. The optimal results on each task are shown in **bold**.

Target	Attack model	IMDB			AG			MR			YELP		
		%P	PPL	GErr	%P	PPL	GErr	%P	PPL	GErr	%P	PPL	GErr
Bert	Textfooler	6.88	102.11	1.88	25.08	209.51	0.78	20.98	124.23	0.78	14.58	92.59	2.26
	Bert-attack	4.58	57.24	1.40	16.76	124.21	0.20	20.06	125.54	0.82	11.54	70.52	1.49
	BAE-R	5.09	55.14	0.88	16.13	99.80	0.10	19.88	103.07	0.69	11.30	64.75	1.06
	Our Approach	2.93	52.37	0.61	10.96	93.24	0.08	16.48	102.92	0.64	9.08	62.28	0.91
CNN	Textfooler	3.70	53.06	0.88	16.67	143.65	0.49	14.88	95.17	0.61	8.75	62.38	1.31
	Bert-attack	3.70	53.58	1.06	14.24	116.75	0.19	16.73	103.72	0.75	8.75	59.08	1.06
	BAE-R	4.33	52.50	0.79	16.00	105.90	0.10	16.79	97.28	0.66	9.63	56.68	0.81
	Our Approach	2.83	51.32	0.73	10.97	100.18	0.06	14.88	91.63	0.63	7.52	56.41	0.80
LSTM	Textfooler	5.39	58.25	1.51	22.98	189.91	0.75	15.01	92.88	0.25	11.27	73.78	1.76
	Bert-attack	4.63	57.62	1.19	18.11	136.00	0.41	16.21	97.65	0.71	9.71	63.72	1.18
	BAE-R	5.31	56.21	1.08	19.52	119.16	0.20	16.91	88.53	0.61	10.29	60.15	0.95
	Our Approach	3.81	55.32	1.02	16.22	118.94	0.18	13.89	87.67	0.55	8.53	60.05	0.84

Table 4. A typical adversarial example from YELP dataset (Unmodified part at the end of the example is omitted), which indicates our approach achieves the attack with the smallest, most natural perturbation.

Original Sentence: POS	Really good Chinese food. The duck and pork noodle soup is awesome. ...
TEXTFOOLER: NEG	Awfully advantageous Chinese food. The duck and pork noodle soup is awesome. ...
BAE-R: NEG	Pretty pretty Chinese rice. The duck and pork noodle soup is edible. ...
BERT-ATTACK: NEG	Really fine Chinese food. The duck and hog noodle soup is nice. ...
Our Method: NEG	Really okay Chinese food. The duck and pork noodle soup is awesome.

Attack Success Rate. As shown in Table 2, our method achieves the highest attack success rates in all experiments. Specifically, for those attacks where the baseline models perform poorly, such as the BERT model, the multi-classification dataset (AG), and the short-text dataset (MR), our approach significantly outperforms the baseline model. For those attacks where the baseline models perform relatively well, such as the wordCNN model, our model gains further improvement.

Validity of Adversarial Examples. As shown in Table 3, it is encouraging that our approach significantly outperforms the baseline overall. The results show that the method is able to generate more natural and fluent samples with minimal perturbations. From human evaluation, as shown in Table 5, almost all adversarial examples are very natural and their true labels are unchanged, thus ensuring the effectiveness of the attack. Finally, a typical sample is shown in Table 4.

Table 5. Human evaluation results on IMDB(wordCNN) and MR(BERT). %Lab, NatScore and SemCons indicate the label consistency rate, the average naturalness score, the semantic consistency score, respectively.

		%Lab	NatScore	SemCons
IMDB	Ori	94	4.51	4.13
	Adv		4.32	
MR	Ori	92	4.19	4.07
	Adv		4.05	

Table 6. Ablation study results of our approach, where -WS indicates that the word substitution method is changed to that shown in parentheses, -GS indicates that the search algorithm is changed to the greedy search.

Method	%Acc↑	%P↓	PPL↓	GErr↓
Our approach	93.1	10.96	93.24	0.08
- WS (Keep)	90.2	13.12	110.30	0.13
- WS (Mask)	87.7	12.98	95.14	0.10
- WS (Glove)	80.3	19.85	159.42	0.56
- GS	89.1	13.84	96.53	0.10
Textfooler	70.5	25.08	209.51	0.78
BERT-ATTACK	85.6	16.76	124.21	0.20
BAE-R	85.1	16.13	99.80	0.10

5 Analysis and Discussions

5.1 Ablation Analyses

In order to understand the improvement, we conduct an ablation test on AG dataset (BERT). According to Table 6, we observe that when the word substitution algorithm in our method is replaced with other algorithms, all the metrics decreased, which shows our fuzzy word information injection method improves the fitness of BERT in adversarial example generation. When replacing the search algorithm in our method with the greedy search, the metrics fall into the middle of our method and the baseline models, which again proves the validity of both steps. For BERT, the method of word masking always generates samples with better grammaticality and naturalness compared to the method of word keeping, but the latter leads to a higher attack success rate, which we believe is due to the fact that the substitutes generated by word keeping method, such as inflections, have similar semantic vectors to the original word, and are more able to bypass the restrictions of the sentence similarity model, however, they are not valid. In addition, the context-based approach clearly performs better than the target-based approach because the language model contains a large amount of semantic information and has the ability to generate good replacements.

In order to more intuitively demonstrate the advantages of the first part, a real sample and the substitutes selected by each method are shown in Table 7. It can be seen that our method is able to find the most suitable substitute words. In addition, we visualized the attention of different methods in Fig. 2. Obviously, our method smoothes the attention in predicting the target word “executive”.

Table 7. Substitutes selected by different methods for a real sample. The suitable substitutes are marked in red.

Original sentence: A forceful drama of an alienated executive who reinvents himself.	
Our approach	director, manager, businessman, ceo, producer
BERT(Keep)	ceo, executives, corporate, enterprise, chairman
BERT(Mask)	man, youth, child, artist, student
Synonym(Glove)	administrative, managerial, management, bureaucratic, governance

Table 8. Attack performance on BERT model and MR dataset before and after adversarial training.

	OriAcc	%Succ	%P
Original	90.4	93.5	16.79
+Adv training	88.3	79.2	19.13

5.2 Effect of Beam Size

As shown in Fig. 3, the model attack success rate increases with increasing beam size in general, which eventually stabilizes. Overall, a lower beam size can significantly improve the attack performance while maintaining a high efficiency. An interesting point is that according to Fig. 3 beam search is more effective for short texts, presumably because the candidate space for short texts is much smaller than for long texts, and the branches cut in each step by greedy search have a greater negative impact on the search results.

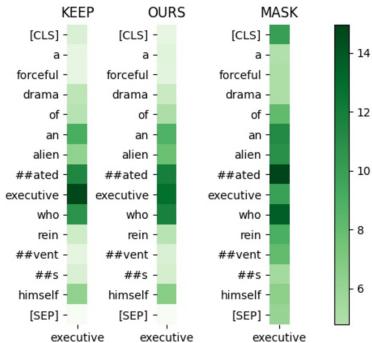


Fig. 2. The attention distribution of the BERT model in predicting the target word “executive”. Our method (middle) clearly leads to smoother attention.

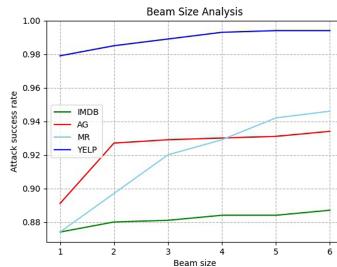


Fig. 3. Relationship between attack success rate and beam size

5.3 Adversarial Training

We mixed the original training set of MR dataset with their adversarial examples and used them to fine-tune the BERT model. The results in Table 8 show that adversarial training decreases the attack success rate and increases the perturbation rate, which indicates the target model is more robust. Moreover, adversarial training did not cause a marked decrease in the prediction ability of the model for clean samples. This suggests that the adversarial examples generated in this paper have the potential to be used to improve the robustness of DNNs.

6 Conclusion

In this paper, we study word-level adversarial attacks against state-of-the-art text classification models. A powerful method for adversarial example generation based on whole-sentence information is proposed. Extensive experiments demonstrate the superiority of this method, which achieves the state-of-the-art attack success rate and maintains the lowest perturbation rate. The generated adversarial examples achieve high semantic consistency and linguistic fluency, and are capable of helping models resist potential adversarial attacks. In the future, we will further investigate the application of adversarial examples generated by this method for adversarial defense. Moreover, we will study the interpretability of BERT token embeddings based on geometric properties.

References

1. Alzantot, M., Sharma, Y., Elgohary, A., Ho, B.J., Srivastava, M.B., Chang, K.W.: Generating natural language adversarial examples. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2890–2896 (2018)
2. Belinkov, Y., Glass, J.R.: Analysis methods in neural language processing: a survey. In: NAACL-HLT (1), pp. 3348–3354 (2019)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.N.: 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), pp. 4171–4186 (2018)
4. Ebrahimi, J., Lowd, D., Dou, D.: On adversarial examples for character-level neural machine translation. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 653–663 (2018)
5. Gao, J., Lanchantin, J., Soffa, M.L., Qi, Y.: Black-box generation of adversarial text sequences to evade deep learning classifiers. In: 2018 IEEE Security and Privacy Workshops (SPW), pp. 50–56 (2018)
6. Garg, S., Ramakrishnan, G.: BAE: BERT-based adversarial examples for text classification. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 6174–6181 (2020)
7. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: ICLR 2015 : International Conference on Learning Representations 2015 (2015)

8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
9. Hosseini, H., Kannan, S., Zhang, B., Poovendran, R.: Deceiving Google’s perspective API built for detecting toxic comments. arXiv preprint [arXiv:1702.08138](https://arxiv.org/abs/1702.08138) (2017)
10. Iyyer, M., Wieting, J., Gimpel, K., Zettlemoyer, L.: Adversarial example generation with syntactically controlled paraphrase networks. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), vol. 1, pp. 1875–1885 (2018)
11. Jin, D., Jin, Z., Zhou, J.T., Szolovits, P.: Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 8018–8025 (2020)
12. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746–1751 (2014)
13. Li, D., et al.: Contextualized perturbation for textual adversarial attack. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 5053–5069 (2021)
14. Li, L., Ma, R., Guo, Q., Xue, X., Qiu, X.: BERT-ATTACK: adversarial attack against BERT using BERT. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 6193–6202 (2020)
15. Mrksic, N., et al.: Counter-fitting word vectors to linguistic constraints. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 142–148 (2016)
16. Pang, B., Lee, L.: Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 115–124 (2005)
17. Pruthi, D., Dhingra, B., Lipton, Z.C.: Combating adversarial misspellings with robust word recognition. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 5582–5591 (2019)
18. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8), 9 (2019)
19. Ren, S., Deng, Y., He, K., Che, W.: Generating natural language adversarial examples through probability weighted word saliency. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1085–1097 (2019)
20. Ribeiro, M.T., Singh, S., Guestrin, C.: Semantically equivalent adversarial rules for debugging NLP models. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, pp. 856–865 (2018)
21. Schakel, A.M.J., Wilson, B.J.: Measuring word significance using distributed representations of words. arXiv preprint [arXiv:1508.02297](https://arxiv.org/abs/1508.02297) (2015)
22. 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 (Volume 1: Long Papers), vol. 1, pp. 1715–1725 (2016)
23. Szegedy, C., et al.: Intriguing properties of neural networks. In: ICLR 2014 : International Conference on Learning Representations (ICLR) 2014 (2014)
24. Wilson, B.J., Schakel, A.M.J.: Controlled experiments for word embeddings. arXiv preprint [arXiv:1510.02675](https://arxiv.org/abs/1510.02675) (2015)

25. Zang, Y., et al.: Word-level textual adversarial attacking as combinatorial optimization. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6066–6080 (2020)
26. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: NIPS 2015 Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, vol. 28, pp. 649–657 (2015)



RAST: A Reward Augmented Model for Fine-Grained Sentiment Transfer

Xiaoxuan Hu^{1,4}, Hengtong Zhang^{2,5}, Wayne Xin Zhao^{3,4(✉)}, Yaliang Li⁶,
Jing Gao², and Ji-Rong Wen^{1,3,4}

¹ School of Information, Renmin University of China, Beijing, China

² School of Electrical and Computer Engineering, Purdue University,
West Lafayette, USA

³ Gaoling School of Artificial Intelligence, Renmin University of China,
Beijing, China

⁴ Beijing Key Laboratory of Big Data Management and Analysis Methods,
Beijing, China

⁵ Department of Computer Science and Engineering, University at Buffalo,
Buffalo, USA

⁶ Alibaba Group, Hangzhou, China

Abstract. In this paper, we propose a novel model *RAST* (**R**eward **A**ugmented **S**entiment **T**ransfer) for fine-grained sentiment transfer. Existing methods usually suffer from two major drawbacks, *i.e.*, blurred sentiment distinction and unsatisfactory content preservation. Considering the above issues, we design two kinds of rewards to better control *sentiment* and *content*. Specially, we develop a pairwise comparative discriminator that enforces to generate sentences with clear distinctions for different sentiment intensities. Moreover, we utilize an effective sampling strategy to obtain pseudo-parallel sentences with minor changes on the input sentence to enhance content preservation. Experiments on a benchmark dataset show that the proposed model outperforms several competitive approaches.

Keywords: Fine-grained sentiment transfer · Reward augmented training

1 Introduction

Sentiment transfer [11, 19, 20] refers to the task of editing a sentence to alter sentiment as desired while meanwhile preserving the essential content in the input sentence. Such a task can potentially contribute to a large variety of downstream tasks, such as emotional conversation generation [22].

Conventionally, previous sentiment transfer methods mainly focus on the setting of binary sentiment labels, only transferring coarse-grained sentiments (*e.g.*,

X. Hu and H. Zhang—Equal contribution.

positive and negative). Recently, several studies [9, 12] have extended the sentiment label set by considering fine-grained sentiment intensities (*e.g.*, five levels). Since there are no parallel data in the sentiment transfer task, reinforcement learning algorithms [12, 13, 20] have been utilized to train the sentiment transfer model. However, in fine-grained sentiment transfer, the search space of the policy is large, and it is difficult to effectively control both *sentiment* and *content* in the generated texts.

Therefore, existing fine-grained sentiment transfer models usually suffer from two major drawbacks: (1) blurred sentiment intensity distinction, and (2) unsatisfactory content preservation. Here, we provide an example to show two imperfect outputs as motivating cases in Table 1. In this example, five target sentiment intensity scores ranging from 0.1 (most negative) to 0.9 (most positive) are given. The first imperfect output suffers from semantic drift, as “*food*” is changed to “*service*”. The second imperfect output shows excessive negativity for the target sentiment intensity of 0.3. Thus it is hard to distinguish it from other sentences with the most negative sentiment score of 0.1, since they both express the sentiment of “*horrible*”.

Table 1. An example of the input, reference outputs and imperfect outputs of the fine-grained sentiment transfer task. Target sentiment is abbreviated as “TS”.

Input	
horrible food , i would not go there again .	
TS	Reference output
0.1	the food tasted awful , i would not go there again .
0.3	the food was not tasty , maybe i would not go there again .
0.5	plain and normal food , maybe i would go there again .
0.7	fresh and tasty food , i would go there again .
0.9	the food was extremely delicious, i would go there again .
TS	Imperfect output
0.3	slow service , maybe i would not go there again .
0.3	the food tasted horrible , i would not recommend it at all .

To tackle these two drawbacks, we propose a novel model, named *RAST* (**R**eward **A**ugmented **S**entiment **T**ransfer), based on the reinforcement learning framework for the fine-grained sentiment transfer task. The major highlights of the proposed model are that we utilize reward augmented training and pairwise sentiment critics to enhance *content preservation* and alleviate *blurred sentiment distinction* for the fine-grained sentiment transfer task, respectively. Such a training strategy is particularly suitable to our task since it obtains rewards from augmented samples that are slightly perturbed based on real samples, instead of samples produced by the text generator. Such a nice property naturally enhances content preservation for the sentiment transfer task. Besides, we also introduce

pairwise sentiment critics to alleviate the problem of *blurred sentiment distinction*. Compared with existing regression-based critics [12], the pairwise sentiment critics enforce more clear sentiment distinctions among sentences of different levels. To our knowledge, it is the first time that reward augmented training has been applied to fine-grained sentiment transfer with specially designed content and sentiment rewards.

2 Methodology

2.1 Overview

Let us consider a labeled sentence dataset $\mathcal{D} = \{(x^1, s^1), \dots, (x^n, s^n)\}$ containing n sentences, where each sentence x^i is paired with a target sentiment intensity score s^i from a fine-grained sentiment intensity set $\mathcal{S} = \{s_1, \dots, s_k\}$. Given a sentence x and a sentiment intensity score s , our task is to generate a sentence y_g that preserves the essential content of x and meanwhile reflects the sentiment intensity suggested by the target sentiment s .

Figure 1 presents an overview of the proposed *RAST (Reward Augmented Sentiment Transfer)* model, which is built on the idea of reward augmented training [14]. In RAST, an encoder-decoder based generator is utilized to produce the new sentence. The generator will be updated or optimized according to rewards or feedbacks received from the training scheme.

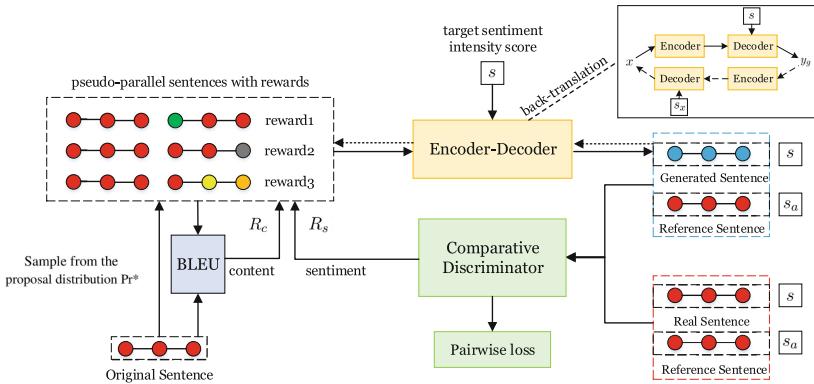


Fig. 1. An overview of the proposed RAST. The detailed back-translation process is illustrated in the top right.

2.2 Encoder-Decoder Based Sentiment Transfer Model

To encode the content of the original sentences, we employ a bi-directional LSTM encoder. Each token of the source sentence x , is firstly represented by a trainable

semantic representation and then encoded into a sequence of hidden vectors $\{\overleftarrow{\mathbf{h}}_i\}_{i=1}^m$.

With the hidden representations of source sentence x and target sentiment intensity score s , the decoder generates a sentence y_g that preserves the content of x and conforms to the sentiment of s . Here, we build the decoder upon LSTM network and use an additional sentiment embedding to map each word to its sentiment representation. We define the hidden state \mathbf{h}_{y_t} of the decoder at time t as:

$$\mathbf{h}_{y_t} = f(\mathbf{h}_{y_{t-1}}, [\mathbf{e}_{y_{t-1}}^c; \mathbf{e}_{y_{t-1}}^s], \mathbf{c}_{y_t}), \quad (1)$$

where $\mathbf{e}_{y_{t-1}}^c$ is the content representation of the word y_{t-1} , $\mathbf{e}_{y_{t-1}}^s$ is the sentiment representation, and the \mathbf{c}_{y_t} is a context vector obtained by the attention mechanism.

Finally, similar to previous work [12, 21], we model the generation of each token in the transferred sentence by considering both sentiment and content information. Formally, we model the probability of generating each token combining content- and sentiment-based probabilities as:

$$\text{Pr}_{w_t=w} = \gamma \text{Pr}_{w_t=w}^c + (1 - \gamma) \text{Pr}_{w_t=w}^s, \quad (2)$$

where γ is a trade-off parameter, $\text{Pr}_{w_t=w}^c$ and $\text{Pr}_{w_t=w}^s$ are content- and sentiment-based generation probabilities, respectively. The content-based generation probability is as follows:

$$\text{Pr}_{w_t=w}^c = \text{softmax}(\mathbf{W}_c \mathbf{h}_{y_t})_{[w]}, \quad (3)$$

where \mathbf{W}_c is a trainable parameter matrix. Furthermore, the sentiment-based generation probability is defined as:

$$\text{Pr}_{w_t=w}^s = \text{softmax}(\mathbf{o}_t^s), \quad (4)$$

$$\mathbf{o}_t^s = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(g(\mathbf{E}_s, \mathbf{h}_{y_t}) - s)^2}{2\sigma^2}\right), \quad (5)$$

$$g(\mathbf{E}_s, \mathbf{h}_{y_t}) = \text{sigmoid}(\mathbf{E}_s \mathbf{W}_s \mathbf{h}_{y_t}), \quad (6)$$

where \mathbf{E}_s is a matrix that stores sentiment representations of all the words, s is the target sentiment, \mathbf{W}_s is a trainable parameter matrix and σ is the standard deviation. Here, $g(\mathbf{E}_s, s_t)$ plays the role of sentiment prediction for the current token, and we restrict it to be around the target sentiment using a Gaussian kernel layer (Eq. 5). Intuitively, the word that is more consistent with the target sentiment intensity score will be assigned a higher sentiment-based generation probability.

2.3 Comparative Discriminator

In this part, we propose to use a pairwise comparative discriminator, which compares the sentiment intensity levels of two samples, to enhance fine-grained sentiment control.

Comparative Sentiment Discrimination. Let x_1 and x_2 denote two sentences with different sentiment intensity levels. Suppose x_1 has a more positive sentiment compared with x_2 , we can form a partial comparison for the pair $\langle x_1, x_2 \rangle$.

Specially, we only focus on the pairs of *adjacent sentiment intensity*, since they are generally more difficult to discriminate. In this way, we can construct comparative pairs using two sentences of adjacent sentiment levels using training data. With these comparative samples, the discriminator learns to predict whether one of the sentences has a higher (or lower) sentiment intensity than the other. Formally, the probability of predicting the $\langle x_1, x_2 \rangle$ as label q is denoted by $\Pr_\phi(q|x_1, x_2)$, where $q \in \{\text{higher, lower}\}$ and ϕ denotes the parameters of the discriminator D_ϕ . We adopt a BERT [2] based pairwise classifier as the comparative discriminator. Pretrained representations from BERT are taken for initialization and fine-tuned according to our classification task. Specifically, we feed both sentences of a pair into the BERT model and use the final hidden layer \mathbf{C} as the aggregate representation of the sentence pair. Then we calculate $\Pr_\phi(q|x_1, x_2)$ as:

$$\Pr_\phi(q|x_1, x_2) = \frac{\mathbf{W}_q \mathbf{C}^\top}{\sum_{q'} \mathbf{W}_{q'} \mathbf{C}^\top}, \quad (7)$$

where \mathbf{W}_q is a trainable weight matrix for label q . Based on $\Pr_\phi(q|x_1, x_2)$, we define the concept of *sentiment reward* as:

$$R_s(x) = \Pr_\phi(q|x, x'), \quad x' \sim \Pr_{data}(x), \quad (8)$$

where $x' \sim \Pr_{data}(x)$ denotes that x' is a sentence sampled from the sentences with adjacent intensity of x in training data. This reward is used in the rest of this paper for the sake of conciseness.

Discriminator Training. We take three kinds of sentences as input in the discriminator, namely *original*, *reference* and *generated* sentences. Here, reference sentences are real sentences with adjacent sentiment intensity levels of the original sentence from training data. First, our discriminator favors sentiment intensity distinction between the real sentence and reference sentences. Then, it further compares sentiment intensities between generated sentences and reference sentences. If we could accurately transfer the sentiment, sentiment comparison results between generated and reference sentences should be same as those between real and reference sentences. We will punish the case that a generated sentence mismatches its target sentiment intensity compared with reference sentences of adjacent sentiment levels.

To train the discriminator, we propose a loss function that maximizes the sentiment reward of real samples and minimizes that of generated samples compared with referenced real samples:

$$\mathcal{L}_{D_\phi} = -(\mathbb{E}_{x \sim \Pr_{data}}[R_s(x)] - \mathbb{E}_{y_g \sim \Pr_{G_\theta}}[R_s(y_g)]), \quad (9)$$

where x and y_g denote the real sentence and the generated sentence, respectively, and G_θ is the sentiment transfer model introduced in Sect. 3.4.

2.4 Reward Augmented Training of Sentiment Transfer Model

Since there are no parallel data in the fine-grained sentiment transfer task, we can not directly rely on the maximum likelihood estimation (MLE) for training. To tackle this problem, existing works [12, 13, 20] incorporate reinforcement learning algorithms and utilize task-specific rewards to guide the training of the sentiment transfer models. However, in fine-grained sentiment transfer, the search space of the policy is large and the content-related rewards cannot provide sufficient precise feedback on how well the content information is preserved. In this section, we propose a reward augmented training scheme to address these difficulties.

Reward Augmented Training. The reward augmented training strategy is inspired by the reward augmented maximum likelihood (RAML) approach [14], which is originally proposed to incorporate task-specific rewards into the MLE training. Compared with vanilla reinforcement learning, the rewards are derived using the samples generated from the exponentiated pay-off distribution instead of the policy (*i.e.*, fine-grained sentiment transfer models in this paper). Here, the exponentiated payoff distribution is defined as:

$$Q(y; \tau) = \frac{\exp(R(y)/\tau)}{\sum_{y' \in \mathcal{Y}} \exp(R(y')/\tau)}, \quad (10)$$

where $R(y)$ is a reward function that measures the quality of a sentence y (to be specified later), \mathcal{Y} is the set of all possible generation results, and τ is the temperature hyper-parameter. Since the denominator exhausts all the possible generation results, it is usually quite challenging to directly sample from Q . In this case, we can resort to using another tractable proposal distribution $\Pr^*(y)$ by importance sampling.

Such a proposal distribution enables us to restrict the sample space that we use to get rewards. Particularly in our task, we can pay more attention to samples with minor changes on original sentences. In this way, it is encouraged to seek content changes as small as possible to accomplish the sentiment transfer goal.

Design of the Proposal Distribution. Specifically, we construct the proposal distribution $\Pr^*(y)$ as follows. Inspired by a simple template-based baseline [8], we firstly construct pseudo-parallel data $\langle x, \tilde{y} \rangle$ generated by replacing the sentiment words from the source sentence with those of semantically similar target sentence. For each \tilde{y} , we further use a stratified sampling approach following RAML [14] to obtain new samples. The sampling proceeds in three steps:

$$\Pr^*(y|\tilde{y}) = P(d, p, w|\tilde{y}) = P(d|\tilde{y}) P(p|\tilde{y}, d) P(w|\tilde{y}, d, p). \quad (11)$$

The first step is to sample an edit distance d . Let $c(e, m)$ denotes the number of sentences at an edit distance e from a sentence \tilde{y} of length m . It can be calculated approximately as follows: $c(e, m) = \binom{m}{e} \cdot (|\mathcal{V}| - 1)^e$, where $|\mathcal{V}|$ denotes the size of vocabulary. Following [14], we reweight the counts by $\exp(-e/\tau)$ and

normalize them, thus we can sample an edit distance \tilde{d} from

$$P(d = \tilde{d}|\tilde{y}) = \frac{\exp(-\tilde{d}/\tau)c(\tilde{d}, m)}{\sum_{e=0}^m \exp(-e/\tau)c(e, m)}. \quad (12)$$

The next step is to randomly select \tilde{d} positions $\{p_1, p_2, \dots, p_{\tilde{d}}\}$ in the sequence to be modified. The probability of selecting the position \tilde{p} is computed as $\Pr(p = \tilde{p}|\tilde{y}, d = \tilde{d}) = \frac{\tilde{d}}{m}$.

At the final step, we also randomly sample \tilde{d} substitutions uniformly from the vocabulary \mathcal{V} and the probability is computed as follows:

$$\Pr(w|\tilde{y}, d = \tilde{d}, p = p_1, p_2, \dots, p_{\tilde{d}}) = \prod_{j=1}^{\tilde{d}} P(w_j|\tilde{y}_{j-1}, p = p_j). \quad (13)$$

Here, we use the random sampling strategy instead of some well-designed distribution of language models. This is because such random samplings make the augmented samples more diverse, and both positive and negative samples can benefit the training process.

Defining the Reward. We design the reward function $R(y)$ considering both the goals of task and the stationary proposal distribution. Specifically, the reward function can be computed as follows:

$$R(y) = \tau \cdot [\log \Pr^*(y) + (1 + \beta^2) \frac{R_c(y) \cdot R_s(y)}{(\beta^2 \cdot R_c(y)) + R_s(y)}], \quad (14)$$

where β is a harmonic weight that controls the trade-off between sentiment reward $R_s(y)$ and content reward $R_c(y)$. To implement the function, the sentiment reward $R_s(y)$ is provided by the output of comparative discriminator D_ϕ (Eq. 8), and the content reward $R_c(y)$ is calculated as the BLEU-2 score [15] between original and current sentences. By combining the two rewards, our model advocates to generate sentences with high content preservation confirming to the target sentiment. Another note is that Eq. 14 has involved the term of the proposal probability $\Pr^*(y)$. Revisiting Eq. 11, we can see that $\Pr^*(\cdot)$ prefers to sample sentences with minor changes, which further enhances the content preservation.

Learning Objective. Finally, in order to optimize the RAML objective, we use importance sampling to sample from $\Pr^*(y)$ instead of Q . The importance weight can be further calculated as follows:

$$\text{weight}(y) \propto \frac{Q(y)}{\Pr^*(y)} \propto \frac{\exp(R(y)/\tau)}{\Pr^*(y)} \propto \exp\left(\frac{R_c(y) \cdot R_s(y)}{(\beta^2 \cdot R_c(y)) + R_s(y)}\right). \quad (15)$$

Thus the RAML objective can be re-expressed as:

$$\mathcal{L}_{RAML,\theta} = -\mathbb{E}_{y \sim \Pr^*(y)} [\text{weight}(y) \log \Pr_{G_\theta}(y|x)]. \quad (16)$$

The exploration space exposed for model training is mainly the regions surrounding the real source data, thus contributing to higher content preservation and more stable reward signals.

Algorithm 1 The learning algorithm for RAST.

Input:

A dataset $\mathcal{D} = (x^i, s^i)$ including input sequence x^i with a target fine-grained sentiment intensity score s^i ; pseudo-parallel data $\mathcal{D}_0 = (x^i, y^i)$;

- 1: Pre-train the encoder-decoder model G_θ using MLE loss on \mathcal{D}_0 ;
- 2: Generate samples from Pr_{G_θ} ;
- 3: Pre-train the comparative discriminator D_ϕ via Eq. 9;
- 4: Construct a stationary distribution $\text{Pr}^*(y)$ via Eq 11-Eq 13;
- 5: **for** each iteration $t = 1, 2, \dots, T$ **do**
- 6: **for** each iteration $n = 1, 2, \dots, N$ **do**
- 7: Update G_θ using RAML loss via Eq. 16;
- 8: Update G_θ using back-translation loss via Eq. 17;
- 9: **end for**
- 10: **for** each iteration $m = 1, 2, \dots, M$ **do**
- 11: Update D_ϕ using pairwise loss via Eq. 9;
- 12: **end for**
- 13: **end for**

Back-Translation. Besides, we note that back-translation technique [1, 7, 11, 12] in unsupervised machine translation, which is potentially useful in our task. The idea is to map the source sequence x to target sequence $G_\theta(x, s)$ and then map it back to produce an identical source sequence. The back-translation loss is defined as:

$$\mathcal{L}_{bt} = -\log \text{Pr}_{G_\theta}(x|G_\theta(x, s), s_x), \quad (17)$$

where s_x is the source sentiment intensity score. We integrate back-translation as a complementary technique to further improve content compatibility. We present the learning algorithm of RAST in Algorithm 1.

3 Experiments

3.1 Experiment Settings

Datasets. We conduct experiments on the widely used *Yelp dataset* [12]. The dataset consists of product reviews aligned with sentiment ratings from 1 to 5. In this section, we normalize these ratings to the range of [0, 1] and use them as the sentiment intensity scores. After data cleaning, we randomly select 240K samples for training, 30K for validation and 500 for testing. Such a pre-process procedure is identical to [12].

Implementation Details. We implement our framework using Tensorflow. We set both content and sentiment embeddings to be 300-dimensional and train these embeddings from scratch. The encoder is a 1-layer bidirectional LSTM, and the decoder is a 1-layer single directional LSTM. The hidden sizes of both

the encoder and the decoder are set to 256. For the discriminator, we use a pre-trained BERT-based model with all the internal states (*i.e.*, hidden states, word embeddings and positional embeddings) set to 768-dimensional. The hyper-parameter σ in the decoder is set to 0.01. We use Adam optimizer for model training and set the batch size to 32. The learning rates for the pre-train and the adversarial training process are set to 10^{-3} and 10^{-5} , respectively. We also employ dropout with the rate 0.5 to avoid overfitting. The temperature hyper-parameter τ in Eq. 10 is 0.8, γ in Eq. 2 is 0.5 and harmonic weight β in Eq. 14 is 1.

Comparison Methods. We consider the following methods for comparison:

- *Rv-VAE* [9]: It employs a revised VAE to disentangle latent content factor and outcome factor from a sentence and then edit the sentence to change the outcome.
- *Rv-VAE+extra* [9]: It incorporates a coupling component modeling pseudo-parallel sentence pairs with three extra loss into *Rv-VAE*.
- *SC-Seq2Seq* [21]: It adopts an attention-based Seq2Seq model with an extra specificity variable to generate specificity-controlled response, trained under the cycle reinforcement learning algorithm following [12].
- *Seq2SentiSeq* [12]: It incorporates sentiment intensity values into the attention-based Seq2Seq model and adopts a cycle reinforcement learning algorithm to guide model training.

3.2 Evaluation Metrics

Automatic Evaluation. In this paper, we consider two series of evaluation metrics, *i.e.*, content-related metrics and sentiment-related metrics.

For *content-related metrics*, we calculate the *BLEU* score [15] between the outputs and human references provided by [12] for content preservation.

For *sentiment-related metrics*, we consider both absolute and relative gap between sentiment intensity of outputs and target sentiment intensity. We include Mean Absolute Error (MAE) (*i.e.*, deviation between predicted sentiment intensity scores and ground truth) and Mean Relative Reciprocal Rank (MRRR) (*i.e.*, the relative intensity ranking of outputs) as evaluation metrics. The smaller the MAE or the larger the MRRR is, the better a model performs. Finally, to evaluate the *quality* of the generated sentences, we calculate their perplexity (PPL) [1, 7, 10] by a pre-trained Kneser-Ney smooth trigram language model [6] using KenLM [4]. The smaller the PPL is, the better a model performs.

Human Evaluation. We conduct human evaluation to further verify the performance of different methods. We randomly select 20 sentences from the test set and let all the methods generate sentences of five sentiment intensity levels for each input sentence. Thus we get 100 generated sentences for each method, and then distribute them to three annotators. Annotators are required to score the

generated sentences and human references from 1 to 5 in terms of three criteria: the preservation of the original content, the accuracy of the target sentiment and the fluency.

3.3 Results and Analysis

Automatic Evaluation Results. The performance comparison of different methods on automatic evaluation are shown in Table 2. As we can see, the proposed model RAST achieves the best performance in almost all cases. Regarding content preservation, RAST achieves absolute improvement of 3.0/0.1 points on content-related metrics like BLEU-1 / BLEU-2, compared with the state-of-the-art Seq2SentiSeq [12]. These results indicate that the proposed reward augmented training strategy can effectively enforce content preservation in sentiment transfer.

Table 2. Results of different methods for automatic evaluation.

Model	Automatic evaluation				
	BLEU-1↑	BLEU-2↑	MAE↓	MRRR↑	PPL↓
Rv-VAE	22.6	7.2	0.22	0.61	2746.6
Rv-VAE+extra	20.7	5.7	0.20	0.63	1042.1
SC-Seq2Seq	23.9	3.8	0.24	0.60	20.9
Seq2SentiSeq	32.5	10.3	0.23	0.63	21.1
Ours	35.5	10.4	0.22	0.66	18.8

Moreover, we observe that in terms of sentiment control, RAST performs better on the relative sentiment metric MRRR than the absolute sentiment metric MAE. For fine-grained sentiment control, RAST achieves 0.01/0.03 improvement on sentiment-related metrics MAE/MRRR. Such improvements are more significant than that of the state-of-the-art model Seq2SentiSeq. This is because the pairwise comparative loss in our model tends to enforce a clear sentiment distinction between sentences with different target sentiment scores. Finally, we also evaluate the fluency of the generated sentences, i.e., PPL and Fluency. Again, the proposed method outperforms all the baselines. These results mean that the proposed method successfully alleviates the major drawbacks of the existing methods, i.e., blurred sentiment distinction and unsatisfactory content preservation, without sacrificing the linguistic quality of the generated sentences.

Human Evaluation Results. Here, we report the performance comparison of different methods on human evaluation in Table 3. The results indicate that the proposed method significantly outperforms the state-of-the-art baseline Seq2SentiSeq in terms of content preservation and sentiment and merely suffer a less than 8% disadvantage in terms of fluency. We find that there is a trade-off

between content preservation/sentiment and fluency in the sentiment transfer task. Given higher content and sentiment requirements as constraints, the sentences generated by the sentiment transfer method may suffer slight fluency degrade.

Table 3. Results of different methods for human evaluation.

Model	Human evaluation		
	Content↑	Sentiment↑	Fluency↑
Rv-VAE	2.09	2.42	1.88
Rv-VAE+extra	1.93	2.61	1.86
SC-Seq2Seq	2.30	3.15	3.74
Seq2SentiSeq	2.81	3.10	3.97
Ours	2.97	3.59	3.70
Human Reference	4.51	4.70	4.60

3.4 Ablation Study

Furthermore, we study the impact of important extensions in our model. Here, we retrain the model by removing the back-translation component, the comparative discriminator or the entire reward augmented training scheme (short by *RAT*).

Table 4. Ablation study with automatic evaluation.

Model	Automatic evaluation				
	BLEU-1↑	BLEU-2↑	MAE↓	MRRR↑	PPL↓
Full model	35.5	10.4	0.22	0.66	18.8
w/o back-translation	28.1	5.8	0.19	0.69	25.1
w/o discriminator	34.8	10.8	0.24	0.62	19.6
w/o RAT	31.8	8.8	0.23	0.64	20.4

Table 4 presents the results of the ablation study. As one can see, when the learning strategy is removed, the performance decreases by 11% / 15% for content-related metrics BLEU-1/BLEU-2, while the relative sentiment metric MRRR decreases by 0.02. This phenomenon indicates that the reward augmented training strategy mainly contributes to content preservation. Moreover, the results of relative sentiment metric MRRR also decreases when the comparative discriminator is removed. It indicates that the comparative discriminator can help improve the sentiment transfer performance.

3.5 Case Study

In Table 5, we qualitatively compare the results of our model and the state-of-the-art method Seq2SentiSeq [12] on five samples of different sentiment intensities from Yelp dataset. As we can see, our model has captured subtle sentiment difference across the five sentiment levels, such as “*a complete waste of money*” → “*a little pricey*” → “*a little slow*” → “*great*” → “*love*”, while Seq2SentiSeq fails to accurately reflect fined-grained sentiment difference especially for adjacent sentiment levels.

Table 5. Comparison between RAST and the baseline Seq2SentiSeq with five sentiment levels.

Input	food is always amazing no matter what i order
Output	RAST (Our model)
s=0.1	food is always amazing but it 's just a complete waste of money
s=0.3	food is always amazing , but it 's just a little pricey
s=0.5	food is always amazing , but the service is a little slow
s=0.7	food is always amazing , and the service is always great
s=0.9	food is always amazing , and i love the food
Output	Seq2SentiSeq
s=0.1	food is always good , and i am never disappointed
s=0.3	food is always good , and i am never disappointed
s=0.5	food is always good and they are always very good
s=0.7	food is always good and they are always very good
s=0.9	food is always good and they are always very good

4 Related Work

Recently, text style transfer with non-parallel data has drawn much attention from the research community. There are a large majority of existing works [1, 3, 5, 7, 8, 11, 16, 17, 19] focus on coarse-grained sentiment transfer, in which sentiment labels are binary. However, to the best of our knowledge, there are merely a few existing works focus on the fine-grained control of sentiment. For instance, Liao et al. [9] employ a revised VAE to learn disentangled representations and construct pseudo-parallel sentence pairs to train the model in a supervised setting. Luo et al. [12] adopt a sentiment-controlled Seq2Seq model and introduces a cycle reinforcement learning algorithm to provide rewards to guide the model training. Several works [10, 18] revise the text in a continuous representation space by iteratively editing the latent representation with gradient until conforming to the target sentiment.

Compared with existing fine-grained sentiment transfer methods, our model alleviates two major drawbacks of the existing works, *i.e.*, blurred sentiment intensity distinction and unsatisfactory content preservation. The proposed reward training strategy restricts the sentiment transfer policy to search around the input sentences for the ideal transfer results with minor changes. This strategy largely contributes to content preservation and training stability compared to vanilla reinforcement learning-based approaches in [12]. Moreover, this paper also proposes a pairwise comparative discriminator, enforcing larger intensity gaps between different sentiment levels compared with regression/multi-class discriminators in existing works [12, 18].

5 Conclusion

In this paper, we proposed a novel model RAST for the fine-grained sentiment transfer task. In order to alleviate the unsatisfactory content preservation and blurred sentiment distinctions, we developed a reward augmented training scheme and incorporated a pairwise comparative discriminator into it. Experiments have shown that our model outperforms several state-of-the-art baselines in terms of content preservation and sentiment modification in both automatic and human evaluation.

Acknowledgments. The work was supported in part by the National Science Foundation of China under Grant No. 61872369, Beijing Academy of Artificial Intelligence (BAAI), and the National Science Foundation of the United States of America under Grant No. IIS-1747614.

References

1. Dai, N., Liang, J., Qiu, X., Huang, X.: Style transformer: unpaired text style transfer without disentangled latent representation. In: Proceedings of ACL, pp. 5997–6007 (2019)
2. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL, pp. 4171–4186 (2019)
3. Fu, Z., Tan, X., Peng, N., Zhao, D., Yan, R.: Style transfer in text: exploration and evaluation. In: Proceedings of AAAI, pp. 663–670 (2018)
4. Heafield, K.: KenLM: faster and smaller language model queries. In: Proceedings of the Sixth Workshop on Statistical Machine Translation, WMT@EMNLP 2011, Edinburgh, Scotland, UK, 30–31 July 2011, pp. 187–197 (2011)
5. John, V., Mou, L., Bahuleyan, H., Vechtomova, O.: Disentangled representation learning for non-parallel text style transfer. In: Proceedings of ACL, pp. 424–434 (2019)
6. Kneser, R., Ney, H.: Improved backing-off for m-gram language modeling. In: 1995 International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1995, Detroit, Michigan, USA, 08–12 May 1995, pp. 181–184 (1995)
7. Lample, G., Subramanian, S., Smith, E.M., Denoyer, L., Ranzato, M., Boureau, Y.: Multiple-attribute text rewriting. In: Proceedings of ICLR (2019)

8. Li, J., Jia, R., He, H., Liang, P.: Delete, retrieve, generate: a simple approach to sentiment and style transfer. In: Proceedings of NAACL, pp. 1865–1874 (2018)
9. Liao, Y., Bing, L., Li, P., Shi, S., Lam, W., Zhang, T.: QuaSE: sequence editing under quantifiable guidance. In: Proceedings of EMNLP, pp. 3855–3864 (2018)
10. Liu, D., Fu, J., Zhang, Y., Pal, C., Lv, J.: Revision in continuous space: fine-grained control of text style transfer. CoRR (2019)
11. Logeswaran, L., Lee, H., Bengio, S.: Content preserving text generation with attribute controls. In: Advances in NIPS, pp. 5108–5118 (2018)
12. Luo, F., et al.: Towards fine-grained text sentiment transfer. In: Proceedings of ACL, pp. 2013–2022 (2019)
13. Luo, F., et al.: A dual reinforcement learning framework for unsupervised text style transfer. In: Proceedings of IJCAI, pp. 5116–5122 (2019)
14. Norouzi, M., et al.: Reward augmented maximum likelihood for neural structured prediction. In: Advances in NIPS, pp. 1723–1731 (2016)
15. Papineni, K., Roukos, S., Ward, T., Zhu, W.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of ACL, pp. 311–318 (2002)
16. Shen, T., Lei, T., Barzilay, R., Jaakkola, T.S.: Style transfer from non-parallel text by cross-alignment. In: Advances in NIPS, pp. 6830–6841 (2017)
17. Sudhakar, A., Upadhyay, B., Maheswaran, A.: “Transforming” delete, retrieve, generate approach for controlled text style transfer. In: Proceedings of EMNLP, pp. 3267–3277 (2019)
18. Wang, K., Hua, H., Wan, X.: Controllable unsupervised text attribute transfer via editing entangled latent representation. In: Advances in NIPS, pp. 11034–11044 (2019)
19. Wu, X., Zhang, T., Zang, L., Han, J., Hu, S.: Mask and infill: applying masked language model to sentiment transfer. In: Proceedings of IJCAI, pp. 5271–5277 (2019)
20. Xu, J., et al.: Unpaired sentiment-to-sentiment translation: a cycled reinforcement learning approach. In: Proceedings of ACL, pp. 979–988 (2018)
21. Zhang, R., Guo, J., Fan, Y., Lan, Y., Xu, J., Cheng, X.: Learning to control the specificity in neural response generation. In: Proceedings of ACL, pp. 1108–1117 (2018)
22. Zhou, H., Huang, M., Zhang, T., Zhu, X., Liu, B.: Emotional chatting machine: emotional conversation generation with internal and external memory. In: Proceedings of AAAI, pp. 730–739 (2018)



Pre-trained Language Models for Tagalog with Multi-source Data

Shengyi Jiang^{1,2}, Yingwen Fu¹, Xiaotian Lin¹, and Nankai Lin^{1,2(✉)}

¹ School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou, China

² Guangzhou Key Laboratory of Multilingual Intelligent Processing, Guangdong University of Foreign Studies, Guangzhou, China

Abstract. Pre-trained language models (PLMs) for Tagalog can be categorized into two kinds: monolingual models and multilingual models. However, existing monolingual models are only trained in small-scale Wikipedia corpus and multilingual models fail to deal with Tagalog-specific knowledge needed for various downstream tasks. We train three existing models on a much larger corpus: *BERT-uncased-base*, *ELECTRA-uncased-base* and *RoBERTa-base*. At the pre-training stage, we construct a large-scale news text corpus for pre-training in addition to the existing open-source corpora. Experimental results show that our pre-trained models achieve consistently competitive results in various Tagalog-specific natural language processing (NLP) tasks including part-of-speech (POS) tagging, hate speech classification, dengue classification and natural language inference (NLI). Among them, POS tagging dataset is a self-constructed dataset aiming to alleviate the insufficient labeled resource for Tagalog. We will release all pre-trained models and datasets to the community, hoping to facilitate the future development of Tagalog NLP applications.

Keywords: Pre-trained language model · Tagalog · POS tagging

1 Introduction

Pre-trained language models (PLMs) represented by BERT [1] have been proven to significantly improve the performance of various downstream natural language processing (NLP) tasks and thus become extremely popular for many NLP researches. Despite of success of pre-trained BERT and its variants, they have largely limited to high-resource languages such as English. For a new language, one could pre-train a new language-specific model based on BERT architecture and training method [2–5] or utilize existing pre-trained multilingual BERT-based models [1, 6, 7].

In terms of PLMs for Tagalog, monolingual models [8–10] and multilingual models [7, 11] are both publicly available. However, there are two main concerns about these two kinds of models:

- (1) **Monolingual models:** All the existing monolingual models for Tagalog are only pre-trained on the Tagalog Wikipedia corpus [8]. While Wikipedia data is not representative of a general language use, and the Tagalog Wikipedia data size is relatively

small (283M in size uncompressed), pre-trained language models can be significantly improved by using more pre-training data [12] from different data sources such as news.

- (2) **Multilingual models:** Multilingual pre-trained models struggle to explain their applicability in acquiring language-invariant knowledge for downstream tasks of various languages. As different languages have different sequence structures, multilingual pre-trained models are more suitable for cross-language applications than in monolingual applications. As an agglutinative language, Tagalog shows some characteristics of inflectional languages. It also has a variety of lexical morphology, complex syntactic structure and relatively free sequence order. It is necessary to pre-train monolingual models for Tagalog to improve the performance of downstream tasks.

To tackle the two issues above, we train three monolingual BERT-based models using 1444M Tagalog corpus (four times more than Wikidata used in previous works) from multiple data sources. At the pre-training stage, we construct a large-scale news text corpus for pre-training in addition to the existing open-source corpora. We evaluate our models on three benchmark Tagalog text classification datasets: Hate Speech classification, Dengue classification and natural language inference (NLI) [9, 10]. In addition to text classification, pre-trained models should be evaluated in more kinds of NLP tasks such as sequence labeling tasks. However, the recent sequence-labeled resources in Tagalog are scarce that they cannot meet the development of deep learning technology in terms of scale and quality. Therefore, we construct a Tagalog part-of-speech (POS) tagging (referred as a common sequence labeling task) dataset consisting of 14438 sentences. Experimental results show that our models obtain competitive results on all these tasks.

The contributions in this paper are summarized as follows:

- (1) We present a series of large-scale monolingual language models pre-trained for Tagalog on a much larger size of corpus.
- (2) We construct a large-size news corpus for Tagalog language, which could make up for the gap in scarce Tagalog NLP resources.
- (3) We construct a large-scale and high-quality Tagalog POS tagging dataset to alleviate the current situation of insufficient language resources.
- (4) Our models achieve competitive performances on four downstream datasets, showing the effectiveness of BERT-based monolingual language models for Tagalog.
- (5) The pre-trained models and the POS tagging dataset would be publicly available serving as strong baselines.

2 Related Previous Research

2.1 Natural Language Processing for Tagalog

Part-of-Speech Tagging. Part-of-speech (POS) is a fundamental grammatical attribute of tokens that signifies the morphological and syntactic behaviors of a lexical item. It is

designed as one of the sequence labeling tasks. Cheng and Rabo [13] construct a POS tagging corpus comprised of 141 sentences and 59 tags and propose a template-based n-gram POS tagger. Reyes et al. [14] develop a Tagalog POS tagger (SVPOST) using support vector machines (SVMs) and their corpus consists of 122318 tokens and 64 tags. Olivo et al. [15] are the first to use conditional random field (CRF) for Tagalog POS tagging. There are two main concerns about the POS tagging research in Tagalog: (1) Tagalog is represented as a low-resource language that most of the Tagalog POS taggers are still based on rules and machine learning (ML). (2) The corpora above are not publicly available, which makes it impossible for us to properly compare performance of different models and techniques. In this work, we build and release a high-quality POS corpus and use neural methods to construct baseline POS tagger.

Text Classification. Cruz et al. [10] create and release News PH-NLI, the first Natural Language Inference (NLI) benchmark dataset in Tagalog. Moreover, they produce new pre-trained transformers to further alleviate the resource scarcity in Tagalog. Cruz and Cheng [9] release two text classification datasets, namely Hate Speech Dataset (binary classification) and Dengue Dataset (multilabel text classification). They also pre-train transformer-based language models for use within Tagalog setting. Our pre-trained models are evaluated in these three benchmark datasets for comparison.

2.2 Pre-trained Language Model for Tagalog

Monolingual Pre-trained Language Model. Cruz and Cheng [8] pre-train a new Tagalog BERT model using the WikiTextTL-39 dataset. In order to cater to low-resource settings in an equipment perspective, they also construct a smaller version of the BERT model via model distillation, producing a DistilBERT model. Cruz et al. produce four ELECTRA models: a cased and an uncased model respectively in the base size and small size, using the WikiText TL-39 dataset [10].

Multilingual Pre-trained Language Model. Publicly transformer-based multilingual PLMs represented by multilingual BERT (mBERT) [1], XLM [9] and mt5 [11] are trained in a large dataset including multiple language datasets to obtain language-invariant information. It is notable that XLM-100 and mt5 support Tagalog language while mBERT does not support Tagalog language.

3 Model

Three model for Tagalog are introduced in this paper: an uncased BERT model¹, an uncased ELECTRA² model and a RoBERTa³ model. They are all in the base size (12 layers, 768 hidden units, 12 attention heads).

¹ <https://huggingface.co/GKLMIP/bert-tagalog-base-uncased>.

² <https://huggingface.co/GKLMIP/electra-tagalog-base-uncased>.

³ <https://huggingface.co/GKLMIP/roberta-tagalog-base>.

3.1 BERT

BERT (Bidirectional Encoder Representations for Transformers) [1], is designed to learn deep bidirectional representations from unlabeled text by jointly modeling context from both forward and backward directions in all layers. It consists of multiple bidirectional transformer encoders [17].

BERT is comprised of two unsupervised subtasks, namely Mask Language Model (MLM) and Next Sentence Prediction (NSP): (1) MLM refers to masking some words from the input sequence and then predicting the masked word through the context; (2) NSP is designed to enhance the relationship between a sentence pair. Its objective is to predict whether the sentence pair are continuous. Pre-trained BERT can be fine-tuned for a variety of downstream tasks such as text classification, named entity recognition (NER) and question answering (QA) tasks (Fig. 1).

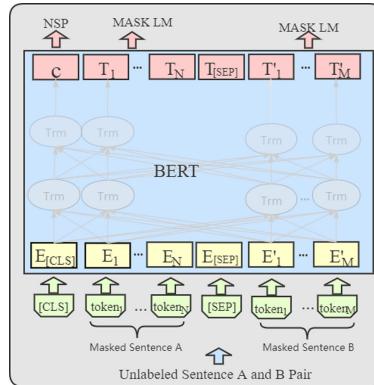


Fig. 1. BERT model.

3.2 RoBERTa

Being a variant of BERT, RoBERTa [12] aims to make full use of BERT architecture and training methods. There are three improvements in RoBERTa compared with BERT: (1) **More training data:** RoBERTa leverages more unlabeled data to pre-train the model for a more robust performance in downstream tasks; (2) **Abundance of NSP task:** Liu et al. [12] verified the invalidity of the NSP task and removed this task; (3) **Dynamic word masking:** RoBERTa uses dynamic word masking to train the MLM task instead of the static word masking proposed by BERT model, which allows the parameters of the pre-trained model to be more fully optimized and the model can better capture sequence features (Fig. 2).

3.3 ELECTRA

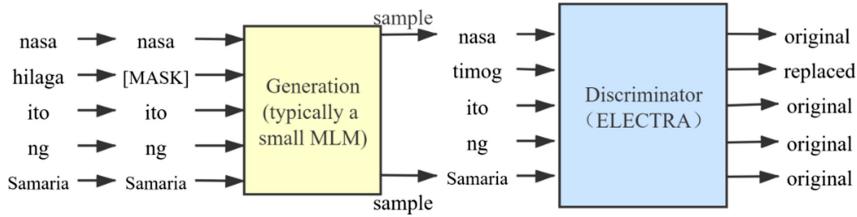


Fig. 2. ELECTRA model.

Apart from BERT model, a new pre-trained framework, ELECTRA [16], uses the combination of generator and discriminator.

Compared with BERT, the innovations of ELECTRA are as follows: (1) It proposes replaced token detection (RTD), a pre-training task in which the model learns to distinguish real input tokens from plausible but synthetically generated replacements. Instead of masking, ELECTRA corrupts the input by replacing some tokens with samples from a proposal distribution, which is typically the output of a small masked language model. (2) Instead of training a model that predicts the original identities of the corrupted tokens, ELECTRA trains a discriminative model that predicts whether each token in the corrupted input was replaced by a generator sample or not. (3) In order to effectively learn context information, it uses weight sharing to share the generator's embedding information with the discriminator. (4) The model jointly trains a small generator and a discriminator to ease the training difficulty of the discriminator.

4 Pre-training Corpus

In this paper, the models are pre-trained on massive data collected from three sources, namely Oscar corpus, Wikipedia corpus and news corpus. The whole corpus used for pre-training has a size of 1.3G, which is four times more than the corpus used by the existing Tagalog pre-trained models. We use 99% of the corpus as the training set and 1% as the validation set. The corpus statistics for pre-training are shown in Table 1.

Table 1. Statistics of the pre-training corpus.

Source	Size of File	Num. of Document	Num. of Line	Num. of Tokens
Oscar	417M	–	3580299	78236499
Wiki	283M	174444	2004429	5173224
News	744M	396941	6341530	140186315

4.1 Oscar

Oscar corpus is a large-scale unlabeled corpus constructed by Ortiz et al. [18]. In order to create the multilingual OSCAR corpus, Ortiz et al. (2019) reproduce the pipeline proposed by Grave et al. [19] to process, filter and classify Common Crawl, which is a non-profit organization that produces and maintains an open, freely available repository of crawled data from the web. The filtering step used to create OSCAR involves keeping only the lines containing at least 100 UTF-8 encoded characters. Finally, as in Grave et al. [19], the OSCAR corpus is deduplicated, i.e. for each language, only one occurrence of a given line is included. In this paper, we only use Tagalog corpus from the OSCAR corpus.

4.2 Wiki

Wikipedia is a multilingual, free encyclopedia that contains a lot of text information. We use the Tagalog Wikipedia corpus “WikiText-TL-39” [8] as one of the training corpora. “TL” stands for Tagalog and “39” refers to the dataset having 39 million tokens in the training set. In this paper, we use the training set, validation set and test set of this corpus for pre-training.

4.3 News

We crawl massive news articles from 13 Tagalog news websites to construct a large-scale news corpus for Tagalog. The corpus is comprised of around 400,000 news articles as shown in Table 2.

Table 2. Statistics of news websites.

Website	Num. of Document
https://www.pna.gov.ph	85655
http://balita.net.ph/	37051
http://bandera.inquirer.net	73525
http://cnnphilippines.com	96
http://eaglenews.com	9802
https://www.bworldonline.com	7728
https://tonite.abante.com.ph	28045
https://www.topgear.com.ph	355
https://philnews.ph	181
https://kickerdaily.com	11153
https://www.hatawtabloid.com	39486
https://www.remate.ph	93908
https://www.pinoyparazzi.com	9956
Total	396941

5 Experiment

5.1 Downstream Tasks

POS Tagging. The existing Tagalog POS tagging datasets cannot meet the development of deep learning technology in terms of scale and quality and all of them are not publicly available. Therefore, we build a dataset containing 14438 samples (totally 286706 words) within 39 tags based on the Tagalog news articles crawled from Bailta⁴. In the annotating process, each sample is labeled by two annotators. Then samples with the same labeling results are added to the dataset. Instead, samples with different annotation results will be annotated again by the third Annotators. If the annotation results are the same as one of the first two persons, They will also be added to the dataset. A split of (70%, 15%, 15%) of the dataset is respectively for (training, test, validation). Statistics of the POS tagging dataset and POS Tagset are represented in Table 3 and Table 4.

Table 3. Data distribution of the POS tagging dataset.

Data	Num. of Sentence	Num. of Token
Train	10108	195468
Dev	2165	46971
Test	2165	44267
Total	14438	286706

Table 4. Statistics of the POS tagging dataset.

Tag	Proportion (%)	Explanation
CN	13.7018	Common noun
AD	2.2776	Auxiliary verb
P	4.9263	Particle
CP	3.6675	Completed
PREP	13.4961	Preposition
A	3.1935	Adjective
ART	5.5548	Article
PN	6.0414	Proper noun
Z	10.8941	Punctuation
INF	3.1307	Infinitive

(continued)

⁴ <http://balita.net.ph/>.

Table 4. (*continued*)

Tag	Proportion (%)	Explanation
CS	4.2158	Connection structure
INTP	0.1531	Interrogative pronoun
PP	4.8457	Personal pronoun
CC	1.986	Coordinating conjunction
SC	2.5556	Subordinating conjunction
NP	0.3007	Negative pronoun
F	10.013	Foreign Word
INTADV	0.1779	Indefinite adverb
NADV	0.8273	Negative adverb
CT	1.3251	Contemplated
DP	1.1943	Demonstrative pronoun
INC	1.8667	Incompleted
JOD	0.6446	Ordinal number of adjective
CD	1.1224	Cardinal number
X	1.1451	Unknown
INDP	0.1221	Indefinite pronoun
INT	0.0743	Interjection
AS	0.1102	Adjective, superlative degree
DADV	0.3101	Demonstrative adverb
VOD	0.0345	Ordinal number of adverb
INDADV	0.0244	Indefinite adverb
DD	0.0593	Demonstrative determiner
HADV	0.0014	Adverb of the same class
NUM	0.0007	Numeral
ADJ	0.0028	Adjective
ADV	0.0003	Adverb
SADV	0.001	Adverb, superlative degree
QD	0.0007	Quantitative determiner
V	0.001	Verb

Natural Language Inference. Natural Language Inference (NLI) is a sentence-pair classification for inference of the relationship between two sentences, such as a sentence with a premise and a sentence with a hypothesis. Their relationship can be entailment, neutrality and contradiction. NewsPH-NLI [10] is an NLI benchmark dataset in Tagalog comprised of multiple news articles from all major Tagalog news sites online. The dataset is divided into (420000, 90000, 9000) documents for (training, test, validation) sets.

Hate Speech Classification. Hate Speech dataset [9] is a collection of tweets mined in real-time during the 2016 Philippine Presidential Election debates, and from tweets related to the 2016 election hashtags. The dataset is introduced as a binary classification task benchmark in Tagalog, with each tweet labeled as 0 (non-hate) or 1 (hate). The training set has 10,000 labeled examples with 5340 and 4660 non-hate and hate tweets respectively. An even split of 4232 validation and 4232 test samples are included for evaluation.

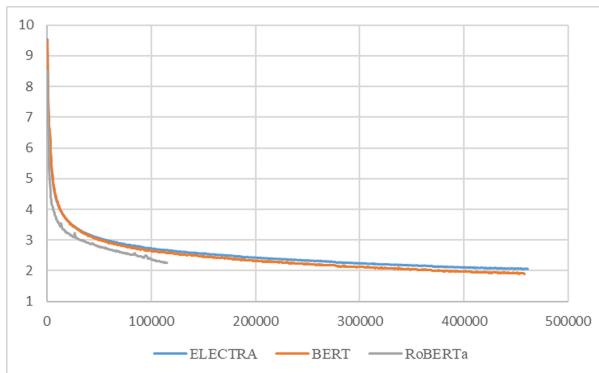
Dengue Classification. Dengue dataset [9], a multiclass classification dataset, is composed of tweets collected from Twitter in the Tagalog language. There are five labels for each tweet in the dataset: absent, dengue, health, mosquito, and sick. The dataset is represented as a low-data dataset, with only 4015 training examples and an even split of 500 validation and 500 test examples. More importantly, the classes are highly imbalanced with a distribution of (905, 49, 1804, 528, 1035) samples in five labels (absent, dengue, health, mosquito, sick).

5.2 Pre-training

Two improvements are made in our pre-trained models compared with the existing pre-trained models [8]: (1) **more data**; and (2) **a larger vocabulary size**. The vocabulary in the models pre-trained by Cruz and Cheng [8] is 32K, while the size of our pre-trained dictionary is 52K. In addition, in the pre-training stage in [8], the model pre-training epoch exceeds 50 batches, while we only conduct 5 batches. The BERT and ELECTRTRA models we build are both uncased models, because in general, the performance of uncased models is better than the cased models [8]. When training BERT and ELECTRTRA, we use the word piece [20] segmentation method, and when training RoBERTa, we use the BPE [21] segmentation method. The hyperparameters in the pre-training stage are shown in Table 5.

Table 5. Hyperparameters for pre-training.

Parameter	BERT	ELECTRA	RoBERTa
Layer Num	12	12	12
Hidden Size	768	768	768
FFN inner hidden size	3072	3072	3072
Attention heads	12	12	12
Vocab Size	52000	52000	52000
Tokenizer Type	Word Piece	Word Piece	BPE
Adam β_1	0.9	0.9	0.9
Adam β_2	0.999	0.999	0.98
Adam ϵ	1e-6	1e-6	1e-6
Learning Rate Decay	Linear	Linear	Linear
Weight Decay	0.01	0.01	0.01
Batch Size	128	128	128
Peak Learning Rate	1e-4	1e-4	5e-4
Dropout	0.1	0.1	0.1
Attention Dropout	0.1	0.1	0.1
Epoch	5	5	5
Warmup Steps	5K	5K	5K
Max Length	512	512	512

**Fig. 3.** Pre-training losses for all models over the steps.

5.3 Fine-Tuning

We compare our pre-trained models with six existing pre-trained models. For classification tasks, we report the accuracy of the test set as in [9, 10], and report the accuracy,

precision, recall and F1-score for POS tagging task. We use the same classification fine-tune code⁵ from Cruz et al. [8–10]. In the classification task, we uniformly set the maximum length to 128, while in the POS tagging task, the maximum length is set to 200. For two small ELECTRA models, we fine-tune for 3 epochs with a learning rate of 2e-4. For three base BERT model and base ELECTRA model, we fine-tune for 3 epochs with a learning rate of 5e-5. For RoBERTa model, we fine-tune for 5 epochs with a learning rate of 3e-5 for NewsPH-NLI task because it needs a longer training time and a smaller learning rate to converge. And for other task, we fine-tune RoBERTa for 3 epochs with a learning rate of 5e-5. For XLM model, we fine-tune for 5 epochs with a learning rate of 1e-6 for hate speech classification task and NIL task, and fine-tune for 3 epochs with a learning rate of 5e-5 for POS tagging task and dengue classification task. The GPU we use for model fine-tuning is TIAN RTX.

5.4 Experiment Results and Analysis

As shown in Fig. 3, in the warm-up phase, the loss values of BERT, ELECTRA and RoBERTa drop significantly, while in the subsequent phases, the loss values slowly decrease. Among them, the loss value of ELECTRA dropped from 9.5212 to 2.0620, the loss value of BERT dropped from 9.4844 to 1.9041, and the loss value of RoBERTa dropped from 8.5030 to 2.2657. It seems that BERT model fits the best.

Table 6. The result of NewsPH-NLI.

Model	Test Loss	Test Accuracy
BERT (base, cased)	0.3169	0.8870
BERT (base, uncased)	0.3114	0.8884
ELECTRA (base, cased)	0.2572	0.9113
ELECTRA (base, uncased)	0.2528	0.9168
ELECTRA (small, cased)	0.1896	0.9279
ELECTRA (small, uncased)	0.1948	0.9249
XLM	0.2116	0.9115
BERT (base, uncased, our)	0.1872	0.9466
ELECTRA (base, uncased, our)	0.1858	0.9489
RoBERTa (base, uncased, our)	0.3678	0.9128

⁵ <https://github.com/jcblaisecruz02/Filipino-Text-Benchmarks>.

Table 7. The result of hate speech classification.

Model	Test Loss	Test Accuracy
BERT (base, cased)	0.6172	0.7695
BERT (base, uncased)	0.5849	0.7862
ELECTRA (base, cased)	0.6264	0.7648
ELECTRA (base, uncased)	0.5925	0.7608
ELECTRA (small, cased)	0.4725	0.7883
ELECTRA (small, uncased)	0.5009	0.7683
XLM	0.5508	0.7110
BERT (base, uncased, our)	0.5578	0.8193
ELECTRA (base, uncased, our)	0.5588	0.8264
RoBERTa (base, uncased, our)	0.5497	0.7930

Table 8. The result of dengue classification.

Model	Test Loss	Test Accuracy
BERT (base, cased)	0.1886	0.9318
BERT (base, uncased)	0.1708	0.9405
ELECTRA (base, cased)	0.1953	0.9288
ELECTRA (base, uncased)	0.1750	0.9330
ELECTRA (small, cased)	0.1833	0.9316
ELECTRA (small, uncased)	0.1754	0.9296
XLM	0.2014	0.9133
BERT (base, uncased, our)	0.1395	0.9541
ELECTRA (base, uncased, our)	0.1454	0.9525
RoBERTa (base, uncased, our)	0.1572	0.9425

Table 6–9 present the results of different pre-trained models for 4 downstream tasks. Our models outperform the existing models in three text classification tasks: (1) Our pre-trained ELECTRA model works best in NLI task and hate speech classification task which achieve an accuracy of 0.9489 and 0.8264; (2) for dengue classification task, our pre-trained BERT model reaches state-of-the-art performance with an accuracy of 0.9541. It is worthwhile to note that in POS tagging task, our pre-trained BERT model and ELECTRA model have the same F1-score, and BERT reaches state-of-the-art performance with an accuracy of 0.9532.

Table 9. The result of POS tagging.

Model	Accuracy	Precision	Recall	F1
BERT (base, cased)	0.9441	0.9259	0.9222	0.9241
BERT (base, uncased)	0.9429	0.9264	0.9222	0.9243
ELECTRA (base, cased)	0.9358	0.9149	0.9134	0.9142
ELECTRA (base, uncased)	0.9368	0.9164	0.9140	0.9152
ELECTRA (small, cased)	0.9412	0.9231	0.9190	0.9211
ELECTRA (small, uncased)	0.9387	0.9200	0.9162	0.9181
XLM	0.9517	0.9352	0.9328	0.9340
BERT (base, uncased, our)	0.9532	0.9381	0.9351	0.9366
ELECTRA (base, uncased, our)	0.9531	0.9379	0.9353	0.9366
RoBERTa (base, uncased, our)	0.9473	0.9301	0.9271	0.9286

6 Conclusion

In this paper, we present three monolingual language models for Tagalog pre-trained in a much larger corpus. Additionally, we construct a part-of-speech (POS) tagging dataset to relieve the insufficient sequence-labeled resources in Tagalog. Experimental results demonstrate the effectiveness of our pre-trained models in various Tagalog natural language processing (NLP) tasks of POS tagging, hate speech classification, dengue classification and natural language inference (NLI). By publicly releasing the pre-trained models, we hope that they can have implications for future research for Tagalog NLP.

Acknowledgement. This work was supported by the National Natural Science Foundation of China (No. 61572145), the Major Projects of Guangdong Education Department for Foundation Research and Applied Research (No. 2017KZDXM031) and National Social Science Foundation of China (No. 17CTQ045). The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

References

- Devlin, J., Chang, M.W., Lee K., Toutanova K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACLHLT 2019, pp. 4171–4186 (2019)
- Cui, Y., Che, W., Liu, T., Qin, B., Yang, Z., Wang, S., Hu, G.: Pre-training with whole word masking for Chinese BERT. CORR (2019)
- de Vries, W., van Cranenburgh, A., Bisazza, A., Caselli, T., van Noord, G., Nissim, M.: BERTje: a Dutch BERT model. CORR (2019)
- Vu, X.S., Vu, T., Tran, S.N., Jiang, L.: ETNLP: a visual-aided systematic approach to select pre-trained embeddings for a downstream task. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing, pp. 1285–1294 (2019)

5. Martin, L., et al.: CamemBERT: a tasty French language model. In: Annual Meeting of the Association for Computational Linguistics, pp. 7203–7219 (2020)
6. Lample, G., Conneau A.: Cross-lingual language model pretraining. CORR (2019)
7. Conneau, A., et al.: Unsupervised cross-lingual representation learning at scale. In: Annual Meeting of the Association for Computational Linguistics, pp. 8440–8451 (2020)
8. Cruz, J.C.B., Cheng, C.: Evaluating language model finetuning techniques for low-resource languages. CORR (2019)
9. Cruz, J.C.B., Cheng, C.: Establishing baselines for text classification in low-resource languages. CORR (2020)
10. Cruz, J.C.B., Resabal, J.K., Lin, J., Velasco, D. J., Cheng, C.: Investigating the true performance of transformers in low-resource languages: A case study in automatic corpus creation. CORR (2020)
11. Xue, L., et al.: mT5: a massively multilingual pre-trained text-to-text transformer. CORR (2021)
12. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. CORR (2019)
13. Cheng, C., Rabo, S.: TPOST: a template-based, n-gram part-of-speech tagger for Tagalog. *J. Res. Sci. Comput. Eng.* **3**(1) (2004)
14. Reyes, C.D.E., Suba, K.R.S., Razon, A.R., Naval Jr., P.C.: SVPOSTA part-of-speech tagger for Tagalog using support vector machines. In: Proceedings of the 11th Philippine Computing Science Congress (2011)
15. Olivo, J.F.T., Hari, P.J.T., dela Fuente, M.B.: CRFPOST: part-of-speech tagger for Filipino texts using conditional random fields. In: Proceedings of the 2nd International Conference on Algorithms, Computing and Artificial Intelligence, pp. 444–449 (2019)
16. Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: ELECTRA: pre-training text encoders as discriminators rather than generator. In: Proceedings of International Conference on Learning Representations (2020)
17. Vaswani, A., et al. Attention is all you need. CORR (2017)
18. Suárez, P.O., Romary, L., Sagot, B.: A monolingual approach to contextualized word embeddings for mid-resource languages. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (2020)
19. Grave, E., Bojanowski, P., Gupta, P., Joulin, A., ikolov, T.: Learning word vectors for 157 languages. In: Proceedings of the 11th Language Resources and Evaluation Conference, European Language Resource Association (2018)
20. Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., et al.: Google’s neural machine translation system: bridging the gap between human and machine translation. CORR (2016)
21. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. CORR (2015)



Accelerating Pretrained Language Model Inference Using Weighted Ensemble Self-distillation

Jun Kong, Jin Wang^(✉), and Xuejie Zhang

School of Information Sciences and Engineering, Yunnan University, Kunming, China
wangjin@ynu.edu.cn

Abstract. Pretrained language models (PLMs) have achieved remarkable results in various natural language processing tasks. As the performance of the model increases, it is also accompanied by more computational consumption and longer inference time, which makes deploying PLMs in edge devices for low-latency applications challenging. To address this issue, recent studies have recommended applying either model compression or early-exiting techniques to accelerate the inference. However, model compression permanently discards the modules of the model, leading to a decline in model performance. Train the PLMs backbone and the early-exiting classifier separately with early-exiting strategies. It not only brings extra training cost but also loses semantic information from higher layers, resulting in unreliable decisions of early-exiting classifiers. In this study, a weighted ensemble self-distillation method was proposed to improve the early-exiting strategy, which well balanced the performance and the inference time. It enables early-exiting classifiers to obtain rich semantic information from different layers with an attention mechanism according to the contribution of each layer to the final prediction. Furthermore, it simultaneously performs weighted ensemble self-distillation and fine-tuning of the PLMs backbone so that the PLMs can be fine-tuned in the training process of the early-exiting classifier to preserve the performance as much as possible. The experimental results show that the inference of the proposed model was accelerated at the minimum cost of performance loss, thus outperforming the previous early-exiting models. The code is available at: <https://github.com/JunKong5/WestBERT>.

Keywords: Pretrained language model · Accelerating inference · Weighted ensemble self-distillation · Early-exiting

1 Introduction

Practical applications of natural language processing (NLP) have completely been revolutionized with the advent of PLMs. Models such as BERT [2], RoBERTa [9] and ALBERT [8] have been widely used for sentiment analysis [7, 18, 19], text summarization [4], and subject labeling of academic papers [16].

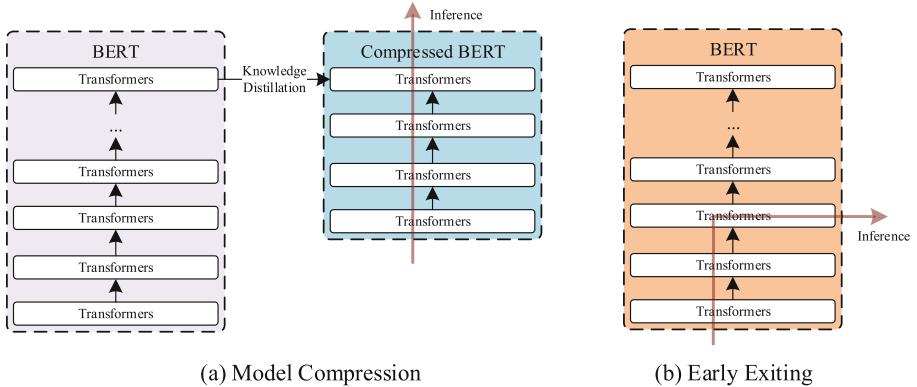


Fig. 1. Conceptual diagram of both model compression and early-exiting strategies for accelerating inference of the PLMs.

As the performance of PLMs continues to increase, the requirements for computational resources and time consumption become urgent. Several variants of PLMs significantly increase the latency and computational cost of inference due to the large-scale parameters used in PLMs. Practically, the speed of inference is as important as the prediction performance of the model. It may severely limit their deployment to resource-limited devices for real-time applications. For instance, the execution speed of an application directly impacts battery life and user experience for a smartphone.

Recent studies have recommended using model compression techniques to deploy PLMs, including knowledge distillation [13], pruning [1], and quantization [17]. Knowledge distillation is usually implemented to transfer knowledge from the original PLMs to a smaller one, as shown in Fig. 1(a). For example, DistilBERT [10] and PKD-BERT [12] can distill the 12-layer teacher model of BERT into a 6-layer student in either pre-training or finetuning stages, and the new model can have sufficient capacity to learn a concise knowledge representation. Alternately, the technique of pruning can be used to remove unnecessary parts of the network. Based on this, Fan et al. [3] proposed layer drop strategies to prune the layers of the network by a dropout mechanism to reduce the redundancy in a trained network. Finally, quantization [17] truncates floating point numbers to only use a few bits, thus speeding up the computation of the values. These compression methods permanently discard the modules of the model and may reduce the inference performance of the model.

Another option for accelerating inference is the early-exiting strategy, as shown in Fig. 1(b). For instance, DeeBERT [15], BrachyNet [14] and Right-Tool [11] introduced extra classifiers in each layer of the BERT model. Instead of using the representation from the last layer of the BERT, the models evaluate the confidence by those classifiers to dynamically decide which layer can be used as off-ramps for prediction. At the inference stage, after a sample goes through a transformer layer, it is passed to the following off-ramp. If the off-ramp is

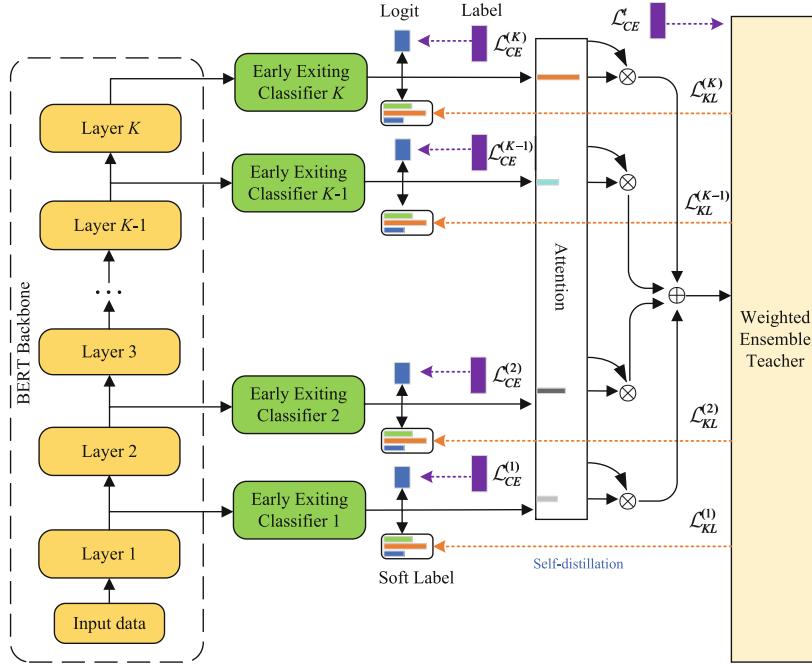


Fig. 2. Overall architecture of adaptive inference with weighted ensemble self-distillation for BERT.

confident enough for the prediction, the result is returned; otherwise, the sample is sent to the next transformer layer.

These early-exiting methods were trained in two separate stages. That is, the backbone network of BERT is trained first and then frozen to train the early-exiting classifiers, which brings extra costs for training time and computational resources. Furthermore, the separate training process freezes the BERT backbone network, resulting in weak knowledge transfer between multiple early-exiting classifiers. In addition, a previous study [6] argued that surface features are expressed in lower layers, syntactic features are expressed more in middle layers, and semantic features are expressed in higher layers for PLMs. This brings the contradiction that, the earlier the model exits, the less semantic features needed for the task are used. As a result, the lack of higher-layer semantic features may lead to significant performance degradation of the early-exiting classifier.

In this study, a **Weighted Ensemble Self-disTillation** method was proposed to accelerate the inference time of the BERT model (WestBERT). To ensure that the lower layers can obtain the powerful representation abilities as the higher layers, a weighted ensemble teacher was dynamically composed by using an attention mechanism according to the contribution of each layer to the final prediction. Then, it was used to distill all layers of the original BERT to ensure that the lower layers can obtain the appropriate abilities for inference. Even if

the model exits very early, accurate predictions can still be obtained because the proposed WestBERT extracts rich knowledge from inside the network. Such weighted ensemble self-distillation can transfer knowledge from different parts of the model, which contributes the most to the final inference to the lower layers of the model. In addition, it does not require the external training cost of the teacher and the student models. Another advantage of the proposed WestBERT is that it simultaneously performs weighted ensemble self-distillation and fine-tuning of the backbone BERT. Therefore, the backbone BERT model can be fine-tuned in the process of self-distillation to preserve the performance as much as possible.

Extensive experiments were conducted on the GLUE dataset. The results show that the proposed WestBERT well balanced the model performance and inference time. The weighted ensemble self-distillation strategy significantly reduces the inference time with little performance loss. In contrast to model compression, the proposed WestBERT not only suppresses unnecessary computation of simple samples but also provides a dynamic architecture with an early-exiting strategy for different samples.

The rest of the paper is organized as follow. Section 2 presents a detailed description of the proposed weighted ensemble self-distillation to accelerate the inference time of the BERT model. Section 3 summarizes the implementation details and experimental results. The conclusions of this study are finally drawn in Sect. 4.

2 Weighted Ensemble Self-distillation

The proposed WestBERT performs weighted ensemble self-distillation and fine-tuning of the backbone BERT simultaneously. For self-distillation, each early-exiting classifier was regarded as a student model. An attention mechanism was applied to measure the importance of different students and compose them into an appropriate teacher. Figure 2 shows an overview of the proposed WestBERT model. The details of each module are presented as follows.

2.1 Early Exiting

The BERT model usually contains K layers of transformer structure. The input sample was first tokenized as a sequence of subwords, i.e., $\mathbf{x} = [x_1, x_2, \dots, x_N]$. Its corresponding ground truth label is y . A special token [CLS] was first added to the head of the sequence so that the corresponding hidden state $h_{[\text{CLS}]}^{(k)} \in \mathbb{R}^{d_h}$ in each layer is encoded including all representative information of all tokens through the multilayer encoding procedure. For each layer, the encoding process is defined as

$$[h_{[\text{CLS}]}^{(k)}, h_1^{(k)}, \dots, h_N^{(k)}] = f_\theta^{(k)}([h_{[\text{CLS}]}^{(k-1)}, h_1^{(k-1)}, \dots, h_N^{(k-1)}]; \theta) \quad (1)$$

where $f_\theta^{(k)}$ is the transformer encoder in the k -th layer parameterized by θ . Then, $h_{[\text{CLS}]}^{(k)}$ was regarded as features to train the early-exiting classifier $z^{(k)}$. Thus,

the predicted probability distribution $\hat{y}^{(k)} \in \mathbb{R}^m$ of the early-exiting classifier towards the ground-truth label is calculated as follows:

$$z^{(k)} = \tanh(W_z^{(k)} h_{[\text{CLS}]}^{(k)} + b_z^{(k)}) \quad (2)$$

$$\hat{y}_z^{(k)} = \text{softmax}(z^{(k)}) \quad (3)$$

where m is the number of classes, \tanh is the activation function. $W_z^{(k)}$ and $b_z^{(k)}$ are weights and bias, respectively. The loss function for the target task is a categorical cross-entropy, defined as

$$\mathcal{L}_{CE}^{(k)} = - \sum_{k=1}^K \mathbb{I}(y) \circ \log(\hat{y}_z^{(k)}) \quad (4)$$

where y and $\hat{y}_z^{(k)}$ represent the ground truth and the probability distribution in the k -th layer. $\mathbb{I}(y)$ denotes a one-hot vector with the y -th component being one, and \circ represents an elementwise multiplication operation.

2.2 Weighted Ensemble Self-distillation

The supervised cross-entropy loss based on labels alone leads to weak expressiveness of early-exiting classifiers due to the lack of higher-level semantic information in each hidden representation $h_{[\text{CLS}]}^{(k)}$. Thus, weighted ensemble self-distillation was applied to further improve the representation abilities and obtain rich semantic information of the early-exiting classifiers. Each early-exiting classifier $z^{(k)}$ was regarded as a student model. An attention mechanism was used to compose all students as the teacher model according to their importance to the final prediction. In knowledge distillation [5], student models can learn from the distribution of the teacher model to improve their generalization ability. The distribution $p_s^{(k)} \in \mathbb{R}^{d_p}$ of each student is

$$p_s^{(k)} = \text{softmax}\left(\frac{W_s z^{(k)} + b_s}{\tau}\right) \quad (5)$$

where τ is the temperature, which is used to control the softness of the distribution.

To compose the teacher model, the attention weights $\alpha^{(k)}$ are calculated by a softmax function over all layers, defined as

$$\alpha^{(k)} = \frac{\exp(V^T \tanh(W_\alpha z^{(k)} + b_\alpha))}{\sum_{k=1}^K \exp(V^T \tanh(W_\alpha z^{(k)} + b_\alpha))} \quad (6)$$

where V is the context vector and W_α and b_α are weights and bias, respectively. The feature vector t of the teacher model is then composed of

$$t = \sum_{k=1}^K \alpha^{(k)} z^{(k)} \quad (7)$$

To further ensure the quality of the teacher model, supervised training was applied with respect to the ground-truth label y , defined as

$$p_t = \text{softmax}\left(\frac{W_t t + b_t}{\tau}\right) \quad (8)$$

$$\mathcal{L}_{CE}^t = -\mathbb{I}(y) \circ \log(p_t) \quad (9)$$

where $p_t \in \mathbb{R}^{d_p}$ is the probability distribution of the composed teacher. Self-distillation was achieved by minimizing the Kullback-Leibler divergence (KL) between the distributions of the teacher model and all of the student models, which is defined as

$$\mathcal{L}_{KL}^{(k)} = \tau^2 KL(p_s^{(k)} || p_t) \quad (10)$$

where $KL(\bullet || \bullet)$ is the KL-divergence function. τ^2 compensates for the size of the gradient scaled by the soft target, ensuring that there is no negative impact on the gradient size. $p_s^{(k)}$ and p_t are the soft probability distributions of the k -th student classifier and the teacher model, respectively. Based on this, the overall loss of the proposed WestBERT is expressed as

$$\mathcal{L} = \sum_T \left[(1 - \lambda) \sum_{k=1}^K \mathcal{L}_{CE}^{(k)} + \mathcal{L}_{CE}^t + \lambda \sum_{k=1}^K \mathcal{L}_{KL}^{(k)} \right] \quad (11)$$

where λ is a decay factor that is used to balance the cross-entropy loss and the knowledge distillation loss.

2.3 Adaptive Inference

The entropy $Ent^{(k)}$ was quantified as the confidence based on the output distribution of the early-exiting classifier in the k -th layer, which is defined as

$$Ent^{(k)} = - \sum_{d_p} p_s^{(k)} \log p_s^{(k)} = \ln\left(\sum_{d_p} \exp(p_s^{(k)})\right) - \frac{\sum_{d_p} p_s^{(k)} \exp(p_s^{(k)})}{\sum_{d_p} \exp(p_s^{(k)})} \quad (12)$$

where d_p is the dimension of hidden representation of the early-exiting classifier. Notably, the larger the entropy value is, the greater the uncertainty of the off-ramp is. When a sample arrives at the early-exiting classifier, the entropy of its output distribution $p_s^{(k)}$ in each layer is successively compared with a preset confidence threshold F to determine whether the model should exit early or continue to the next layer of inference.

Based on this, the prediction can be made by exiting from the off-ramp if the entropy value is less than the threshold value, i.e., $Ent^{(k)} < F$. Thus, different adaptive inference paths are customized for each sample such that it is unnecessary for all layers of BERT to participate in inference, thus reducing the inference time. Otherwise, the model continues with the next layer of inference when it is greater than the threshold value, i.e., $Ent^{(k)} > F$. Intuitively, a larger F leads to faster but less accurate prediction, while a small F leads to accurate but slower prediction. The selection of the threshold depends on whether the task benefits from inference speed or predictive performance.

3 Experiments

3.1 Datasets and Evaluation Metrics

To evaluate the effectiveness of the proposed weighted ensemble self-distillation to accelerate the BERT inference, we conducted experiments on the GLUE dataset. MNLI contains both matched and mismatched versions of the dataset. The average values of F_1 and accuracy were used as evaluation metrics for QQP and MRPC. Accuracy is used as an evaluation metric for SST-2, MNLI, QNLI and RTE. The Matthews correlation coefficient is used as the evaluation metric of CoLA.

3.2 Baselines

To comprehensively evaluate the proposed WestBERT model, we compared it with model compression and other adaptive inference methods. The details of baselines are presented as follows: **BERT-base** [2]. The original 12-level uncased BERT is used as the baseline, and its inference time is used as the benchmark and is noted as 100%. **DistilBERT** [10]. Distillation is performed during the pretraining phase. **BERT-PKD** [12]. Knowledge is extracted from the middle layer of the teacher model. **LayerDrop** [3]. Structural pruning of layers is performed using dropout for BERT. **DeeBERT** [15]. Early-exiting classifiers are added to speed up inference. We use the available official code to implement them.

3.3 Implementation Details

We use BERT-base-uncased as the backbone network architecture of the model. BERT contains 12 transformer layers, and we add a total of 12 early-exiting classifiers behind the corresponding layers. Each early-exiting classifier consists of a linear layer with a hidden state dimension of 768, and a dropout layer of dropout rate is set to 0.1. When an input sample reaches the early-exiting classifier and is less than the confidence threshold, it exits early to reduce inference time. To train the model, Adam was used to optimize the training objective. The learning rate is 2e-5, and the epsilon is 1e-8. The batch size is set to 128, the maximum length of the input sample is set to 128, and the number of epochs is 10. We set λ to 0.1 and τ to 3.0.

Table 1. Experimental results comparison of baseline methods on the development set splits of the GLUE.

Methods	SST-2		CoLA		MNLI (m)		QQP	
	Acc%	Time%	Mcc%	Time%	Acc%	Time%	A/F1%	Time%
BERT-base	92.1	100	54.3	100	83.9	100	89.9	100
DistilBERT	90.7	50.0	43.6	50.0	79.0	50.0	84.9	50.0
BERT-PKD	91.3	50.0	45.5	50.0	81.3	50.0	88.4	50.0
LayerDrop	90.7	50.0	45.4	50.0	80.7	50.0	88.3	50.0
DeeBERT	90.8	53.1	52.5	73.1	80.0	65.2	86.0	56.2
WestBERT	91.5	35.1	53.1	70.1	81.9	49.4	88.8	42.6
DeeBERT	88.8	44.7	47.7	68.4	78.5	60.4	84.4	50.3
WestBERT	89.1	23.9	49.1	59.1	79.9	34.7	87.4	25.9
DeeBERT	86.1	37.4	42.1	63.7	75.7	56.6	82.1	43.5
WestBERT	87.1	15.8	46.0	53.1	76.2	25.7	86.0	21.1

3.4 Comparative Results

Table 1 and Table 2 show the accuracy and inference time of the proposed WestBERT against the baselines on the GLUE dataset. The adaptive inference methods with similar performance were grouped to obtain different expected inference times by adjusting the confidence threshold F. As indicated, WestBERT consistently exhibits better performance than other model compression and advanced adaptive inference. Fewer inference times are used while maintaining similar performance.

In SST-2, WestBERT achieved 91.5% accuracy but only used 35.1% of the original BERT time. The accuracy is 0.7% higher than that of DeeBERT, but the inference time is still 18% less. The accuracy and inference acceleration of DeeBERT are lower than those of WestBERT due to the simple use of cross-entropy to train the early-exiting classifier. Therefore, the early-exiting classifiers of DeeBERT only have weak expressive ability and make incorrect decisions. Another observation is that freezing the BERT backbone network limits the expression ability of BERT in downstream tasks. In addition, the proposed method also performs better in SST-2 compared to the compression model and is more flexible in the tradeoff between model accuracy and inference speed. As indicated, the WestBERT accuracy is 0.2% higher than the best model compression method, BERT-PKD, and 14.9% lower in inference time. Because the model compression method discards the parts of the model permanently, it reduces the informative representation in the model.

Table 2. Experimental results comparison of baseline methods on the development set splits of the GLUE.

Methods	MRPC		QNLI		RTE		MNLI (mm)	
	A/F1%	Time%	Acc%	Time%	Acc%	Time%	Acc%	Time%
BERT-base	89.5	100	91.2	100	71.1	100	83.8	100
DistilBERT	87.5	50.0	85.3	50.0	59.9	50.0	81.5	50
BERT-PKD	85.7	50.0	88.4	50.0	66.5	50.0	-	-
LayerDrop	85.9	50.0	88.4	50.0	65.2	50.0	-	-
DeeBERT	82.9	53.9	87.3	54.5	66.4	73.1	81.5	68.5
WestBERT	83.1	45.0	89.0	41.6	66.8	49.0	83.1	57.9
DeeBERT	81.9	48.5	86.2	50.5	63.8	52.4	79.3	62.7
WestBERT	82.2	35.3	87.0	32.0	65.3	39.1	80.2	37.3
DeeBERT	80.6	41.6	84.3	43.7	60.6	41.0	77.9	59.1
WestBERT	81.3	29.1	85.1	27.8	64.6	36.3	78.3	30.2

Table 3. Results of ablation study of the proposed WestBERT model.

Methods	SST-2		CoLA		QNLI		QQP	
	Acc%	Time%	Mcc%	Time%	Acc%	Time%	A/F1%	Time%
WestBERT-L	91.4	43.2	51.5	72.1	89.2	45.3	87.3	50.2
WestBERT-naive	91.3	39.5	51.2	71.7	89.3	43.4	87.9	41.0
WestBERT	91.5	35.1	53.1	70.1	89.0	41.6	88.8	39.5
WestBERT-L	89.2	35.0	49.6	69.3	88.0	37.7	86.4	40.9
WestBERT-naive	90.0	31.7	50.7	66.7	88.1	37.1	87.4	36.1
WestBERT	90.4	28.2	51.0	65.9	88.2	36.7	87.9	29.0
WestBERT-L	87.0	20.1	45.5	64.0	87.1	34.5	85.5	34.0
WestBERT-naive	88.0	20.3	48.3	57.1	87.0	33.1	86.7	31.8
WestBERT	88.1	19.5	48.5	56.9	87.2	32.0	86.8	23.2

3.5 Ablation Experiments

Table 3 shows the results of the ablation experiments performed with the proposed WestBERT model. WestBERT-L denotes that only the last layer of the BERT backbone network is used as the teacher without a weighted ensemble. Additionally, WestBERT-Naive is composed of a simple summation for all early-exiting classifiers as the teacher without a weighted ensemble.

As indicated, WestBERT takes less inference time than Ensemble-Naive on the SST-2 dataset for different accuracy. With a similar accuracy of 91.5%, WestBERT takes 4.4% less inference time than WestBERT-Naive. The attention weights can learn the importance of each early-exiting classifier instead of the equal contribution of early-exiting classifiers in WestBERT-Naive.

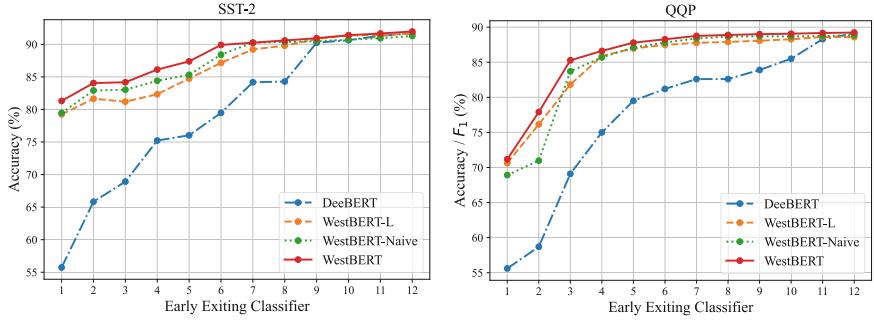


Fig. 3. The performance of different methods in early exiting classifier on SST-2 and QQP.

In addition, the proposed model outperformed WestBERT-L. In the QQP dataset, the proposed model achieves an average accuracy and F1 of 88.8% and an inference time of 39.5%. However, the performance of WestBERT-L is 1.5% lower than that of WestBERT, and the inference time is more than 10.7%. WestBERT-L learns the last layer singularly and does not consider the semantic information diversity of each layer. This demonstrates that the weighted ensemble teacher model can learn rich semantic information from different layers to enhance the performance of the performance of the early-exiting classifier.

3.6 The Effect of Weighted Ensemble Self-distillation

To investigate the effect of the proposed weighted ensemble self-distillation, Fig. 3 compares the proposed WestBERT against DeeBERT and other self-distillation methods.

As shown in Fig. 3, the accuracy of the proposed WestBERT is much higher than that of DeeBERT for the first six early-exiting classifiers. In particular, the accuracy of the first early-exiting classifier of WestBERT is higher than that of DeeBERT by almost 25%. This is because the early-exiting classifier of DeeBERT lacks high-level semantic information, which leads to weak expressiveness of the early-exiting classifier and incorrect decisions. Additionally, the performance of DeeBERT in the lower layers is poor. Thus, if DeeBERT seeks to exit from the lower layer, its prediction performance will be reduced.

The proposed method also has an improvement over WestBERT-L and WestBERT-Naive. This is due to the effective knowledge learned from powerful teachers in different early-exiting classifiers. The weighted ensemble teacher model improves the accuracy of the student models (early-exiting classifier) and makes a correct decision that guarantees improvement of the inference speed of the BERT model.

Figure 4 show the statistics of the number of samples that exit early at different layers. As indicated, the proposed WestBERT tends to exit model inference at earlier classifiers than DeeBERT for similar model performance. Nearly half

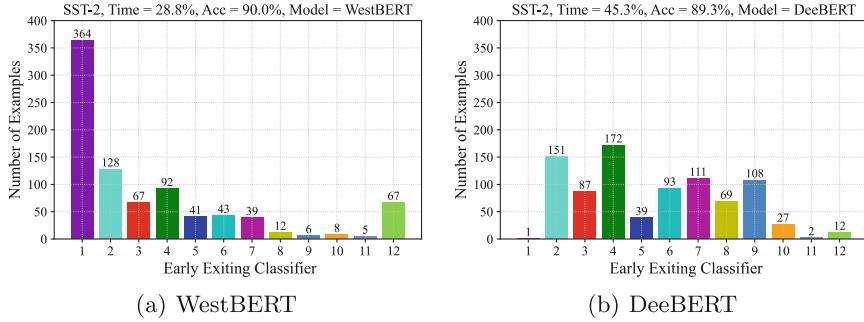


Fig. 4. Distribution of different performances of early exiting classifier on SST-2.

of the samples of the proposed model exited the model immediately in the first layer, which is faster than DeeBERT, and an accuracy of 90% was achieved. Another observation is that the early-exiting classifier performance of WestBERT is much higher than that of DeeBERT, indicating that the proposed model can exit as early as possible while still maintaining high performance.

4 Conclusions

In this paper, adaptive inference with weighted ensemble self-distillation was proposed to accelerate the inference time of BERT. The attention mechanism is used to compose a better teacher that takes full advantage of each student's knowledge (early-exiting classifier). Great teachers can teach better students, and great students can be further ensembled to become better teachers, which can lead to self-enhancing model performance through iteration. As a result, the inference speed is greatly improved while preserving the model performance.

Future work will explore the dynamic adjustment of knowledge distillation losses and supervisory loss weights. Further speed up of model inference is expected.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (NSFC) under Grants Nos. 61702443, 61966038 and 61762091. The authors would like to thank the anonymous reviewers for their constructive comments.

References

1. Bao, S., He, H., Wang, F., Wu, H., Wang, H.: PLATO: pre-trained dialogue generation model with discrete latent variable. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 85–96 (2020)

2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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 (NAACL-HLT 2019), pp. 4171–4186 (2019)
3. Fan, A., Grave, E., Joulin, A.: Reducing transformer depth on demand with structured dropout. In: International Conference on Learning Representations (2019)
4. Hermann, K.M., et al.: Teaching machines to read and comprehend. In: Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS-2015), pp. 1693–1701 (2015)
5. Hinton, G., Vinyals, O., Dean, J.: Distilling the Knowledge in a Neural Network. arXiv preprint [arXiv:1503.02531](https://arxiv.org/abs/1503.02531). (2015)
6. Jawahar, G., Sagot, B., Seddah, D.: What does BERT learn about the structure of language? In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL-2019), pp. 3651–3657 (2019)
7. Lai, K., Yu, L.C., Zhang, X., Wang, J.: Tree-structured regional CNN-LSTM model for dimensional sentiment analysis. IEEE/ACM Trans. Audio Speech Lang. Process. **28**, 581–591 (2019)
8. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: ALBERT: a lite BERT for self-supervised learning of language representations. arXiv preprint [arXiv:1909.11942](https://arxiv.org/abs/1909.11942) (2019)
9. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. arXiv preprint [arXiv:1907.11692](https://arxiv.org/abs/1907.11692) (2019)
10. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint [arXiv:1910.01108](https://arxiv.org/abs/1910.01108) (2019)
11. Schwartz, R., Stanovsky, G., Swayamdipta, S., Dodge, J., Smith, N.A.: The right tool for the job: matching model and instance complexities. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6640–6651 (2020)
12. Sun, S., Cheng, Y., Gan, Z., Liu, J.: Patient knowledge distillation for BERT model compression (2019)
13. Tang, R., Lu, Y., Liu, L., Mou, L., Vechtomova, O., Lin, J.: Distilling task-specific knowledge from BERT into simple neural networks. [arXiv:1903.12136](https://arxiv.org/abs/1903.12136) (2019)
14. Teerapittayanon, S., McDanel, B., Kung, H.T.: BranchyNet: fast inference via early exiting from deep neural networks. In: International Conference on Pattern Recognition, pp. 2464–2469 (2016)
15. Xin, J., Tang, R., Lee, J., Yu, Y., Lin, J.: DeeBERT: dynamic early exiting for accelerating BERT inference. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (2020)
16. Yang, P., Sun, X., Li, W., Ma, S., Wu, W., Wang, H.: SGM: sequence generation model for multi-label classification. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 3915–3926 (2018)
17. Zafrir, O., Boudoukh, G., Izsak, P., Wasserblat, M.: Q8BERT: quantized 8Bit BERT. arXiv preprint [arXiv:1910.06188](https://arxiv.org/abs/1910.06188) (2019)
18. Zhang, Y., Wang, J., Zhang, X.: Learning sentiment sentence representation with multiview attention model. Inf. Sci. **571**, 459–474 (2021)
19. Zhang, Y., Wang, J., Zhang, X.: Personalized sentiment classification of customer reviews via an interactive attributes attention model. Knowl.-Based Syst. **226**, 107135 (2021)

Information Extraction and Knowledge Graph



Employing Sentence Compression to Improve Event Coreference Resolution

Xinyu Chen, Sheng Xu, Peifeng Li, and Qiaoming Zhu^(✉)

School of Computer Sciences and Technology Soochow University, Suzhou, China
{xychen@nlp.sxu.edu.cn}, {pfli,qmzhu@suda.edu.cn}

Abstract. Most previous studies on event coreference resolution usually focused on measuring the similarity between two event sentences. However, a sentence may contain more than one event and the redundant event information will interfere with the calculation of event similarity. To address the above issue, this paper proposes an event coreference resolution framework based on event sentence compression mechanism, which used an AutoEncoder-based model to compress the extracted event sentences based on the event triggers. Meanwhile, the information interaction between the compressed sentences and their original event sentences is used to supplement the missing important information in the compressed sentences. Experimental results on both KBP 2016 and KBP 2017 datasets show that our proposed model outperforms several state-of-the-art baselines.

Keywords: Event coreference resolution · Event sentence compression · Information interaction

1 Introduction

In real-world texts, there are usually a large number of sentences that describe the same event in reality, and an event will be repeatedly mentioned in the documents. When multiple event mentions (an instance of a specific event in texts) point to the same event ontology, these event mentions are coreferent. Event coreference resolution aims to discover the coreferent event mentions in texts and gather them into the coreferent chains. Events are mainly composed of triggers and arguments. Triggers are the main words that can most clearly express the occurrence of events, so each event can be represented as its corresponding triggers. Arguments are the entity mentions involved in the events, such as agents, patients, time, and place. Those coreferent event mentions generally have similar triggers and arguments, as shown in follows:

*E1: With Palestinian cameramen, **shot** and killed by an Israeli soldier on the west bank in mid-April.*

*E2: Those close to miller believe he was **shot** by an Israeli personnel carrier.*

Both the triggers of E1 and E2 are “shot” with the same type “Attack”, and the agents in E1 and E2 are “an Israeli soldier” and “an Israeli personnel carrier”, respectively, which have similar semantics. Additionally, the patient

“he” in E2 refers to the patient “Palestinian cameramen” in E1. Hence, E1 and E2 are coreferent event mentions. Event coreference resolution is beneficial for many NLP applications, such as information extraction [8], topic detection [17] and question answering [18].

Most previous works first used neural networks to measure the similarity of event mention pairs and then judge whether they are coreferent or not. However, most of them simply take the event sentences where the triggers are located as input, ignoring the situation where an event sentence contains multiple events, as shown below:

*S1: If the US **pressured** the Jamaican government to **capture** and **extradite** him, then it was Jamaica that **arrested** and **imprisoned** him, not the US.*

*S2: I don’t see ..., when he was **arrested** by Jamaican police.*

S1 has five triggers that refer to five event mentions, while S2 has one trigger that refers to one event mention. If we directly sent these two event sentences to a neural network model, the redundant event information in the two sentences will interfere with the model and mistakenly regard the event mention “arrested” in S1 and S2 as non-coreferent.

To address the above issue, this paper proposes an event coreference resolution framework based on event sentence compression. We train a trigger-oriented AutoEncoder-based event sentence compression model, and compress the event sentences to obtain compressed sentences, which will only contain a few words related to the specific event. Meanwhile, the information interactions between the compressed sentences and their original event sentences are used to supplement the lost important information in the compressed sentences. Experimental results on both KBP 2016 and KBP 2017 datasets show that our proposed model outperforms several state-of-the-art baselines.

2 Related Work

Most previous studies on event coreference resolution were performed on two popular corpora ACE [16] and KBP [12]. The ACE corpus contains 599 documents with 8 event types and 33 event subtypes. The ACE corpus annotates the coreference information in the document, including event sentences, triggers, arguments, tenses, and polarities. In addition, the ACE corpus also annotates the compressed sentences, which are concise descriptions of the event sentences and usually only contain trigger and argument information. The KBP 2015 corpus contains 369 documents and defines 8 event types and 38 event subtypes. The KBP 2016 and KBP 2017 corpora remove some event subtypes that can be easily identified corresponding coreference relations and only remain 18 event subtypes. Compared with the ACE corpus, the KBP corpora do not annotate any entities, arguments, and other event information. Hence, it is much more challenging to resolve coreferent events in the KBP corpora, especially the KBP 2016 and KBP 2017 corpora.

Early researches on event coreference resolution usually directly use the annotation features of events [2, 6]. Since the above methods rely heavily on manual

annotation information, recent studies pay more attention to the coreference between unlabeled raw texts, which is more challenging and has practical significance. Specifically, Peng et al. [13] designed a pipeline model of event extraction and event coreference resolution, using various vectorization methods (such as Brownian clustering, word vector, dependency parsing, etc.) to represent events as structured event vectors and finally judge whether events are coreferent by comparing the similarity between event vectors. Considering the pipeline model has the error delivery problem, Lu et al. [9] proposed an event coreference resolution model based on joint learning of the Markov logic network. Compared with shallow network and rule-based event coreference resolution methods, Huang et al. [5] incorporated knowledge of argument compatibility from a large number of the unlabeled corpus and improved the performance of coreference resolution. Fang et al. [4] expanded a semantically sparse dataset using data augmentation, and improved the quality of the expanded dataset by reinforcement learning, which effectively improved the performance.

3 Event Coreference Resolution on Sentence Compression

Our event coreference resolution framework consists of three stages: event extraction, event sentence compression, and event coreference resolution. The overall architecture is shown in Fig. 1. Firstly, we use the event sentences, the triggers, the arguments, and the compressed sentences annotated in the ACE 2005 corpus to train an event sentence compression model, as shown in part (a) of Fig. 1. Secondly, we extract event mentions from the raw text in the KBP corpus as shown in part (b) to obtain the event sentences, their triggers, and event subtypes. Thirdly, the event sentences and triggers are sent to the Event Sentence Compression model to obtain the compressed sentences. Finally, the compressed sentences and their original event sentences are encoded by BERT and used to predict whether two event mentions are coreferent through the interaction attention layer. Since the same event subtype is a necessary condition for coreferent events, we only take the event pairs with the same subtype as candidate coreferent event pairs.

3.1 Event Extraction

We extract event information from unlabeled raw texts and then detect all the event mentions in the documents. Similar to Fang's method [4], we use a BiLSTM-based classifier as the event extractor, which encodes the input sentences by BiLSTM, and outputs the extracted event sentences, triggers, and subtypes by a Softmax layer. Different from Fang, we use weighted voting method to obtain final event subtype in ensemble learning.

3.2 Event Sentence Compression

Generally speaking, sentence compression is the task of shortening a sentence while retaining its meaning. Inspired by Malireddy et al. [11], we propose an

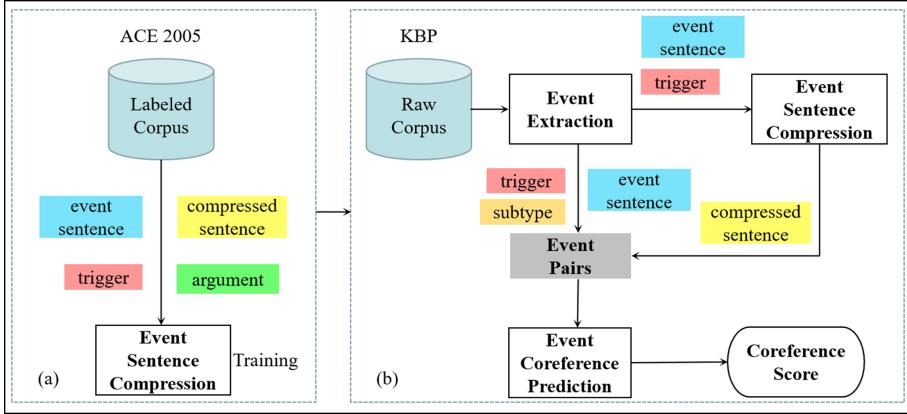


Fig. 1. Overall flow chart of our event coreference resolution

event sentence compression model, called ESCA (Event Sentence Compression AutoEncoder), shown in Fig. 2 mainly including two components: (a) Label Generator and (b) AutoEncoder. Different from Malireddy, our ESCA takes trigger information as the core and uses argument and compressed sentence as important auxiliary information. Since the ACE corpus contains event sentences (tag: ldc_scope) and their compressed sentences (tag: extent) (an example is as follows), which is not available in the KBP corpus. Therefore, we train our ESCA model on the ACE corpus and apply it to the KBP corpus.

S3: Well, I I guess now that that he's he's **died** – it'll be, uh, or it it should be a change for the PLO. (event sentence)

S4: he's **died**. (compressed sentence)

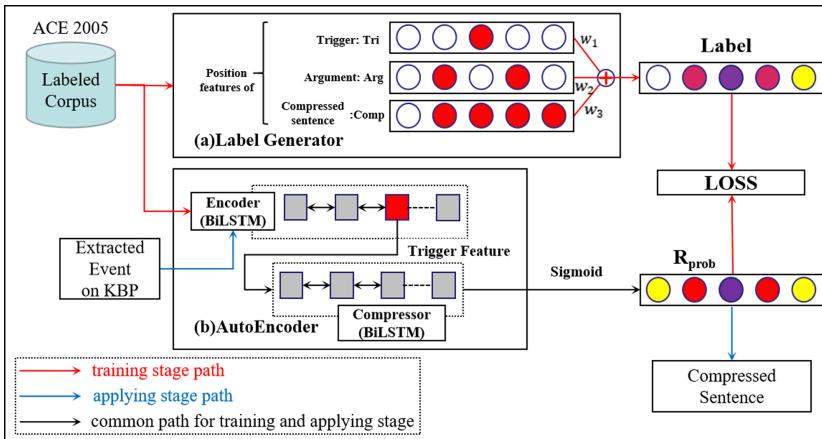


Fig. 2. Architecture of ESCA

In the training stage, the Label Generator receives the event triggers, arguments, and compressed sentence as input, and performs sequence annotation by comparing them with the information of the event sentence to obtain their position features. Specifically, the positions of trigger words, argument words, and compressed sentence words in event sentences are labeled as 1 (as shown by red circle in Fig. 2 (a)), otherwise they are labeled as 0 (as shown by white circle in Fig. 2 (a)), representing their position features simply as *Tri*, *Arg*, and *Comp* respectively. Then the weighted summation of *Tri*, *Arg*, and *Comp* performed to generate *Label*, which is used as the training label of the compression model AutoEncoder. The generation process of *Label* can be formalized as follows:

$$\text{Label} = w_1 \cdot \text{Tri} + w_2 \cdot \text{Arg} + w_3 \cdot \text{Comp} \quad (1)$$

Here, we set $w_1 > \max\{w_2, w_3\}$ since trigger is the core feature of an event, so that each component value in *Label* reflects the different importance of each word in an event sentence. When we provide Label Generator with an event sentence contains multiple events, different *Label* will be obtained according to different triggers, arguments, and compressed sentences.

AE (AutoEncoder) receives the event sentences and the triggers as input and compresses the sentences around the triggers to obtain R_{prob} , which is a probability representation to determine whether each word in an event sentence remains in the compressed sentence. The words corresponding to the positions close to 0 in R_{prob} will be discarded, and that close to 1 will remain. Specifically, AE has two components: Encoder and Compressor, both of which are composed of a BiLSTM. Encoder can encode event sentences with the trigger-central index. Generally, in many NLP tasks, the output of the last position of LSTM is taken as the feature vector. Different from them, we extract the feature vector of the position where the trigger is located by its index since the trigger is taken as the core feature. And the extracted feature vector also contains context information of the trigger. Then, this feature vector is sent to Compressor for decoding to obtain the compressed hidden state. Finally, the hidden state is transformed by a fully-connected layer, and the retention probability of each word in the sentence is obtained by the Sigmoid activate function, using R_{prob} represent it. We motivate AE to learn the ability to compress information related to events by reducing MSE (Mean-Square-Error) loss between *Label* and R_{prob} .

After trained by *Label*, the keywords (e.g., trigger and arguments) in the sentence that have a high retention probability will exist in the compressed sentence, while other words with retention probability close to zero will be removed. Then, we sent sentences in the KBP corpus with their triggers obtained from the stage of event extraction into AE to obtain compressed sentences.

3.3 Event Coreference Resolution

Our event coreference resolution model Coref-CS (**C**oreference Resolution based on **C**ompressed **S**entence) takes compressed sentence pairs and their original sentences as input. Coref-CS includes three components: Input Layer, Interaction Attention Layer, and Classification Layer, as shown in Fig. 3.

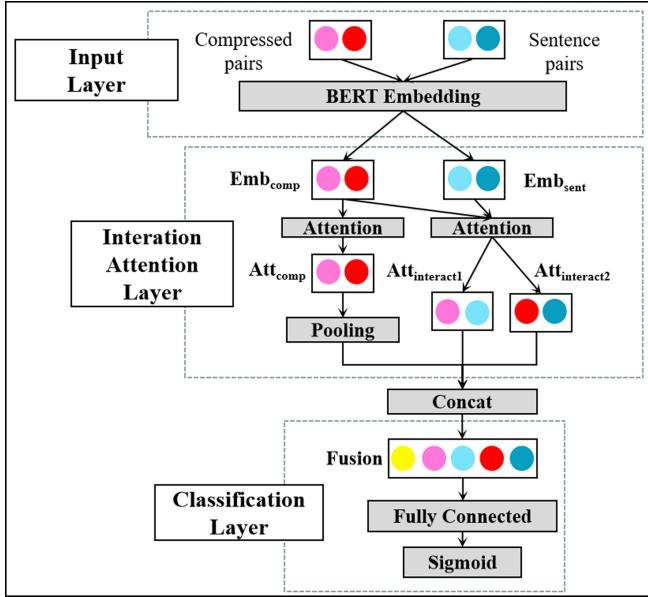


Fig. 3. Architecture of Coref-CS

Input Layer: We apply BERT to encode each compressed sentence pairs and their original event sentence pairs to get their embeddings Emb_{comp} , Emb_{sent} , respectively. Here, we use Emb_{comp_i} ($i \in \{1, 2\}$) to refer single sentence embedding in the compressed sentence pairs embedding Emb_{comp} , Emb_{sent} is the same.

Interaction Attention Layer: The attention mechanism is usually used to assign a weight to each word in a sentence, and irrelevant words will get smaller weights, thus reducing the noise interference. For further denoising and capture coreference relationship between compressed sentence pairs, we use the multi-head attention mechanism to capture the common features between their embedding Emb_{comp} as follows:

$$Att_{comp_i} = \text{Attention}(Emb_{comp_i}, Emb_{comp_j}), i \in \{1, 2\}, j \in \{2, 1\} \quad (2)$$

Since the subtasks event extraction and event sentence compression are upstream of event coreference resolution, there are cascading errors. For example, those missing triggers in event extraction will lead to the loss of important information in the compressed sentence, because sentence compression model ESCA depends on the trigger information, which is not conducive to judging whether the event pair is coreferent or not. To alleviate this issue, we interact the information of the compressed sentences and their original event sentences, so that the information lost in compressed sentences can be properly supplemented from original event sentences.

Specifically, we input the embedding of compressed sentence pairs Emb_{comp} and their original sentence pairs Emb_{sent} into an Attention mechanism to obtain

the interactive information representation between them, as shown below:

$$Att_{interacti} = Attention(Emb_{comp_i}, Emb_{sent_i}), i \in \{1, 2\} \quad (3)$$

Then, we use the Max Pooling operation to express the most associated information between two compressed sentences, but it will ignore some important auxiliary information, so we perform both Max Pooling and Avg Pooling operation on Att_{comp} to obtain the expression of the relevance of two compressed sentences. As shown by yellow circle in Fig. 3, which represents the result of Max Pooling and Avg Pooling operations.

Classification Layer: We concatenate the relevant information of the compressed sentences and the interaction information $Att_{interact1}$, $Att_{interact2}$ to obtain the final fusion features $Fusion$, and feed it to a fully-connected layer to identify the coreferent events, as formalized below:

$$Fusion = Concat(Pooling(Att_{comp}), Att_{interact1}, Att_{interact2}) \quad (4)$$

$$Score = Sigmoid(W_{out} \cdot Fusion + b_{out}) \quad (5)$$

Here, W_{out} is the weight matrix and b_{out} is the bias of fully-connected layer.

Since the positive and negative samples in the dataset are very unbalanced, we choose focal loss [7] as the loss function.

$$FocalLoss = -\alpha(1 - p)^\gamma \log(p) \quad (6)$$

Finally, we use the undirected graph connection method to handle the results and link the even coreference pairs to each other to build an event chain.

4 Experimentation

4.1 Experimental Settings

Following previous work [4,5,9], we use the KBP 2015 dataset as the training set and the official complete test set of KBP 2016 and KBP 2017 as the test set and choose 10% of the training set as the development set. Note that we did not use any annotated information in the official test set. Following the previous work [4,5,9], we use MUC [15], B³ [1], BLANC [14] and CEAF_e [10] to evaluate the performance of our Coref-CS and also report the average scores (AVG) of the above four metrics. Comprehensive use of the above four metrics and their AVG can more objectively measure the performance of the model.

We use PyTorch as deep learning framework. In the training of the model, We set the training epochs of Coref-CS as 20 rounds, each round takes about 50 min and 10000 Mib GPU memory. We set the learning rate as 10^{-5} , and use Adam optimizer to update the parameters. Each word is embedded into a 768-dimensional vector by BERT. The number of heads of attention mechanism is set to 3. Besides, w_1 , w_2 and w_3 in Eq. (1) are set to 0.6, 0.3 and 0.2, respectively.

4.2 Results on Event Extraction

The event extraction systems proposed by Lu et al. [9], Huang et al. [5] and Fang et al. [4] are state-of-the-art baselines, which achieve excellent performance in KBP 2016 dataset and KBP 2017 dataset. Therefore, in the stage of event extraction, we compare our event extractor with their event extraction systems. Table 1 shows the F1 scores of event extraction on KBP 2016 and KBP 2017 datasets, which shows that our event extractor has achieved comparable performance. It is fair to compare the performance of event coreference resolution based on such similar event extraction results.

Table 1. F1 Scores of Event Extraction.

	Lu	Huang	Fang	Ours
KBP 2016	45.02		44.71	45.37
KBP 2017		48.14	47.82	48.28

4.3 Results on Event Coreference Resolution

To verify the performance of Coref-SC, we select the state-of-the-art baseline Lu et al. [9], Huang et al. [5] and Fang et al. [4] on the KBP 2016 and KBP 2017 datasets for comparison. It is worth mentioning that Lu and Huang only reported the experimental results on a single KBP dataset. Besides, we also use two other model as baselines for comparison, one is the pre-training model BERT, another is the cross-document model proposed by Eirew1 et al. [3], we reproduce their model and apply it on the KBP dataset. The performance comparison between our model and the above four baselines is shown in Table 2. The results show that our Coref-CS outperforms all the baselines significantly ($P < 0.01$) and this indicates that the sentence compression mechanism can help the model to denoise the event sentence and is effective for event coreference resolution.

Table 2. Comparison of experimental results on Event Coreference Resolution.

KBP 2016						KBP 2017					
System	MUC	B ³	BLANC	CEAF _e	AVG	System	MUC	B ³	BLANC	CEAF _e	AVG
Lu	27.41	40.90	25.00	39.00	33.08	Huang	35.66	43.20	32.43	40.02	36.75
Fang	34.76	49.07	36.29	43.24	40.84	Fang	37.58	45.83	41.32	38.03	40.69
BERT	36.73	46.98	39.69	40.94	41.08	BERT	42.21	46.77	40.07	37.04	41.52
Eirew1	37.67	49.37	38.65	41.89	41.90	Eirewl	43.63	46.87	40.29	38.03	42.21
Coref-CS	39.50	48.52	33.30	45.90	43.41	Coref-CS	47.15	46.34	40.91	39.95	43.59

Compared with the best baseline Fang, our Coref-CS improves the AVG by 2.57 and 2.90 on KBP 2016 and KBP 2017 datasets, respectively. Fang extracted

a large number of positive samples from a large unlabeled corpus to balance the positive and negative examples and used the reinforcement learning method to enhance the quality of samples, effectively improving the performance of event coreference resolution. Different from Fang, our Coref-CS compresses the event sentences to obtain compressed sentences with more concise information, which greatly reduces the interference of noisy information in event sentences. In addition, considering the possibility that important information may be compressed and lost in compressed sentences, we use the interaction information between compressed sentences and event sentences to recover the lost useful information in compressed sentences. The significant improvement on the MUC metric (4.74 and 9.57) verifies that our model can extract more coreferent event mention pairs and then improve the overall performance.

In comparison with BERT, our Coref-CS still improves average metric AVG by 2.33 and 2.07 on KBP 2016 and KBP 2017 datasets, respectively. This indicates that on the basis of BERT’s feature extraction of event sentences and compressed sentences, we can extract the common features between compressed sentence pairs with more concise information and information interactions, which can effectively improve the resolution performance of event coreference. It is worth mentioning that, compared with the baseline of Fang, our model has significantly improved MUC, especially on the KBP 2017 dataset. This may be due to the fact that the triggers extracted from KBP 2017 are sparse in comparison with those from KBP 2016 in the stage of event extraction, especially the event mentions on a single-event-mention chain (i.e., non-coreferent event).

In comparison with the baseline Eirewl, our Coref-CS improves average metric AVG by 1.51 and 1.38 on KBP 2016 and KBP 2017 datasets, respectively. This baseline also uses BERT-based model to extract event mention features for event coreference resolution and achieve comparable performance with the baseline BERT. It indicates that the methods between document-level and cross-document event coreference resolution can be used for reference each other and sentence compression may improve performance in cross-document field.

5 Analysis

To further analyze our Coref-CS model, we conduct the comparative experiments under two different settings, one is ablation experiments, and the other is experiments of different sentence compression strategies.

5.1 Ablation Study

To justify the effectiveness of the sentence compression mechanism and the interaction mechanism, we performed ablation experiments on KBP 2016 and KBP 2017 datasets. Table 3 shows the results of several ablation experiments.

The results show that, when deleting the sentence compression mechanism, i.e., the input of the model only has event sentences, the AVGs on KBP 2016 and KBP 2017 datasets are reduced by 1.3 and 1.46, respectively. This justifies

Table 3. Performance comparison on different modules.

System	KBP 2016					KBP 2017				
	MUC	B ³	BLANC	CEAF _e	AVG	MUC	B ³	BLANC	CEAF _e	AVG
Coref-CS	39.50	48.52	39.40	45.90	43.41	47.15	46.34	40.91	39.95	43.59
-Compress	37.16	50.00	39.63	41.63	42.11	43.1	47.62	40.7	37.11	42.13
-Interaction	38.83	49.28	39.71	42.76	42.65	45.01	47.61	41.36	38.09	43.02

that the sentence compression mechanism plays an important role in reducing data noise since less interference from noisy data is helpful to distinguish whether two events are coreferent or not more accurately. Additionally, when deleting the interaction mechanism, i.e., the input of the model is only compressed sentence pairs, the AVGs on KBP 2016 and KBP 2017 datasets are reduced by 0.76 and 0.57, respectively. This indicates that the interaction mechanism can effectively help the compressed sentence to expand useful information.

5.2 Analysis on Different Compression Strategies

To prove the effectiveness of our sentence compression mechanism, we also conduct comparative experiments on different sentence compression strategies on KBP 2016 and KBP 2017 datasets. We reproduced the state-of-the-art compression model SCAR proposed by Malireddy et al. [11] and sent its results to our Coref-CS. The result comparison is shown in Table 4. The results show that our ESCA-based event coreference resolution outperforms the SCAR-based model in all four metrics and AVG.

Table 4. Comparison on different compress strategy.

System	KBP 2016					KBP 2017				
	MUC	B ³	BLANC	CEAF _e	AVG	MUC	B ³	BLANC	CEAF _e	AVG
ESCA	39.50	48.82	39.40	45.90	43.41	47.15	46.34	40.91	39.95	43.59
SCAR	37.47	48.27	38.32	40.89	41.24	42.50	46.12	39.76	37.27	41.41

The SCAR model mainly trains the model by controlling the ratio of compressed sentences to original event sentences. In their experiments, this ratio is set to 0.4, which makes the ratio of SCAR compressed event sentences to compressed sentences stable. However, due to the variability of the argument number in each event in the event coreference resolution task, it is not suitable to use the ratio of compressed and original event sentences to control the compression of sentences, especially when we want to obtain the compressed sentences containing only event-related information. In addition, although the SCAR model is designed to compress important information in original sentences, SCAR can't treat different triggers differently when the original event sentence contains triggers of multiple events, which makes the compressed sentence mixed with other events. Therefore, it can't reduce noise effectively.

5.3 Case Study

In this subsection, we give a few examples to analyze the effectiveness of our sentence compression mechanism and interaction mechanism.

Using our ESCA, the two original event sentences S1 and S2 in Sect. 1 are compressed as S5 and S6 as follows when the two event triggers are “capture” and “arrested”, respectively. Since the triggers “capture” and “arrested” in those two compressed sentences S5 and S6 have similar meanings and these two sentences have the same words “Jamaican” and “he/him”, Coref-CS is relatively easy to identify them as coreferent. If we use SCAR to compress S1 and S2, the compressed sentences are S7 and S8. Obviously, the two compressed sentences S7 and S8 are mixed with other trigger words (e.g., “extradite” and “imprisoned”) and arguments (e.g., “Al Jazeera” and “US”), and the difference from the original sentences is only the reduction of word numbers. This is because SCAR can’t accurately distinguish the importance of different triggers in different events, and treat them all as important information.

*S5: Jamaican government **capture** him.*

*S6: he was **arrested** by Jamaican police.*

*S7: US **capture** and **extradite** him Jamaica **arrested** and **imprisoned**.*

*S8: Al Jazeera put out make a claim US has arrested anyone **imprisoned** Jamaican police **extradited**.*

The interaction mechanism is also useful in our event coreference resolution model. For example, the event sentences S9 and S10 are compressed as S11 and S12, respectively, using our ESCA. Obviously, the important argument information “Red Cross” is lost in S12 and the information of two compressed sentences is unbalanced. In this case, only using two compressed sentences to capture their common features is unfavorable to recognize their coreferent relation. Therefore, our model uses the interaction mechanism of event sentences and compressed sentences, the argument information “Red Cross” is added to the compressed sentence S12 from its original event sentence, which makes the model accurately judge this event mention pairs as coreferent.

*S9: also the **donation** is to the American Red Cross even though the Philippines has its own Red Cross organization*

*S10: **donate** to the Red Cross*

*S11: **donation** to the American Red own organization*

*S12: **donate***

6 Conclusions

This paper proposed an event sentence compression mechanism and an interaction mechanism to improve the performance of event coreference resolution. Firstly, the compressed sentences containing event-related information are obtained by sentence compression, and then the interaction mechanism is used to supplement event sentences and compressed sentences to recover the missing information, hence improving the performance of event coreference resolution. Experimental results on both the KBP 2016 and KBP 2017 datasets show that

our proposed model Coref-CS outperforms several state-of-the-art baselines. Our future work will focus on how to extend our model to the field of cross-document or cross-language event coreference resolution due to the limitation of our work only in document level and monolingual event coreference resolution.

Acknowledgments. The authors would like to thank the three anonymous reviewers for their comments on this paper. This research was supported by the National Natural Science Foundation of China (No. 61836007, 61772354 and 61773276.), and the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).

References

1. Bagga, A.: Evaluation of coreferences and coreference resolution systems. In: LREC, pp. 563–572 (1998)
2. Bejan, C.A., Harabagiu, S.M.: Unsupervised event coreference resolution with rich linguistic features. In: ACL, pp. 1412–1422 (2010)
3. Eirew, A., Cattan, A., Dagan, I.: WEC: deriving a large-scale cross-document event coreference dataset from Wikipedia. In: NAACL-HLT, pp. 2498–2510 (2021)
4. Fang, J., Li, P.: Data augmentation with reinforcement learning for document-level event coreference resolution. In: NLPCC (1), pp. 751–763 (2020)
5. Huang, Y.J., Lu, J., Kurohashi, S., Ng, V.: Improving event coreference resolution by learning argument compatibility from unlabeled data. In: NAACL-HLT (1), pp. 785–795 (2019)
6. Krause, S., Xu, F., Uszkoreit, H., Weissenborn, D.: Event linking with sentential features from convolutional neural networks. In: CoNLL, pp. 239–249 (2016)
7. Lin, T., Goyal, P., Girshick, R.B., He, K., Dollár, P.: Focal loss for dense object detection. IEEE Trans. Pattern Anal. Mach. Intell. **42**(2), 318–327 (2020)
8. Liu, S., Chen, Y., Liu, K., Zhao, J.: Exploiting argument information to improve event detection via supervised attention mechanisms. In: ACL (1), pp. 1789–1798 (2017)
9. Lu, J., Ng, V.: Joint learning for event coreference resolution. In: ACL (1), pp. 90–101 (2017)
10. Luo, X.: On coreference resolution performance metrics. In: HLT/EMNLP, pp. 25–32 (2005)
11. Malireddy, C., Maniar, T., Shrivastava, M.: SCAR: sentence compression using autoencoders for reconstruction. In: ACL (student), pp. 88–94 (2020)
12. Mitamura, T., Liu, Z., Hovy, E.H.: Overview of TAC KBP 2015 event nugget track. In: TAC (2015)
13. Peng, H., Song, Y., Roth, D.: Event detection and co-reference with minimal supervision. In: EMNLP, pp. 392–402 (2016)
14. Recasens, M., Hovy, E.H.: BLANC: implementing the rand index for coreference evaluation. Nat. Lang. Eng. **17**(4), 485–510 (2011)
15. Vilain, M.B., Burger, J.D., Aberdeen, J.S., Connolly, D., Hirschman, L.: A model-theoretic coreference scoring scheme. In: MUC, pp. 45–52 (1995)
16. Walker, C., Strassel, S., Medero, J., Maeda, K.: ACE 2005 multilingual training corpus. Prog. Theor. Phys. Suppl. **110**(110), 261–276 (2006)
17. Wayne, C.L.: Topic detection & tracking: a case study in corpuses creation & evaluation methodologies. In: LREC, pp. 111–116 (1998)
18. Weissenborn, D., Wiese, G., Seiffe, L.: Making neural QA as simple as possible but not simpler. In: CoNLL, pp. 271–280 (2017)



BRCEA: Bootstrapping Relation-Aware Cross-Lingual Entity Alignment

Yujing Zhang^{1,2(✉)}, Feng Zhou^{1,2}, and Xiaoyong Y. Li³

¹ Key Laboratory of Intelligent Telecommunications Software and Multimedia,
Beijing, China

{z.yj,zfeng}@bupt.edu.cn

² School of Computer Science, Beijing University of Posts and Telecommunications,
Beijing, China

³ School of Cyberspace Security, Beijing University of Posts
and Telecommunications, Beijing, China

lixiaoyong@bupt.edu.cn

Abstract. Entity alignment aims to align entities referring to the same identity in the real world across different Knowledge Graphs (KGs), which is a fundamental task of KG construction and KG fusion. Recent works focus on embedding-based approaches. With the pre-aligned entity pairs, these approaches mainly embed entities based on relation triples to capture structural information and then try to refine the entity embeddings by self-characteristics contained in attribute triples. However, insufficient training data, diverse expressions of attributes, and different importance between self-characteristics and structural information in different KGs are three obstacles to entity embedding. In order to tackle these problems, we propose a novel Bootstrapping Relation-aware model for Cross-lingual Entity Alignment using both relation triples and attribute triples of KGs (BRCEA). Firstly, given the base prior alignments, it separately embeds entities from two aspects, namely self-characteristics and structural information. Then, bootstrapping component discovers two sets of new alignments. Finally, the two sets will be used to construct new training data for the next iteration to overcome the sparsity of training data. We performed our model on several real-world datasets, and the results show that our model outperforms the state-of-art models for cross-lingual entity alignment.

Keywords: Cross-lingual entity alignment · BRCEA · RDGCN · GCN-Align · Bootstrap

1 Introduction

Inspired by the Semantic Web, Google puts forward the concept of KG, which is used to improve the search quality of search engines and increase the search experience of users. As of today, researchers have built various KGs in medicine, education, e-commerce, and many other fields, in order to take full advantage

of KGs in recommendation, question answering, and information retrieval [1], promoting artificial intelligence to leap from perceptual intelligence to cognitive intelligence.

Various KGs store rich real-world facts in structured forms including relation triple (T_r) and attribute triple (T_a). Different KGs cover complementary facts of various domains from different data sources and using different languages, which makes it necessary but challenging to bridge the language gap and integrate multiple KGs. Therefore, the cross-lingual entity alignment is designed to automatically align entities that refer to the same real-world facts in cross-lingual KGs.

Cross-lingual entity alignment is a fundamental task of KG construction and KG fusion, which has been dominated by embedding-based approaches, which embed entities and then directly compute the entity similarities based on the embeddings. Among them, translation-based models [3–6] are the most popular. For example, given pre-alignments, JAPE [2] applies TransE [3] on T_r to translate head entity h to tail entity t by relationship r through $h + r \approx t$, and utilizes attribute correlations on T_a to refine embeddings. Translation-based models are easy to implement and train, but difficult to handle complex relationships. Simultaneously, graph-based methods have been proposed and can achieve promising performance, which points a new and promising way for entity alignment. GCN-Align [7] trains Graph Convolutional Networks (GCNs) [8] to learn entity embeddings into a unified vector space. However, since spectrum-based GCN can only deal with undirected graphs with a single relationship, GCN-Align is also unable to properly model T_r . To model multi-relation graphs, Relational Graph Convolutional Networks (R-GCNs) [10] employs an adjacent matrix for every relationship, resulting in difficult training for excessive parameters. Relation-aware Dual-Graph Convolutional Network (RDGCN) [11], adopts convolution operations between the primal entity graph and its dual relation graph [12] interactively to learn entity embeddings and edge embeddings, but it ignores T_a and still suffers from the lack of labeled training data. In conclusion, there are three main problems for entity alignment:

- The self-characteristics (i.e. attribute information) of entities is complex because cross-lingual KGs usually have different attribute value expressions in data structure and granularity. For examples, a distance can be described as “2 km” or “1.23mi”, and a birthday can be described as “1997.09.10” or “97-9-10”.
- The multi-relation of structure triples makes it hard to model structural information properly.
- Insufficient pre-aligned entity pairs make supervised models lack training data.

To solve the above challenges, we propose a Bootstrapping Relation-aware Cross-lingual Entity Alignment model (BRCEA) using both relation triples and attribute triples. The main contributions of our model are as follows:

- We propose a cross-lingual entity embedding model, which embeds entities from structural information and self-characteristic independently. It employs

improved GCN-Align to capture the self-characteristics of entities. Considering complex multiple-relations, it employs the RDGCN component to capture relation-aware structural information.

- We propose a semi-supervised framework to do entity alignments. It employs a bootstrapping component to generate two sets of discovered alignments respectively. Then, we utilize them to retrain entity embedding components in the next iteration.
- We evaluated our model on real-world cross-lingual datasets. Experiments reveal that BRCEA yields better performance over the state-of-art embedding-based models.

The rest of this paper is organized as follows. Section 2 discuss some related works. Section 3 states the problem definition and describes our approach in detail. Section 4 and Sect. 5 reports experimental results. Finally, we conclude this paper with future work in Sect. 6.

2 Related Work

2.1 KG Embedding

Knowledge graph embedding is to embed components of a KG including entities and relations into a continuous vector space, which has gained massive attention in a variety of downstream tasks such as relation extraction and KG completion. TransE [3] interprets a relation as the translation from its head entity to its tail entity. TransE is feasible to model 1-to-1 relations but difficult to model complex relations. To improve TransE, researchers proposed TransH, TransR, and TransD, which can better model complex relations to some extent. Additionally, researchers are trying GCN-based models [10, 12] to embed KGs.

2.2 Graph Convolutional Network

Since Convolutional Neural Network (CNN) can only process Euclidean spatial data, graph convolutional neural network (GCN) was proposed to deal with non-Euclidean spatial data, such as graph data, which GCNs can be categorized into the spectral-based methods and the spatial-based methods [14]. Based on GCNs, many works [8, 9] have gained promising achievements. GCN-Align [7] first applied spectral GCM to KG embedding, which can only deal with undirected one-relationship KG because spectral GCN requires the normalized graph Laplace operator to be a symmetric positive semi-definite matrix. In order to properly model multi-relation KGs, R-GCN [10] employs an adjacent matrix for each relationship, but it is hard to be trained for its excessive parameters.

RDGCN [11] discussed in Sect. 1 introduces dual-graph to learn multi-relation KG embedding. The primal graph is generated by putting G_1 and G_2 together, where the vertex set and edge set are the unions of all entities and edges in G_1 and G_2 respectively, and its dual graph is generated as: 1) each relation in G_1 and G_2 is mapped to a vertex; 2) there is an edge between i -th vertex and

j -th vertex if r_i and r_j share the same head or tail entity. RDGCN adds two dual-attention-layer-primal-attention-layer before GCN-Align so as to capture complex structural information.

Inspired by the above GCNs, our approach models the attribute information by the improved GCN-Align and models the structural information by RDGCN.

2.3 Cross-Lingual Entity Alignment

Here we introduce the state-of-art Entity Alignment models most related to ours, and discuss the main drawbacks below:

JAPE [2] follows TransE and attribute correlation to embed entities in two KGs into the same vector space. It suffers from insufficient pre-alignments and poor ability of modeling complex relations.

GCN-Align [7] follows GCNs to embed both structural information and attribute information independently, and get final embeddings by giving different important ratios to two entity embeddings sets. The drawback is that it can't properly model multi-relation KGs and also suffers from insufficient pre-alignments.

RDGCN [11] only takes advantage of structural information and ignores attribute information. It introduces a relation-aware dual-graph to capture attribute information and adopts convolution operations between the primal entity graph and its dual relation graph interactively to learn entity embeddings and edge embeddings respectively. And it still suffers from insufficient training data.

BootEA [13] proposes a bootstrapping approach to embedding-based entity alignment, which uses TransE to embed entities and bootstrapping to enrich the pre-alignments. During the bootstrapping process, it proposed alignment editing to reduce labeling errors. The drawback is that BootEA ignores attribute information.

3 Our Approach: BRCEA

In the beginning, we state the formulation of cross-lingual entity alignment. After that, we give a brief introduction to the framework of the proposed model. Then we detail the components of BRCEA.

3.1 Problem Formulation

Formally, we represent two cross-lingual KGs as G_1 and G_2 . Let E_1 be the entity set of G_1 and E_2 be the entity set of G_2 . Cross-lingual entity alignment aims to find entity pairs $A = \{(e_1, e_2) \in E_1 \times E_2 | e_1 \sim_R e_2\}$, where an equivalence relation \sim_R hold between e_1 and e_2 . In this case, we collect a subset pre-aligned entity pairs P of A as ‘Base Alignments’. Our approach it designed to align the rest unaligned entities of A . To be clear, our model is based on the so-called one-to-one entity alignment hypothesis [13] that one entity in G_1 can be aligned only one entity in G_2 .

3.2 Overview

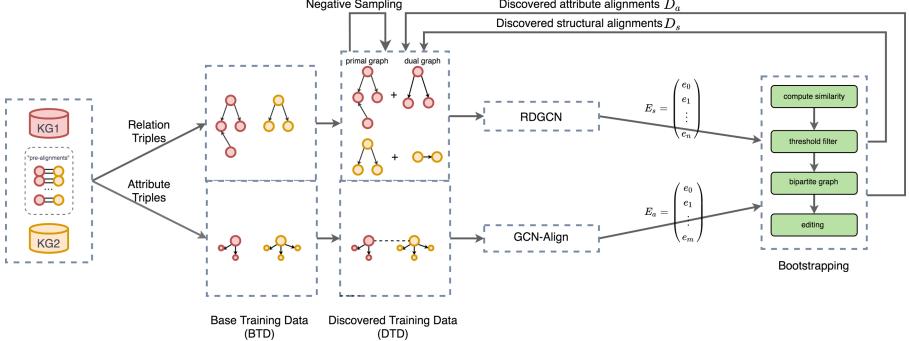


Fig. 1. Framework of the BRCT

Our approach aims to encode entities of Cross-lingual KGs into a unified embedding space where the latent aligned entities are expected to be close. We assume that: 1) equivalent entities tend to have similar attributes. 2) an entity is influenced by its related entities, the so-called neighborhood. The framework of our proposed approach is shown in Fig. 1. Given KG_1 and KG_2 with T_r , T_a and pre-alignments P . Firstly, training data includes Base Training Data (BTD) and Discovered Training Data (DTD). We sample negative relation triples based on training data. Secondly, in each training iteration, we learn entity embeddings E_s and E_a from the structure component (i.e. RDGCN) and attribute component independently. Thirdly, the bootstrapping component discovers new attribute and structure alignments D_s and D_a , which are used to construct DTD of the next iteration. In our model, we co-train attribute and structure components to learn entity embeddings from two views, and we use bootstrapping to discover new alignments to overcome the sparsity of training data.

3.3 Training Data Construction

Training data construction contains training data completing and negative sampling. We take the given pre-alignments P for BTD. The DTD is initialized empty at the first iteration and will be updated in each iteration. Both BTD and DTD contain attribute and structure information. The complete training data is obtained by add BTD and DTD. After that, we only do negative sampling from relation triples. Based on the complete training data, given a relation triple $(h, r, t) \in T_s^+$, negative sampling is to replace either h or r with an arbitrary entity [3], where T_s^+ represents positive samples of T_r . Our approach adopts ϵ -truncated uniform negative sampling idea. To replace entity x where $(h, r, x) \in T_s^+$, we rule out x_2 where $(h, r, x_2) \in T_s^+$ as a negative sample firstly. Then, we calculate entity similarity by Manhattan distance and choose the first

k similar entities as candidates, where $k = \lceil (1 - \epsilon)N \rceil$, N is the number of entities in the KG. In this way, we only sample k entities having high similarities with x as negative samples.

3.4 Attribute Embedding

As mentioned in Sect. 1, attribute information is complex. To avoid introducing excessive noise, we simplify attribute triples to $T_r = \langle e, a \rangle$, where e, a represent entity, attribute name respectively. Inspired by researcher GCN-Align [7], we use 2-layer GCNs to generate entity embeddings. Let H represent the attribute feature matrices of all entities, the convolutional computation is defined as:

$$H^{(l+1)} = \sigma(\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} [H^{(l)} W^{(l)}]) \quad (1)$$

where σ is an activation function, $ReLU(x) = \max(0, x)$; $A \in R^{n \times n}$ is an adjacency matrix of the whole graph; $W^{(l)}$ is the weight matrices for attribute features in the l -th layer of GCN. To improve the ability of modeling attribute information in multi-relation graph, we design a new way to compute the adjacency matrix A of a KG so that $a_{ij} \in A$ indicates the extent of alignment information propagated from the i -th entity to j -th entity. Considering a relation triple (e_i, r_k, e_j) where r_k is a common relation that most entities have, we design an inverse relation functionality to lower the excessive influence, for each relation:

$$irfunc(r) = \frac{\#Tripples_of_r}{1 + \#Relations_of_belongedKG} \quad (2)$$

where $\#Tripples_of_r$ is the number of triples of relation r and $\#Relations_of_belongedKG$ is the number of relations of KG to which relation r belongs. Along with fun and $ifun$ [7], $a_{ij} \in A$ can be computed as:

$$a_{ij} = \sum_{\langle e_i, r, e_j \rangle \in G} ifun(r) \times irfunc(r) + \sum_{\langle e_j, r, e_i \rangle \in G} fun(r) \times irfunc(r) \quad (3)$$

In this way, we reduce the propagation weight through universal relations and raise the propagation weight through universal relations, which enhances the importance of the special relationship.

3.5 Bootstrapping

As an implementation of semi-supervised learning, the bootstrapping component discovers new alignments and adds them to the BTD to overcome the sparsity of training data. The brief procedure is shown in picture 1. Given an entity embedding set E , our bootstrapping method firstly computes entity similarities called sim_mat using Manhattan distance. Then sim_mat is filtered by threshold th to ignore the possibility of aligning entities with similarity below th and choose the $topK$ similar entities as candidates for each entity. After that, we construct a graph, of which nodes represent entities from KG_1 and KG_2 and edges with

weight represent aligned possibility and conduct we conduct Maximum Matching Algorithm on it to get the final aligned graph. Finally, to avoid semantic drift and iteration conflicts of new alignments, we employ error-correction mechanism in BootEA to editing alignments and obtain the final discovered alignments D_s and D_a .

The DTD is generated by method Algorithm 1, where λ_s and λ_a are constants used to strengthen the information discovered by the other model.

Algorithm 1. DTD Generation method

```

1: procedure DTD_GENERATION( $D_s, D_a$ )
2:    $Pure_s \leftarrow D_s - D_s \cap D_a$ ,  $Pure_a \leftarrow D_a - D_s \cap D_a$ 
3:    $DTD_a \leftarrow D_a + \lambda_s \times Pure_s$ ,  $DTD_s \leftarrow D_s + \lambda_a \times Pure_a$ 
4:   return  $DTD_s, DTD_a$ 
5: end procedure
```

3.6 Alignment Prediction

For alignment prediction, the distance of entities is computed as:

$$D(e_1, e_2) = \beta \frac{d(\mathbf{e}_1^s, \mathbf{e}_2^s)}{d_s} + (1 - \beta) \frac{d(\mathbf{e}_1^a, \mathbf{e}_2^a)}{d_a} \quad (4)$$

where $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$; \mathbf{e}^a and \mathbf{e}^b denote the attribute embedding and structure embedding of an entity; d_s and d_a are dimensions of two entities embeddings; β is a hyper-parameter to balance the importance between attribute component and structure component.

3.7 Training

For training, we employ margin-based scoring functions as the training objectives:

$$L_a = \sum_{(e_1, e_2) \in P} \sum_{(e'_1, e'_2) \in P'} [\gamma_a + d(\mathbf{e}_1^a, \mathbf{e}_2^a) - d(\mathbf{e}'_1^a, \mathbf{e}'_2^a)]_+ \quad (5)$$

$$L_s = \sum_{(e_1, e_2) \in P} \sum_{(e'_1, e'_2) \in P'} [\gamma_b + d(\mathbf{e}_1^s, \mathbf{e}_2^s) - d(\mathbf{e}'_1^s, \mathbf{e}'_2^s)]_+ \quad (6)$$

where $[x]_+ = \max\{0, x\}$; P' denotes negative samples; γ_a and γ_b are margin hyper-parameters.

4 Experiments

We developed our model, called BRCEA, using TensorFlow. Our experiments were conducted on a server with Intel Xeon E5-2620 2.1 GHz CPU, an NVIDIA GeForce GTX 1080 Ti GPU, and 32 GB memory.

4.1 Datasets

We evaluate BRCEA on three large-scale cross-lingual datasets. The datasets are built upon DBpedia [2], a large-scale multi-lingual KG containing rich inter-language links between different language versions, containing English, Chinese, Japanese and French versions. Each dataset contains two KGs in different languages and 15 thousand equivalent entity pairs. The statistical details are listed in Table 1. We regard the inner-language links as the gold standard of entity alignment for model training and testing.

Table 1. Details of the datasets

Datasets		Entities	Relations	Attributes	Rel.triples	Attr.triples
<i>DBP15K_{ZH-EN}</i>	Chinese	66,469	2,830	8,113	153,929	379,684
	English	98,125	2,317	7,173	237,674	567,755
<i>DBP15K_{JA-EN}</i>	Japanese	65,744	2,043	5,882	164,373	354,619
	English	95,680	2,096	6,066	233,319	497,230
<i>DBP15K_{FR-EN}</i>	French	66,858	1,379	4,547	192,197	528,665
	English	105,889	2,209	6,422	278,590	576,543

4.2 Experiment Settings

For comparison, we choose 4 state-of-art embedding-based approaches discussed in Sect. 2: JAPE [2], GCN-Align [7], RDGCN [11] and BootEA [13], where the RDGCN achieves the best performance on DBP15K. To evaluate different components of our model, we also provide two implementation variant of BRCEA for ablation studies, including (1) im-GCN-Align-a: an improved approach GCN-Align without structure embedding; (1) BRCEA-s: an approach without attribute component; (2) BRCEA-a: an approach without structure component; (3) RCEA: an approach without bootstrapping component.

The hyper-parameters of BRCEA were as below: $\lambda_a, \lambda_s = 1, 0$; $\gamma_a, \gamma_s = 1.0, 3.0$; $\epsilon_a, \epsilon_s = 0.98, 0.99$ $d_s = d_a = 300$; $th_a, th_s = 0.9, 0.3$; $topK_a = topK_s = 40$; $\beta = 0.5, 0.6, 0.9$ when modeling zh-en, ja-en, fr-en datasets.

For convenience, we use Hits@k as our metrics that are scores measuring the percentage of correctly aligned entities ranked at top k.

5 Results

5.1 Main Results

We randomly choose 30% alignments as base training data and 70% alignments for evaluating cross-lingual entity alignments performance. Table 2 shows the cross-lingual entity alignment results of BRCEA models comparing to the other embedding-based models and ablations on DBP15K. Note that numbers in bold indicate the best performance.

Table 2. Entity alignment results on DBP15k

Models	<i>ZH – EN</i>		<i>JA – EN</i>		<i>FR – EN</i>	
	Hits@1	Hits@10	Hits@1	Hits@10	Hits@1	Hits@10
JAPE	41.18	74.46	36.25	68.50	32.39	66.68
GCN-Align-a	11.79	36.42	6.76	24.99	4.13	18.41
GCN-Align	41.25	74.38	39.91	74.46	37.29	74.49
RDGCN	70.75	84.55	76.74	89.54	88.64	95.72
BootEA	62.94	84.72	62.23	85.39	65.30	87.44
im-GCN-Align-a	26.58	54.90	15.91	41.91	7.16	27.14
BRCEA-s	71.28	85.35	77.91	90.71	88.69	95.99
BRCEA-a	29.94	57.54	23.40	45.36	14.53	36.09
RCEA	71.23	84.76	76.65	89.87	88.12	95.24
BRCEA	73.58	88.21	78.76	91.57	89.01	96.25

We can see that GCN-based embedding methods outperform Translation-based methods for it can learn more structure information by conducting weight propagation from graphs. The performance of our model is 2.83 points higher than RDGCN in DBpedia_{*ZH – EN*} dataset, 2.02 points higher in DBpedia_{*JA – EN*} dataset, and 0.35 points higher in DBpedia_{*FR – EN*}, because we take advantage of attribute information and adopt the training method of semi-supervised learning.

5.2 Ablation Studies

GCN-Align-a vs. im-GCN-Align-a. As we can see in Table 2, im-GCN-Align-a considerably improves GCN-Align-a in all datasets, e.g. over 14.79% improvement of Hits@1 at DBpedia_{*ZH – EN*}, because we enhance the importance of the special relationship, which makes the weight matrix more reasonable.

im-GCN-Align-a vs. BRCEA-a. Comparing im-GCN-Align-a and BRCEA-a, BRCEA-a outperforms im-GCN-Align-s in all datasets, resulting in a 5.97% increase on Hits@1 on DBpedia_{*ZH – EN*}. As discussed in Sect. 3, bootstrapping component generates new alignments as training data for attribute component, which helps attribute component train better.

BRCEA vs. BRCEA-s. Comparing the results of BRCEA and BRCEA-s, we can see that taking attribute information into consideration helps the entity alignment task. However, since DBpedia_{*ZH – EN*} dataset contains more attribute information, BRCEA-s has the best improvement effect on dataset DBpedia_{*ZH – EN*}, up to 2 points. While on DBpedia_{*FR – EN*} dataset, the improvement is slight, less than 1 point.

BRCEA vs. RCEA. Comparing the results of BRCEA and RCEA, bootstrapping component also improves the performance by labeling test data as training data.

5.3 Sensitivity to Proportion of Prior Alignments

To evaluate BRCEA’s sensitivity to the proportion of base training data, we vary the proportion in 10%, 20%, 30%, and 40%, and the results of RDGCN and BRCEA on three real-world datasets are illustrated in Fig. 2. As we can see, when only using 10% pre-alignments as base training data, our model still achieved promising results. To be noticed, the largest gap of model performance appears when the proportion is 10% and the dataset is DBpedia_{ZH-EN}, as much as 4.81%. According to the three slopes in Fig. 2, our model, BRCEA, is less sensitive to the proportion of base training data. Our model has a distinct advantage over RDGCN with limited pre-alignments.

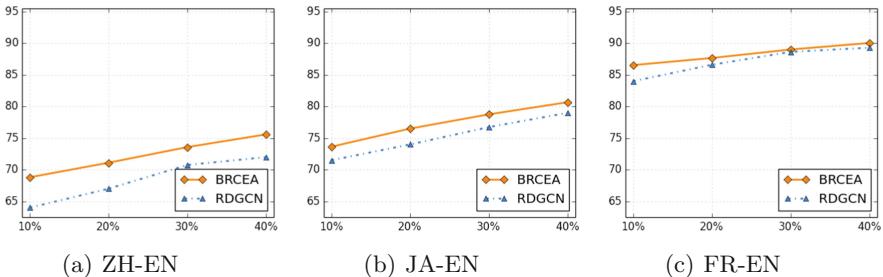


Fig. 2. *Hits@k* w.r.t. proportion of prior alignments. The x-axes are the proportions of pre-alignments, and the y-axes are Hit@1 scores.

6 Conclusion and Future Work

This paper introduces a novel Bootstrapping Relation-aware Entity Alignment over cross-lingual KGs. Our approach is designed to embed entities based on both structure and attribute triples. We evaluate our model on three real-world datasets, and results demonstrated that BRCEA outperforms the state-of-art methods, especially with less training data.

In future work, we look forward to improving our model in several aspects. First, the attribute component can be improved by adding attribute values to model self-characteristics of entities. Second, for entity alignment, we are going to employ a bivariate regression model to learn the respective weight of similarities measuring from the two aspects for a result combination.

Acknowledgments. This work is supported by Beijing Municipal Commission of Education (Co-constructing Program).

References

1. Zou, X.: A survey on application of knowledge graph. *J. Phys.: Conf. Ser.* **1487**, 012016 (2020)
2. Sun, Z., Hu, W., Li, C.: Cross-lingual entity alignment via joint attribute-preserving embedding. In: d'Amato, C., et al. (eds.) ISWC 2017. LNCS, vol. 10587, pp. 628–644. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68288-4_37
3. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Neural Information Processing Systems (NIPS), pp. 1–9 (2013)
4. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28 (2014)
5. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29 (2015)
6. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (vol. 1: Long papers), pp. 687–696 (2015)
7. Wang, Z., Lv, Q., Lan, X., Zhang, Y.: Cross-lingual knowledge graph alignment via graph convolutional networks. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 349–357 (2018)
8. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint [arXiv:1609.02907](https://arxiv.org/abs/1609.02907) (2016)
9. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. arXiv preprint [arXiv:1606.09375](https://arxiv.org/abs/1606.09375) (2016)
10. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
11. Wu, Y., Liu, X., Feng, Y., Wang, Z., Yan, R., Zhao, D.: Relation-aware entity alignment for heterogeneous knowledge graphs. arXiv preprint [arXiv:1908.08210](https://arxiv.org/abs/1908.08210) (2019)
12. Monti, F., Shchur, O., Bojchevski, A., Litany, O., Günnemann, S., Bronstein, M.M.: Dual-primal graph convolutional networks. arXiv preprint [arXiv:1806.00770](https://arxiv.org/abs/1806.00770) (2018)
13. Sun, Z., Wei, H., Zhang, Q., Qu, Y.: Bootstrapping entity alignment with knowledge graph embedding. In: IJCAI 2018, pp. 4396–4402 (2018)
14. Zhang, S., Tong, H., Xu, J., Maciejewski, R.: Graph convolutional networks: a comprehensive review. *Comput. Soc. Netw.* **6**(1), 1–23 (2019)



Employing Multi-granularity Features to Extract Entity Relation in Dialogue

Qiqi Wang and Peifeng Li^(✉)

School of Computer Science and Technology, Soochow University, Suzhou, China
20195227016@stu.suda.edu.cn, pfli@suda.edu.cn

Abstract. Extracting relational triples from unstructured text is essential for the construction of large-scale knowledge graphs, QA and other downstream tasks. The purpose of dialogue relation extraction is to extract the relations between entities from the multi-person dialogue texts. The existing dialogue relation extraction models only focused on coarse-grained global information and ignored fine-grained local information. In this paper, we propose a dialogue relation extraction model BERT-MG to capture the features on different granularity at different BERT layers to take advantage of the fine-grained dialogue features. Moreover, we design a type-confidence mechanism to use the entity type information to assist relation inference. Experimental results on the DialogRE dataset prove that our proposed model BERT-MG outperforms the SOTA baselines.

Keywords: Dialogue relation extraction · Multi-granularity feature · Type-confidence

1 Introduction

Relation extraction is an important subtask of information extraction. Its main goal is to infer the relationship between specified entity pairs based on the text content. Most of the current researches on relation extraction are based on written text [10]. With the increasing number of open dialogue data, relation extraction based on dialogue text has attracted more and more attention.

Figure 1 shows an example from the DialogRE [13] dataset, in which the text above the dividing line is a multi-round dialogue. There are two categories of entities in the dialogue text: each speaker is an entity mention (e.g., Speaker1), and then there are some entity mentions (e.g., Monica) in the utterances. Part of the entity relations existing in this text is given below in Fig. 1.

Compared with the relation extraction in written text, the task in dialogue text has the following challenges. First, the dialogue text has weaker textual consistency. Since the dialogue text is composed of multiple rounds, the speakers of different rounds are different too, which will reduce the consistency of the text between different rounds and make the model more difficult to understand the meaning of the entire text. Second, the dialogue text is more colloquial. Take the

Speaker 1: Forget it Joey, I'm with Bob now.

Speaker 2: Bob? Who the hell's Bob?

Speaker 1: Bob is great. He's smart, he's sophisticated, and he has a real job. You, you go on three auditions a month and you call yourself an **actor**, but Bob...

Speaker 2: Come on, we were great together. And not just at the fun stuff, but like, talking too.

Speaker 1: Yeah, well, sorry, Joe. You said let's just be friends, so guess what?

Speaker 2: What?

Speaker 1: We're just friends.

Speaker 2: Fine, fine, so, why don't the four of us go out and have dinner together tonight? You know, as friends?

Speaker 1: What four of us?

Speaker 2: You know, you and Bob, and me and **my girlfriend**, uh, uh, **Monica**.

head entity	relation	tail entity
Speaker1	"per:girl/boyfriend"	Bob
Speaker1	"per:friends"	Speaker2
Speaker2	"per:title"	actor
my girlfriend	"per:alternate_names"	Monica
Speaker1	"unanswerable"	actor

Fig. 1. An example from the DialogRE dataset.

last round of the dialogue in Fig. 1 as an example, there are two ‘uh’ between ‘my girlfriend’ and ‘Monica’, which interfere with the model’s judgment of their relation.

The current dialogue relation extraction models, including the state-of-the-art models (e.g., AdaPrompt-tuning [1] and GDPNet [12]), focus on extracting coarse-grained semantic features at the document level to improve the model performance by promoting the circulation and integration of information in the full text. However, they ignored the fine-grained features at phrase level. Actually, the fine-grained features (e.g., trigger words and sentence pattern) can play an important role in solving the challenge of dialogue relation extraction. First, these features can provide conducive information to relation classification at phrase level. For example, the sentence pattern ‘I’m with sb. now.’ provides the positive evidence for the existence of the relation ‘per:girl/boyfriend’. These trigger words or sentence patterns have a strong correlation with the specific relation rather than the speaker, so that they can be used as a strong consistency feature to correct the errors in using coarse-grained features. Second, many studies have showed the differences between the different BERT layers [3]. For example, BERT-FiT [8] proved that each BERT layer can capture the different features of input texts. Jawahar et al. [4] pointed out that BERT mainly captured the semantic features in higher layers and mainly captured the phrase-level features in lower layers. The token embeddings at the lower layers often contain more original meanings of words. Therefore, the fine-grained features at the lower layer contain more original word meanings, which can reduce the interference of some meaningless colloquial words.

To address the issue of ignoring the fine-grained features, we propose a dialogue relation extraction model BERT-MG (BERT with Multi-Granularity) to capture the features on multi-granularity. Specially, we first extract the coarse-grained features in the last layer of BERT, and we then use LSTM to aggregate the round-level classification features on the basis of BERTs [13]. Secondly, we introduce TextCNN [15] to aggregate the phrase-level fine-grained features in

the first layer of BERT, and then we design a local mask mechanism to select the attention window of TextCNN under different entity categories. Finally, we aggregate the coarse-grained features and fine-grained features to obtain the probabilities of relation classification. Meanwhile, we apply a type-confidence mechanism to make the classification probabilities more reasonable on the entity type level. Experimental results on the DialogRE dataset prove that our proposed model BERT-MG outperforms the SOTA baselines.

2 Related Work

Entity Relation Extraction in Written Texts. There are a large number of studies on such field and it is mainly divided into two categories: intra-sentence and document-level relation extraction. The goal of the former is to infer the relation between two entities in a single sentence. For example, Multi-turn QA [7] converted the task into multiple turns of question and answer to use the prior information of the question; CASREL [10] treated the relation as a mapping from subject to object. The latter extends the scope of relation extraction to long text. For example, EoG [2] proposed a document-level graph with an iterative algorithm over the graph edges to model intra- and inter-sentence pairs simultaneously; GEDA [6] proposed a graph-enhanced dual attention mechanism; GAIN [14] utilized a heterogeneous mention-level graph to capture the document-aware features and used an entity-level graph to optimize the inference results.

Entity Relation Extraction in Dialogue. There are only a few studies on dialogue relation extraction. DialogRE [13] annotated dialogue relation extraction corpus and proposed a new dialogue-based task for entity relation extraction. BERTs [13] replaced the speakers in the entity pair with special tags for better relation recognition. SimpleRE [11] proposed a model to take into account both the speed and performance. GDPNet [12] constructed a multi-view graph to judge the relation from different perspectives, and then used DTWPool to refine the graph. AdaPrompt-tuning [1] proposed an effective method to fine-tune the language model for the dialogue relation extraction task.

3 Methodology

3.1 Problem Definition

Let $U = \{u_1, u_2 \dots u_m\}$ be the dialogue text sequence, where u_i represents the dialogue text of the i -th round and m is the total number of dialogue rounds. Each round is composed of the speaker and his utterance. The entity categories include the speaker and the entity mention contained in the speaker’s utterance. Each entity has an entity type. Given the dialogue text U , an entity pair in the dialogue text denoted as (E_1, E_2) , and the entity types of the entity pair denoted as (E_1^{type}, E_2^{type}) , the goal of dialogue relation extraction is to infer the relation r between the entity pair (E_1, E_2) .

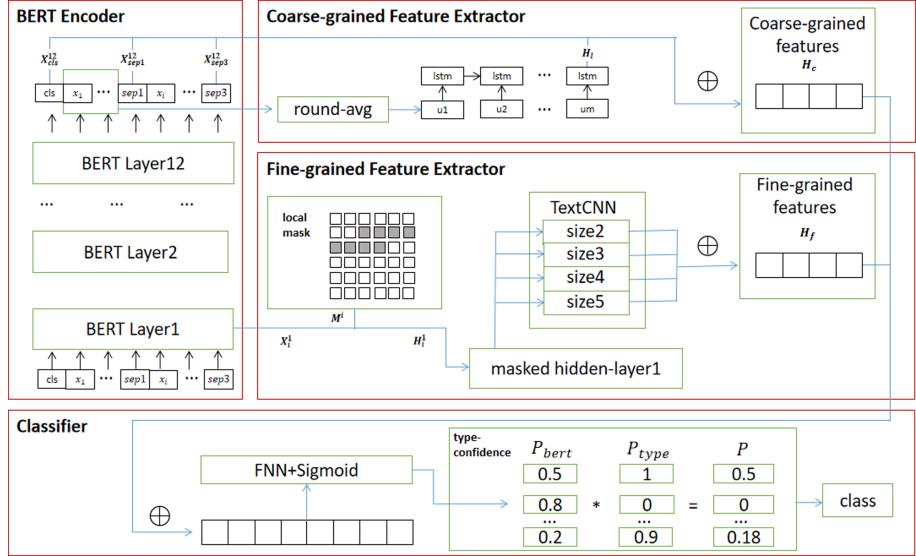


Fig. 2. Architecture of our model BERT-MG.

3.2 Architecture

The overall structure of our model BERT-MG is shown in Fig. 2, which consists of four components: the BERT encoder, Coarse-grained Feature Extractor, Fine-grained Feature Extractor and Classifier. Specially, we use BERT to encode the text sequence in dialogue. As mentioned in Sect. 1, the fine-grained features can provide the phrase-level information and the BERT lower layers are more suitable to represent the fine-grained features, while the coarse-grained features can provide the document-level information and the BERT higher layers are more suitable to represent the coarse-grained features. Hence, we then use the coarse-grained feature extractor and the fine-grained feature extractor to extract the coarse-grained and fine-grained features from the last layer and the first layer of BERT, respectively. Finally, BERT-MG integrates the coarse-grained features with the fine-grained features and uses FNN (Feedforward Neural Network) and sigmoid function to perform multi-label classification. In addition, a type-confidence mechanism is used to optimize classification results using entity type information.

3.3 BERT Encoder

The token sequence “[cls] U [sep1] E1 [sep2] E2 [sep3]” is encoded by BERT firstly. To allow our BERT-MG to do better in recognizing the speaker in the entity pair, we adopt the token input form used by BERT_S. That is, for the dialogue text U and the entity pair (E_1, E_2) , if E_1 (or E_2) is a speaker, we replace all E_1 (or E_2) in the token sequence with [unused1] (or [unused2]), where [unused1]

and [unused2] are two special marks in BERT. If the entity is an entity mention (not a speaker), we keep the original entity text unchanged. The output of the last layer is used as the input of the coarse-grained feature extractor, and we take the output of the first layer as the input of the fine-grained feature extractor.

3.4 Coarse-Grained Feature Extractor

The coarse-grained features are divided into two parts. In the first part, since Kovaleva et al. [5] shows that [cls] and [sep] in BERT have large self-attention weights in the output of the last layer, our BERT-MG uses the conventional classification feature [cls] and the separators [sep1] and [sep3] as the coarse-grained features of the full text. In the second part, since the dialogue text is composed of multiple rounds, the content of each round is relatively independent. To aggregate information from the round level, our BERT-MG averages the representation of each round, and marks the average representation obtained as \mathbf{H}_i^{12} , where i is the corresponding round and 12 is the number of the last layer of the BERT. In order to model the order of the rounds, LSTM is used to aggregate the round-level coarse-grained features, denoted as \mathbf{H}_l . The coarse-grained features of the model are obtained by splicing these two parts, denoted as \mathbf{H}_c . The formula for obtaining \mathbf{H}_l and \mathbf{H}_c is as follows:

$$\mathbf{H}_l = \text{LSTM}(\mathbf{H}_1^{12}, \mathbf{H}_2^{12}, \dots, \mathbf{H}_m^{12}) \quad (1)$$

$$\mathbf{H}_c = \mathbf{X}_{cls}^{12} \oplus \mathbf{X}_{sep1}^{12} \oplus \mathbf{X}_{sep3}^{12} \oplus \mathbf{H}_l \quad (2)$$

where \mathbf{H}_1^{12} to \mathbf{H}_m^{12} are the average representations from round 1 to round m , and \mathbf{X}_{cls}^{12} , \mathbf{X}_{sep1}^{12} and \mathbf{X}_{sep3}^{12} are the representations of [cls], [sep1] and [sep3]. \oplus is the concatenation operation.

3.5 Fine-Grained Feature Extractor

Although the dialogue text is long, only parts of the rounds in the text may be related to a specific entity-pair relation. If we extract the fine-grained features on the entire dialogue text, many interference features will be introduced to our model. To filter out the interference features, we use a local mask mechanism to select the fine-grained window. The local mask mechanism aims to delete rounds with low probability of having information that is conducive to relation classification based on the characteristics of the dialogue information distribution. According to the different entity categories, the local mask mechanism is divided into three modes: speaker-speaker pair, speaker-entity pair, and entity-entity pair. In this paper, mask means to retain information.

Speaker-Speaker Pair. As showed in Fig. 1, if the two entities are both speakers, such as inferring the relation between Speaker1 and Speaker2, information related to the relation between the two speakers is often distributed in the full text. Hence, in this case, we mask the full text as follows:

$$M_{s-s}^i = \begin{cases} 1, & 1 \leq i \leq L \\ 0, & L < i \leq 512 \end{cases} \quad (3)$$

where i is the position number of the token. L is the sequence length from $[cls]$ to $[sep3]$ and it represents the length of the valid part of the token sequence.

Speaker-Entity Pair. If one entity is a speaker and the other entity is an entity mention, we need to find the speaker’s attention window for the entity mention. In the dialogue text, when two adjacent utterances of the speaker surround the entity mention, the dialogue round between the two utterances often contains information that is conducive to relation inference. For example, as shown in Fig. 1, round 2 and round 4 are the two adjacent utterances of the ‘speaker2’ surround the ‘actor’. We can infer from the rounds between the 2nd and 4th rounds that the relation between ‘Speaker2’ and ‘actor’ is ‘per:title’. Therefore, the local mask mechanism should mask these intermediate rounds as follows:

$$head = \begin{cases} \max(U_{(1, u_{em}-1)} \cap U_s), U_{(1, u_{em}-1)} \cap U_s \neq \emptyset \\ u_{em}, U_{(1, u_{em}-1)} \cap U_s = \emptyset \end{cases} \quad (4)$$

$$tail = \begin{cases} \min(U_{(u_{em}+1, m)} \cap U_s), U_{(u_{em}+1, m)} \cap U_s \neq \emptyset \\ u_{em}, U_{(u_{em}+1, m)} \cap U_s = \emptyset \end{cases} \quad (5)$$

$$M_{s-em}^i = \begin{cases} 1, & head \leq u_i \leq tail \\ 0, & u_i < head \text{ or } u_i > tail \end{cases} \quad (6)$$

where u_{em} is the round of the entity mention which is closest to the speaker’s utterance. U_s is the set of rounds of the speaker’s utterances. $U_{(a,b)}$ is the set of rounds between the round a and round b . u_i is the round of the i -th token. m is the total number of rounds of the dialogue text. $\max(U)$ and $\min(U)$ represent the function of taking the largest and smallest elements in the set U , respectively.

Entity-Entity Pair. If both entities are entity mentions, such as inferring the relation between ‘my girlfriend’ and ‘Monica’ in Fig. 1, we mask the rounds between the farthest rounds of the two entities as follows:

$$head = \min(U_1^{em} \cup U_2^{em}) \quad (7)$$

$$tail = \max(U_1^{em} \cup U_2^{em}) \quad (8)$$

$$M_{em-em}^i = \begin{cases} 1, & head \leq u_i \leq tail \\ 0, & u_i < head \text{ or } u_i > tail \end{cases} \quad (9)$$

where U_1^{em} and U_2^{em} represent the set of rounds containing E_1 and E_2 , respectively.

We use TextCNN to capture the fine-grained features in the first layer of BERT. TextCNN can capture the n-gram grammatical features, which are the useful sentence pattern features described in Sect. 1. The TextCNN in our model uses four types of the windows in different sizes, each with multiple channels, to obtain different views. After the maximum pooling of each channel, it integrates

the features in different windows as the fine-grained features, denoted as \mathbf{H}_f as follows:

$$\mathbf{H}_i^1 = M^i \mathbf{X}_i^1 \quad (10)$$

$$\mathbf{c}_h^i = f(\mathbf{W} \mathbf{H}_{i:i+h-1}^1 + \mathbf{b}) \quad (11)$$

$$\mathbf{c}_h = \text{MaxPooling}([\mathbf{c}_h^1, \mathbf{c}_h^2, \dots, \mathbf{c}_h^{L-h+1}]) \quad (12)$$

$$\mathbf{H}_f = \mathbf{c}_1 \oplus \mathbf{c}_2 \oplus \dots \oplus \mathbf{c}_N \quad (13)$$

where $\mathbf{X}_i^1 \in \mathbb{R}^d$ is the representation of the i-th token in the first layer of BERT. $M^i \in \{M_{s-s}^i, M_{s-em}^i, M_{em-em}^i\}$ is the local mask. h is the size of the window. $f()$ denotes the *ReLU* activation function. $\mathbf{W} \in \mathbb{R}^{k \times d_h}$ is a trainable parameter where k is the number of channels and $d_h = d \times h$. $\mathbf{H}_{i:i+h-1}^1 \in \mathbb{R}^{d_h}$ is the splicing embedding from H_i^1 to H_{i+h-1}^1 . $\mathbf{b} \in \mathbb{R}^k$ is a bias vector. *MaxPooling()* is the max-pooling function which can select the largest element of each channel. L is the number of the tokens which is 512 in BERT. N is the number of windows.

3.6 Classifier

After concatenating the coarse-grained and fine-grained features, the probability value of each relation can be obtained through the Feedforward Neural Network (FNN) and the sigmoid function as follows:

$$\mathbf{P}_{bert} = \text{sigmoid}(FNN(\mathbf{H}_c \oplus \mathbf{H}_f)) \quad (14)$$

where the composition form of \mathbf{P}_{bert} is $\mathbf{P}_{bert} = [P_1, P_2, \dots, P_{rn}]$, and rn is the number of relation categories.

Many relations have certain restrictions on the entity type of the head and tail entities. For example, the ‘per:girl/boyfriend’ relation restricts the head and tail entities must be of the ‘PER’ type, so the entity type information can be used to improve the accuracy of the relation classification. Inspired by knowledge graph representation learning [9], we design a type-confidence mechanism, which can use the entity type information to assist relation inference.

Each pair of entities and the correct relation between them can form a positive type triple, denoted as (h, r, t) where h and t are the head entity and the tail entity, respectively, and r is their relation category. We assign randomly initialized embedding to each entity type and relation. The embedding of the relation is divided into two parts: the relation for head entity and the relation for tail entity. We design the following scheme to calculate the triplet score and type-confidence:

$$\text{score}(h, r, t) = \mathbf{r}_h^T \mathbf{h} + \mathbf{r}_t^T \mathbf{t} \quad (15)$$

$$\text{confidence} = \text{sigmoid}(\text{score}) \quad (16)$$

where r_h , h , r_t , t respectively represent the embedding of the relation for head entity, head entity type, the relation for tail entity, and tail entity type.

We randomly replace the r in each positive type triple (h, r, t) with a relation r' that does not exist between the entity pairs, thereby generating a negative type triple (h, r', t) . The following loss function is used to train the type embedding and relation embedding.

$$L_t = -\log\sigma(score(h, r, t) - \gamma) - \log\sigma(-score(h, r', t) - \gamma) \quad (17)$$

where σ is the sigmoid function. γ is a hyperparameter.

For each pair of pre-classified entities, we calculate the confidence of all relations. Then integrate the confidence and multiply it with P_{bert} to get the final probability of each relation:

$$P_{type} = [confidence(h, r^1, t), \dots, confidence(h, r^n, t)] \quad (18)$$

$$P = P_{bert} * P_{type} \quad (19)$$

We use the following loss function to optimize the model:

$$L = -\frac{1}{n} \sum_{i \in N} [t^i * \ln p^i + (1 - t^i) * \ln(1 - p^i)] \quad (20)$$

where N is all relation categories, and n is the total number of relation categories. t^i is a binary value. When the relation i is a correct relation, t^i is equal to 1, otherwise it is equal to 0. p^i is the probability of the relation i .

4 Experimentation

4.1 Experimental Settings

We evaluate our model on DialogRE [13], the first human-annotated dialogue-based relation extraction dataset. This dataset defines 37 entity relations, including 36 relation categories and an ‘unanswerable’ relation that indicates that there is no relation between entity pairs. The dataset consists of 1788 dialogue texts, with a total of 10168 entity-relation triples and is divided into training set, validation (development) set, and test set according to the ratio of 60%, 20%, and 20%.

Same as DialogRE, our BERT-MG uses F1 and $F1_c$ [2] as evaluation metrics. F1 is used to evaluate the model’s ability to infer entity relations when the entire dialogue text is known. $F1_c$ is used to evaluate the performance of the model when only part of the dialogue text is known.

We use PyTorch framework and conduct our experiments with 1 NVIDIA TITAN Xp GPU. We set a learning rate of 3e-5 for BERT, and a learning rate of 1e-4 for LSTM, TextCNN, and type embedding. We use Adam as the optimization function. The hidden layer dimension of LSTM is set to 500. The window size of TextCNN is set to [2–5], and the number of channels in each window is set to 100. γ in L_t is set to 15. The dimension of the embedding of entity type and relation is 500.

4.2 Experimental Results

We use six models as baseline models, including the three basic models in DialogRE: LSTM [13], BERT [13], BERT_S [13] and the three state-of-the-art models GDPNet [12], SimpleRE [11] and AdaPrompt-tuning [1]. The experimental results are showed in Table 1 on both the development and test set. BERT_S has the best performance among the three basic models in [13]. The text input of BERT-MG is the same as that of BERT_S. Compared with BERT_S, BERT-MG has improved the F1 and F1_c scores by 6.3 and 5.5 on the development set, and 7.9 and 6.6 on the test set, respectively. This shows that BERT-MG can effectively extract both the coarse-grained features and fine-grained features and then improve the performance of dialogue relation extraction.

Table 1. Performance comparison of six baselines and our BERT-MG on DialogRE (- indicates that those baselines did not report the performance on the corresponding metric).

Model	Dev		Test	
	F1	F1 _c	F1	F1 _c
LSTM	46.7	44.2	47.4	44.9
BERT	60.6	55.4	58.5	53.2
BERT _S	63.0	57.3	61.2	55.4
AdaPrompt-tuning	67.0	-	65.8	-
SimpleRE	-	-	66.3	-
GDPNet	67.1	61.5	64.9	60.1
Our Model	69.3	62.8	69.1	62

Among all the baselines, SimpleRE achieves the highest F1 score of 66.3 on the test set, and our model BERT-MG improves it by 2.8. GDPNet achieves the highest F1 value of 67.1 and F1_c scores of 61.5 on the validation set, and the highest F1_c score of 60.1 on the test set. Compared with GDPNet, BERT-MG improve them by 2.4, 1.3, and 1.9, respectively. We can find that whether it is the method of modifying the BERT input (SimpleRE), the method based on the graph (GDPNet) or the method of fine-tuning language models (AdaPrompt-tuning), all of them are based on coarse-grained features. Our model BERT-MG surpasses all the state-of-the-art models that based on the coarse-grained features and this indicates that the fine-grained features allow the model to obtain more valuable classification information from the local area, thereby making the classification more accurate.

Figure 3 shows an example of the relation inference error in BERT_S. For the entity pair (Speaker6, Scott), intuitively, through the text ‘Speaker6: I’m Scott.’ of the third-to-last round, the model should easily infer that the relation between this entity pair is ‘per:alternate_names’. However, too many speakers

Speaker 1: Dr. Geller, there's a seat over here.

... ...

Speaker 6: I'm Scott.

Speaker 2: Yeah, okay, Scott!

Speaker 6: And I need to flip the light switch on and off 17 times before I leave a room or my family will die.

head entity	relation	tail entity
Speaker6	"per:alternate_names"	Scott

Fig. 3. An example of the relation inference error of the BERT_S model.

and dialogue rounds lead to poor text consistency. Personal pronouns in the dialogue interfere with the model's judgment of the speaker. Finally, the relation that BERT_S infers is 'unanswerable', that is, there is no relation between the entity pairs. However, BERT-MG can obtain the third-to-last round through the local mask, and then use TextCNN to capture sentence-level features. The sentence pattern level feature in this example is 'I'm Scott.', which provides positive information for the relation 'per:alternate_names'. Finally, BERT-MG can successfully infer the relation between the entity pair.

4.3 Ablation Study

We conduct a series of ablation studies to evaluate the value of different components and the results are showed in Table 2. From Table 2, we can find that the performance of BERT-MG drops by 1.1–2.1 after removing the LSTM (w/o LSTM). The reason is that LSTM can sequentially integrate the information between dialogue rounds, which is helpful for the model to understand the overall dialogue semantics. If BERT-MG removes the fine-grained features (w/o Fine-grained Features), the F1 and F1_c scores on the validation set drop by 2.3 and 1.4, and those on the test set drop by 3.9 and 2.7, respectively. This indicates that the fine-grained features captured by TextCNN can capture phrase-level features and are effective in dialogue relation extraction. In addition, we conduct a study that does not use local masks (w/o Local Mask), that is, allows TextCNN to extract fine-grained features in the full text. The results show that it will reduce the F1 scores by 1.6 and 1.2 on the validation set and test set, respectively. This shows that the local mask mechanism we designed can simulate the attention window between entity pairs, and the performance of the fine-grained feature extractor can be effectively improved after the interference information is filtered out through the attention window.

Table 2. Performance comparison of BERT-MG and its variants.

Model	Dev		Test	
	F1	F1 _c	F1	F1 _c
BERT-MG	69.3	62.8	69.1	62.0
w/o LSTM	-1.6	-2.1	-1.7	-1.1
w/o Fine-grained Features	-2.3	-1.4	-3.9	-2.7
w/o Local Mask	-1.6	-0.7	-1.2	-0.7

Table 3 shows the performance of the type-confidence mechanism. From Table 3, we can find that the type-confidence mechanism can improve the precision (P) by 0.9 and 1.1 on the validation set, and 0.5 and 1.1 on the test set. This proves that the type-confidence mechanism can correct unreasonable relations extracted from the model at the entity type level.

Table 3. Performance (Precision) of type-confidence mechanism. The recall rate of the model remains unchanged, so it is not listed in the table.

Model	Dev		Test	
	P(F1)	P(F1 _c)	P(F1)	P(F1 _c)
BERT-MG	69.7	71.3	69.5	71.4
w/o type-confidence	-0.9	-1.1	-0.5	-1.1

To explore the differences between the BERT layers, we conduct two experiments: 1) feeding the first layer of BERT to the LSTM of the coarse-grained feature extractor (CF-first); 2) feeding the last layer of BERT to the fine-grained feature extractor (FF-last). Table 4 shows the results on BERT-MG and its variants. It can be seen from the results that the performances drop by 0.7-1.5 after the coarse-grained feature extractor is applied to the first layer, and those drop by 0.8-1.9 after the fine-grained feature extractor is applied to the last layer. In addition, through experiments, we find that the performance of fine-grained feature extraction in the first three layers of Bert is similar, and the performance begins to decline from the fourth layer. This proves that the lower layer of BERT contains more phrase-level features and is more suitable to represent the fine-grained features, and the higher layer of BERT has more semantic features and is more suitable to represent coarse-grained features. Hence, it is reasonable and effective to use different modules to capture different granular features.

Table 4. Performance comparison of BERT-MG, CF-first and FF-last.

Model	Dev		Test	
	F1	F1 _c	F1	F1 _c
BERT-MG	69.3	62.8	69.1	62.0
CF-first	-1.1	-0.9	-1.5	-0.7
FF-last	-1.7	-1.1	-1.9	-0.8

5 Conclusion

In this paper, we propose a dialogue relation extraction model BERT-MG which can integrate multi-granularity features in dialogue. Different from those dialogue relation extraction models based on coarse-grained features, BERT-MG extracts the features of different granularity in different layers of BERT, so that the model can obtain phrase-level information. In addition, we propose a type-confidence mechanism that uses entity type information to further optimize the results. Experimental results on the DialogRE dataset show that our BERT-MG outperforms the strong baselines significantly. In the future, we will focus on how to infer the relation between entity pairs using only part of the dialogue.

Acknowledgments. The authors would like to thank the three anonymous reviewers for their comments on this paper. This research was supported by the National Natural Science Foundation of China (No. 61836007, 61772354 and 61773276), and the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).

References

- Chen, X., et al.: AdaPrompt: adaptive prompt-based finetuning for relation extraction. CoRR abs/2104.07650 (2021)
- Christopoulou, F., Miwa, M., Ananiadou, S.: Connecting the dots: document-level neural relation extraction with edge-oriented graphs. In: EMNLP-IJCNLP, pp. 4924–4935 (2019)
- Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT, pp. 4171–4186 (2019)
- Jawahar, G., Sagot, B., Seddah, D.: What does BERT learn about the structure of language? In: ACL, pp. 3651–3657 (2019)
- Kovaleva, O., Romanov, A., Rogers, A., Rumshisky, A.: Revealing the dark secrets of BERT. In: EMNLP-IJCNLP, pp. 4364–4373 (2019)
- Li, B., Ye, W., Sheng, Z., Xie, R., Xi, X., Zhang, S.: Graph enhanced dual attention network for document-level relation extraction. In: COLING, pp. 1551–1560 (2020)
- Li, X., et al.: Entity-relation extraction as multi-turn question answering. In: ACL, pp. 1340–1350 (2019)
- Sun, C., Qiu, X., Xu, Y., Huang, X.: How to fine-tune BERT for text classification? In: CCL, pp. 194–206 (2019)

9. Sun, Z., Deng, Z., Nie, J., Tang, J.: RotatE: knowledge graph embedding by relational rotation in complex space. In: ICLR (2019)
10. Wei, Z., Su, J., Wang, Y., Tian, Y., Chang, Y.: A novel cascade binary tagging framework for relational triple extraction. In: ACL, pp. 1476–1488 (2020)
11. Xue, F., Sun, A., Zhang, H., Chng, E.S.: An embarrassingly simple model for dialogue relation extraction. CoRR abs/2012.13873 (2020)
12. Xue, F., Sun, A., Zhang, H., Chng, E.S.: GDPNet: refining latent multi-view graph for relation extraction. In: AAAI, pp. 14194–14202 (2021)
13. Yu, D., Sun, K., Cardie, C., Yu, D.: Dialogue-based relation extraction. In: ACL, pp. 4927–4940 (2020)
14. Zeng, S., Xu, R., Chang, B., Li, L.: Double graph based reasoning for document-level relation extraction. In: EMNLP, pp. 1630–1640 (2020)
15. Zhang, Y., Wallace, B.C.: A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In: IJCNLP, pp. 253–263 (2017)



Attention Based Reinforcement Learning with Reward Shaping for Knowledge Graph Reasoning

Sheng Wang¹, Xiaoying Chen^{1,2(✉)}, and Shengwu Xiong¹

¹ Wuhan University of Technology, Wuhan, China

{wsheng, xiaoying.chen, xiongsw}@whut.edu.cn

² Hubei Credit Information Center, Wuhan, China

Abstract. Knowledge graph reasoning aims at solving certain tasks by finding reasoning paths, which has aroused extensive attention. Recently, a solution for path reasoning that combines reinforcement learning has achieved successful progress. But these researches mainly focus on the agent's choice of relation and ignore the importance of entity, which will cause the random selection by the agent if 1-N/N-N relations occur. Thus, we propose a reinforcement learning based path reasoning model, which solves this problem from the topological and semantic levels. First, the attention mechanism is introduced in our model, which can extract the hidden feature from neighbor entities and helps the policy network to make a suitable choice instead of random for the actions with the same relation. Then, we introduce a convolutional neural network into our model to distinguish the rationality of the path by the semantic feature. To mitigate the negative impact of terminal rewards, we use a potential-based reward shaping function, which considers the potential gap between agent states as the reward and without any pre-training. Finally, we compare our model with the state-of-the-art baselines on two benchmark datasets, the results of extensive comparison experiments validate the effectiveness of the proposed method.

Keywords: Knowledge graph reasoning · Reinforcement learning · Attention mechanism · Reward shaping

1 Introduction

Knowledge graphs (KGs) contain a large amount of real-world knowledge in the form of triples, which plays an important role in many applications (e.g. question answering [1] and personalized recommendations [2]). However, the performance of many tasks is limited by the incompleteness and sparseness of the knowledge graph, thus it's meaningful to complete the knowledge graph by mining the existing knowledge triples.

Path-based knowledge graph reasoning is the commonly used reasoning method, which searches many paths from the KG and uses them as features to complete the specific task. Recently, research on applying reinforcement learning (RL) to path reasoning has made great progress, and many studies [3, 4] formulate path reasoning as a Markov

decision process (MDP) [5]. Through the stepwise interaction with the KG environment, the agent will choose a rational action until the reasoning is completed.

Entity selection is an important part of the path reasoning process. Especially in the query answering task, the answer entity is always unknown and must be inferred in the whole reasoning process. Previous researches in RL-based path reasoning consider the agent’s action selection as the relation (e.g., DeepPath) or the relation-entity pair (e.g., MINERVA) selection, which underestimate the importance of entity selection and result in the random selection by the agent when 1-N/N-N relation appear during the reasoning process. For example, to answer the question “What is Obama’s nationality?”, there may be several paths as follows:

$Obama \xrightarrow{\text{born_in}} Hawaii \xrightarrow{\text{locate_in}} U.S \text{ and } Obama \xrightarrow{\text{born_in}} Hawaii \xrightarrow{\text{locate_in}} Pacific_Ocean$.

Only the former path is the correct path, but they have the same answer relation *locate_in*. For this problem, some researchers have proposed a solution that uses two agents in path reasoning, one for relations reasoning and other for entities reasoning [6]. But the training of this solution is complicated and the terminal reward used will lead to sparse rewards.

Therefore, we proposed a novel reinforcement learning based knowledge graph reasoning model, which incorporates the attention mechanism with the potential and path semantic based reward shaping function. More specifically, we solve the entity selection problem from the topological and semantic levels. First, our model will extract the neighbor entity’s hidden feature by the attention mechanism, which helps the agent to choose a sensible action instead of random when the 1-N/N-N relation appears. Second, we introduce a CNN into our model to identify the semantic features of the reasoning path and feedback the agent a semantic reward. To obtain more semantic reward, the agent will adjust the choice of actions to generate more logical path. And the CNN will be adversarial trained through some ground truth paths and random paths to obtain path discrimination ability. Furthermore, in addition to the semantic level rewards, we also introduced the potential-based reward to solve the sparse reward problem.

In general, the major contributions of this paper are summarized as follows:

- (1) We propose a multi-hop path reasoning model based on reinforcement learning combined with an attention mechanism, which is used to solve the entity selection problem from the topological level when 1-N/N-N relations appear.
- (2) We design a novel method to enhance the reward from potential and semantic levels, which helps to solve the entity selection and sparse rewards problem.
- (3) We evaluate the effectiveness of our model on two benchmark datasets through extensive comparative experiments, ablation study, and case study.

2 Related Work

Up to now, there are two branches for the research of knowledge graph reasoning. The first is the embedding based model, which aims to map entities and relations into a low-dimension vector space and learn a score function $f(e_s, r, e_o)$ to judge the truth of triples. Such as TransE [7] and its extensions [8–10] treat the sum of head entity embedding and relation embedding as a translation of tail entity embedding. DistMult [11] represents each relation as a matrix and restricts it to be a diagonal matrix, which reduces the

number of parameters. Furthermore, ComplEx [12] represents the entities and relations as complex vectors, which can handle the facts about antisymmetric relations well. ConvE [13] uses a multi-layer convolutional network for knowledge reasoning, and the embedding of entity relation pairs is regarded as a picture for extracting features with a convolution kernel. But for the query answering task, the embedding based models simply return a tail entity e_o according to the score among all entities. Although it can achieve good performance, it lacks interpretability.

Another branch is the path-based methods. Path-Raking Algorithm [14] is the first path-based reasoning model, which to search paths between entities by DFS and random-walk and use those paths to predict the missing relation between entities. Recently, path reasoning based on the reinforcement learning framework has made great progress. DeepPath [3] is the first one to use RL based model to search paths between entities for the link prediction task in a supervised learning way. MINERVA [4] introduced RL to search answer entities end-to-end for the query answering task. However, above models use the terminal reward, which will lead to the sparse rewards problem and slow down the training efficiency. To solve this problem, Lin et al. [15] use a pre-trained well embedding model (e.g. ComplEx and ConvE) as the knowledge-based reward shaping function. Meanwhile, potential-based reward shaping function is used in SRN [1] without external knowledge. Li et al. present a generative adversarial imitation learning based plug-and-play framework DIVINE [16] for enhancing existing RL-based methods and getting rid of the meticulous reward engineering to fit specific datasets.

Some researchers have also made good progress in combining rules and path reasoning. For instance, RuleGuilder [17] leverages high-quality rules generated by symbolic based methods to provide reward supervision for walk-based agents. GNTPs [18] is an extension to NTPs addressing their complexity and scalability limitations, thus making them applicable to real-world datasets. As for the problem of entity selection mentioned above, Li et al. propose MARLPaR [6], which uses two agents in path reasoning, one for relations reasoning and other for entities reasoning. But the training of two agents is complicated and the terminal reward used will lead to sparse rewards.

3 Approach

In this section, we will elaborate on the detail of our method for query answering task. Firstly, we define the query answering task (Sect. 3.1) and introduce the reinforcement learning framework used by our method (Sect. 3.2). Then we describe the attention-based policy network in our proposed model (Sect. 3.3), which will help to solve the entity selection problem in the query answering task. Finally, we introduce the reward shaping function in our method (Sect. 3.4).

3.1 Task Definition

We formally denote the knowledge graph as $G = (E, R)$ where (E, R) represent entity set and relation set respectively. Each directed edge and its linked entities represent the fact triple in the KG $(e_s, r, e_o) \in G$.

Given an entity e_s and query relation r_q , the query answering task can be formulated as $(e_s, r_q, ?)$, e.g., $(Beijing, capital\ of, ?)$, it means the question “Which country’s capital city is Beijing?”. Our goal is to find the answer entities $\{e_{answer}\}$ by traversing the KG, note that the triple $(e_s, r_q, e_{answer}) \notin G$. Meanwhile, we will present possible reasoning paths, which helps to explain the process of the reasoning.

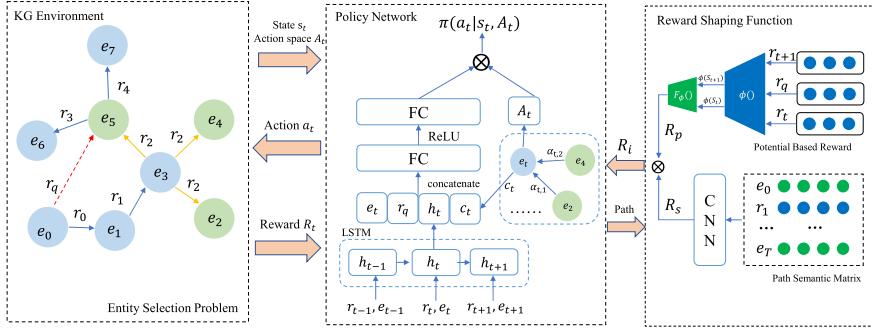


Fig. 1. The architecture of our proposed model

3.2 Reinforcement Learning Formulation

The process of reasoning can be viewed as a Markov Decision Process (MDP) [5], it contains an agent and a KG environment. The agent continuously selects an action by policy network, and the KG environment will feedback a reward according to the action. The main components of the MDP are denoted as follows:

States. The state of agent is a tuple that concatenated with the current entity e_t visited by agent at step t and the global information (e_s, r_q) , where (e_s, r_q) denotes the query entity and query relation from the query answer task $(e_s, r_q, ?)$. More specifically, the state at step t can be formulated as $s_t = (e_t, e_s, r_q)$.

Actions. The possible actions at step t can be defined by the outgoing edges and its linked entities from the current entity e_t . Formally, $A(s_t) = \{(r, e) | (e_t, r, e) \in G\}$. Same as the previous research, we add the self-loop relation and inverse relation to the graph G , i.e., if the triple $(e_s, r, e_o) \in G$, then $(e_s, r_{no_op}, e_s) \in G$ and $(e_o, r^{-1}, e_s) \in G$. Self-loop relation allows the agent to stay in current entity without taking any action. Besides, the agent can find the short path by it when the search is unrolled for a fixed number of steps T . Meanwhile, inverse relation allows our agent to undo a potentially wrong decision [4], and the agent can back to the entity visited at the previous step.

Transition. Because of the properties of the graph, the state of the agent is determined by the current visited entity. Thus, the transition function δ modifies state s_t to new state s_{t+1} based on the action selected by the agent. Formally, $s_{t+1} = \delta(s_t, A_t) = (e_{t+1}, e_s, r_q)$, where $s_t = (e_t, e_s, r_q)$ and $A_t = (r_{t+1}, e_{t+1})$.

Rewards. Reward is the feedback from the KG environment to the agent according to the action taken by the agent. The most commonly used reward is the terminal reward, it can be formulated as $R_t = 1\{e_T = e_{answer}\}$. The agent gets a positive reward (usually 1) if it arrives at the answer entity at final step T and a negative reward (usually 0) otherwise. But the terminal reward will result in the sparse rewards problem, thus we design a reward shaping function, which will be described in detail in Sect. 3.4.

3.3 Attention-Based Policy Network

To solve the MDP problem described above, we parameterize the search policy as a policy network. Moreover, the parameterized policy network takes the state information and actions history selected by the agent as the input and outputs the probability distribution π of action space at each step. The framework of our model is shown in Fig. 1. Formally, the policy network consists of three parts: an LSTM network that encodes the path history; an attention mechanism that obtains more hidden information from neighbor entities and solves the entity selection problem; a feedforward neural network to get the action probability distribution from all possible actions.

In the policy network, each entity and relation have a corresponding vector embedding and use the same symbol to represent for simplicity. i.e. $r_t, e_t \in \mathbb{R}^d$. In this study, we adopt an LSTM network to encode the path history information h_t at step t , and the agent can memorize and learn from the path history in this way. The formula is as follows and the initial history information h_0 set to 0:

$$h_t = LSTM(h_{t-1}, [r_t; e_t]) \quad (1)$$

Typically, many previous researches in path reasoning mainly focus on the relation selection and ignored the importance of entities, which will result in the wrong decision making when 1-N/N-N relation occurs. In reality, the neighbor entities contain many useful hidden features. For example, if an entity is a person and has the neighbor entity *ChicagoBulls*, we can infer that the current entity's work is related to basketball. Therefore, we believe that it is helpful to solve the entity selection problem utilizing the neighbor hidden features extracted by attention. And it can be formulated as follows:

$$c_t = \sum_{i \in N_t} \alpha_{t,i} \cdot e_i \quad (2)$$

$$\alpha_{t,i} = \frac{\exp(\cos(e_t, e_i))}{\sum_{k \in N_t} \exp(\cos(e_t, e_k))} \quad (3)$$

where $c_t \in \mathbb{R}^d$ denotes the neighbor entities information extracted by attention mechanism. $\alpha_{t,i}$ is the attention weight for neighbor entities e_i , which is calculated by the cosine similarity between entity e_t and its neighbor entity e_i . Because of the attention mechanism, the policy network will output more rationale probability instead of random probability for the actions that have the same relation but different entities.

Based on the encoded history path h_t , the attention vector c_t , and the global information (e_s, r_q) from the query triple, the policy network will output the probability distribution $\pi(a_t | s_t, A_t)$ from all possible actions A_t . And it can be formulated as:

$$\pi(a_t | s_t, A_t) = \sigma(A_t \times W_p \text{ReLU}(W_1[e_t; r_q; h_t; c_t])) \quad (4)$$

Where σ is the softmax operator. Finally, we will sample action A_t according to the probability distribution, and the agent will choose an action most likely to be right as the next step action.

3.4 Reward Shaping

Terminal reward is the commonly used reward function in the solution of the MDP problem, and the agent will receive a positive reward only when it arrives at the answer entity at the final step. Consequently, it usually leads to delayed and sparse rewards in large-scale knowledge graph reasoning, which will slow down training efficiency. In this paper, we carry out reward shaping from two levels of potential and semantics.

Potential Based Reward Function. We shape the reward through a potential function, which doesn't need well pretraining. The framework of potential-based reward shaping is proposed in [19]. As the theorem proved by that paper, if F is a potential-based reward shaping function, then every optimal policy in $M' = \langle S, A, P, \gamma, R + F \rangle$ will also be an optimal policy in $M = \langle S, A, P, \gamma, R \rangle$. And the reward shaping function F can be defined as: F is potential based if there exists ϕ s.t. $F(s, a, s') = \gamma\phi(s') - \phi(s)$.

In previous research, Qiu Y et al. assume that a correct decision should contain a relation which covers part of the semantic information of the question in the QA task. And use the cosine similarity between encoded action history and the sum of word embedding in the question as the potential function [1]. Different from it, we only consider the similarity of relation and entity at current step t instead of encoded action history. In this way, the agent will more focus on the relation and entity information at the current step and slow down the negative impact of 1-N/N-N relations. Moreover, we adopt the dot product to measure semantic matching similarity between the current step relation entity $[r_t; e_t]$ and the answer relation entity $[r_q; e_{answer}]$. Consequently, the potential function for the state s_t can be formulated as follows:

$$\phi(s_t) = \begin{cases} \text{ReLU}([r_t; e_t] \cdot [r_q; e_{answer}]), & t > 1 \\ 0, & t = 1 \end{cases} \quad (5)$$

Where ReLU is used to let the potential greater than zero. Based on the potential function, the reward function R_p can be defined as follows and γ is the discount factor.

$$R_p = \gamma\phi(s_{t+1}) - \phi(s_t) \quad (6)$$

Semantic Based Reward Function. In our model, the semantic-based reward function R_s can not only alleviate the sparse reward problem, but also further solve the entity selection problem. To achieve this, we encode the reasoning path to a real value matrix and use a convolutional neural network to distinguish the semantic feature in the path. We consider both entities and relations in the path and believe the semantic feature is helpful to solve the entity selection problem. The formula is as follows:

$$p = e_s \oplus r_1 \oplus e_1 \dots \oplus r_T \oplus e_T \quad (7)$$

$$R_s = \rho(W_2 \text{ReLU}(W_1 \text{ReLU}(p * \omega + b))) \quad (8)$$

Where $p \in \mathbb{R}^{(2T+1) \times d}$ represents the path real value matrix and \oplus denotes the concatenation operator. $(*)$ means the convolution operation, ω is the convolution kernel and b denotes the bias. Then we use two fully connected layers for further semantic feature extraction and ρ represents the sigmoid function to make reward R_s between 0 to 1. Semantic based reward function returns the corresponding reward to the agent according to the semantic feature of the reasoning path, which can help the agent choose more reasonable actions.

Reward Integration. To make the agent have the ability to choose more sensible actions when 1-N/N-N relations occur and get rid of sparse reward, we design a reward integration method from the potential and semantic levels. And it can be formulated as:

$$R_i = \varepsilon R_p + (1 - \varepsilon) R_s \quad (9)$$

$$R(s_t) = R_t + (1 - R_t) R_i \quad (10)$$

Where R_p and R_s represent potential reward and semantic reward respectively, ε is the reward balance weight and $R(s_t)$ is the final reward feedback to the agent. In other words, if the final entity e_T is the answer entity, the agent will receive a positive terminal reward 1, and integration reward R_i otherwise.

3.5 Training

Pre-training. To make the CNN in the semantic reward function has the ability to distinguish the path semantic feature, we use the adversarial training algorithm WGAN-GP [20] to train it with the random path value matrix and the ground truth path value matrix. More specifically, we use the combination of manual and breadth-first search (BFS) algorithms to extract ground truth path between query entity e_s and answer entity e_{answer} . And it can be formulated as follow:

$$\mathcal{L}_s = R_s(p^R) - R_s(p^G) + \lambda (\|\nabla_{\hat{p}} R_s(\hat{p})\|_2 - 1)^2 \quad (11)$$

Where \mathcal{L}_s denotes the loss of the CNN, which consist of original critic loss and gradient penalty. p^R and p^G represent the random path value matrix and the ground truth path value matrix respectively. λ is the gradient penalty coefficient and \hat{p} is sampled uniformly along straight lines between p^R and p^G .

Policy Network Training. Based on our MDP formulation, the policy network is trained by maximizing the expected cumulative reward:

$$J(\theta) = \mathbb{E}_{(e_s, r, e_o) \sim D} \mathbb{E}_{A_1 \dots A_{T-1} \sim \pi_\theta} [R(s_t)|(e_s, r)] \quad (12)$$

Where $\theta = \{W_1, W_p, W_v\}$ is the parameters of the policy network and D is a true underlying distribution. To solve this optimization problem, we use REINFORCE algorithm [5]. Similar to [4], we approximate the second expectation by running multiple rollouts for each training example. And the policy gradient $\nabla_\theta J(\theta)$ is defined as:

$$\nabla_\theta J(\theta) = \nabla_\theta \sum_{t=0}^{T-1} \log \pi(a_t | S_t, A_t) R(s_t) \quad (13)$$

4 Experiments

In this section, we evaluate the performance of our model on two large real-world KG datasets in the query answering task. We first describe the experiment setup, including the introduction of the experiment datasets, baseline models and evaluation methods, and the detail of the implementation. Then we will compare our model with baseline and analyze the results, followed by ablation studies to show the effectiveness of components in our model. Finally, a few case studies also validate the performance of our model for the entity selection problem.

Table 1. Statistics of WN18RR and NELL-995 datasets.

Datasets	#Entities	#Relations	#Triples	#Queries
WN18RR	40945	11	86835	3131
NELL-995	75492	200	154213	3992

4.1 Experiment Setup

Datasets. All experiments were conducted on the WN18RR [13] and NELL-995 [3] datasets, which are created from the WN18 and NELL datasets respectively by removing or modifying some triples. The statistics information of the two datasets is shown in Table 1. And according to the research [6], about 32% of relations in WN18RR and 27% of relations in NELL-995 are 1-N/N-N relations. Thus, it’s important to solve the entity selection problem in the reasoning process when these relations appear.

Baseline Methods and Evaluation. We compare our model against previous state-of-the-art methods, including embedding-based methods (DistMult [11], ComplEx [12], ConvE [13]), path-based methods (MINERVA [4], RS [15], DIVINE [16]), and rule-based methods (RuleGuider [17], GNTPs [18]). The results of baselines come from their paper or reproduction by the open code. Due to lacking small part experimental results or source code in some methods, some baseline results will be omitted in Table 2. For all datasets, we adopt the widely used Hits@k and mean reciprocal rank (MRR) as our evaluation.

Implementation Details. The embedding size of the entity and relation for the two datasets are set to 50, and we use a one-layer LSTM as the path history encoder and set its hidden dimension to 200. Then we set the maximum reasoning path length (the max step number) to 3. Meanwhile, we adopt Adam optimization [21] for parameter optimization with the learning rate $lr = 0.001$. And we set the mini-batch size 512 for the NELL-995 dataset and 256 for the WN18RR dataset. The convolution kernel is set to 3×5 . We conduct parameter sensitivity analysis to determine the values of some important parameters. For the WN18RR dataset, we set the discount factor $\gamma = 0.97$, the reward balance weight $\varepsilon = 0.3$, and the gradient ‘penalty coefficient $\lambda = 15$. For the NELL-995 dataset, we set the discount factor $\gamma = 0.95$, the reward balance weight $\varepsilon = 0.5$, and the gradient penalty coefficient $\lambda = 5$.

4.2 Results and Analysis

In this experiment, we evaluate the performance of our model on the query answering task compared to other baselines on WN18RR and NELL-995 datasets. The results are reported in Table 2, we can find our proposed model consistently outperforms both embedding based methods and interpretable methods on WN18RR dataset in terms of hits@1, hits@3, hits@10, and MRR. Thus, it shows that our model is valid on WN18RR dataset. Furthermore, our model also achieved comparable results on NELL-995 dataset, especially in interpretable reasoning models, but some evaluation indicators are not as good as the best results in the embedding-based methods. We consider the most likely reason is that there are fewer 1-N/N-N relations in test and validation set of NELL-995, so that the attention mechanism and semantic reward function in our model does not play a big role.

To further analyze the reason for the small gap of performance in NELL-995 dataset and the effectiveness of components in our model, we will conduct ablation studies in Sect. 4.3. As for the entity selection problem, we will show how our model solves the entity selection problem in several cases in Sect. 4.4.

Table 2. Overall performance comparison in WN18RR (above) and NELL-995 (below). The results are reported in percentage (%). The best baseline results are marked with stars (*).

Metric	Embedding			Interpretable					
	DistMult	ComplEx	ConvE	MINERVA	RS[15]	RuleGuider	DIVINE	GNTPs	Ours
Hits@1	41.0	38.2	40.3	41.3	43.7	44.3*	43.8	41.0	44.9
Hits@3	44.1	43.3	45.2	45.6	—	—	48.0*	44.2	49.6
Hits@10	47.5	48.0	51.9	51.3	54.2	55.5*	53.8	48.3	56.3
MRR	43.3	41.5	43.8	44.8	47.2	48.0*	46.8	43.4	48.7
Hits@1	61.0	61.2	67.2*	66.3	65.5	66.4-	65.0	—	67.7
Hits@3	73.3	76.1	80.8*	77.3	—	—	75.4	—	78.8
Hits@10	79.5	82.7	86.4*	83.1	83.6	85.9	81.4	—	83.9
MRR	68.0	69.4	74.7*	72.5	72.2	73.6	71.1	—	73.9

4.3 Ablation Study

In this section, we perform an ablation study by removing the components in our model to verify its effectiveness. The result of the ablation study is listed in Table 3, including the performance evaluation and time-consuming analysis.

Effectiveness of Attention Mechanism. To evaluate the effectiveness of the attention mechanism, we compare our model with the ablation model -ATT that removes the attention mechanism. As shown in Table 3, our model performs consistently better than -ATT, which means our model makes the full use of the neighbor entities feature extracted by attention.

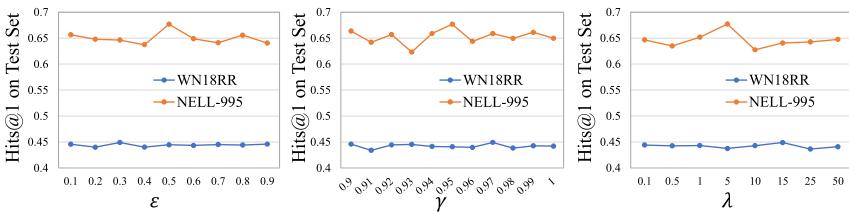
Table 3. The result of ablation study. The results are reported in percentage (%).

Metric	WN18RR					NELL-995				
	-ATT	-PR	-SR	-RS	ALL	-ATT	-PR	-SR	-RS	ALL
Hits@1	44.3	44.2	43.8	44.0	44.9	65.9	65.6	65.5	63.7	67.7
Hits@3	48.6	48.6	48.1	48.2	49.6	78.0	77.4	78.1	77.0	78.8
Hits@10	55.0	55.4	54.7	54.7	56.3	82.6	81.7	83.0	82.0	83.9
MRR	47.6	47.3	47.0	47.2	48.7	72.9	72.3	72.8	71.3	73.9
Training Time	118 s	135 s	122 s	121 s	132 s	109 s	113 s	125 s	128 s	123 s
Reasoning Time	60 s	64 s	60 s	62 s	63 s	29 s	30 s	28 s	30 s	29 s

Effectiveness of Reward Shaping. We design three reward ablation models, removing potential reward -PR, removing semantic reward -SR, and removing all reward shaping function -RS. The comparison result shows that our reward shaping method is effective to alleviate reward sparseness and improve performance.

Time Consuming Analysis. To analyze the impact of the added components on training and reasoning time, we count the time spent by each model in the training phase (50 epochs) and the inference phase on the WN18RR (3134 queries) and NELL-995 (2818 queries). The results are shown in Table 3, we can find that the time spent in inference is basically the same on the two datasets. And the maximum difference in training time is about 15 s.

Parameter Sensitivity Analysis. Figure 2 shows the parameter sensitivity of reward balance weight ε , discount factor γ , and the gradient penalty coefficient λ . We set the value of ε ranges from 0.1 to 0.9, the value of γ ranges from 0.9 to 1, and the range of λ is {0.1, 0.5, 1, 5, 10, 15, 25, 50}. And the parameter values to achieve the best performance are introduced in the implementation details.

**Fig. 2.** Results of parameter sensitivity analysis.

Entity Selection Problem Optimization. To reflect the optimization of the attention mechanism and semantic reward function in the 1-N/N-N relations. We calculate out the proportion of the path contained 1-N/N-N relations reasoned by our model and -ATT and

Table 4. The proportion of 1-N/N-N relations in all paths and feasible paths. The results are reported in percentage (%).

	WN18RR				NELL-995			
	All-path		Feasible-path		All-path		Feasible-path	
	Top-10	Top-50	Top-10	Top-50	Top-10	Top-50	Top-10	Top-50
ALL	54.34	57.78	54.89	63.73	30.80	29.64	27.89	30.54
-ATT	37.35	43.29	46.77	52.61	24.51	29.87	25.04	27.12
-SR	42.32	53.44	48.22	61.14	28.01	27.27	25.70	28.19

-SR. For a more comprehensive analysis, we counted top 10 and top 50 in all paths and feasible paths (the final entity in path is the answer entity). The results are shown in Table 4, which demonstrate our model tends to find path containing 1-N/N-N relations. And we can find that the proportion of path containing 1-N/N-N relations on the NELL-995 is much smaller than WN18RR, which makes the components in our model does not play a big role.

4.4 Case Study

We show some paths found by our model and baselines in Table 5 and entity selection problem that appears in the path is shown in bold. We respectively select a query relation from the WN18RR and NELL-995 datasets. For instance, the query triple of first query relation can be formulated as (02314321, *hypernym*, ?). According to the paths, we find that the path found by MINERVA is wrong due to the 1-N relation *meronym*. As for the self-loop and inverse relation we added to the KG, some cases are found to prove its effectiveness in searching the short path and recovering from mistakes respectively.

Table 5. Examples of reasoning path found by our model and baselines.

Query Relation	Reasoning Path
<i>hypernym</i>	MINERVA: 02314321 $\xrightarrow{\text{member_meronym}^{-1}}$ 02313495 $\xrightarrow{\text{member_meronym}}$ 02313709 $\xrightarrow{\text{hypernym}}$ 01905661 (\times) MARLPaR: 02314321 $\xrightarrow{\text{member_meronym}^{-1}}$ 02313495 $\xrightarrow{\text{member_meronym}}$ 02314001 $\xrightarrow{\text{hypernym}}$ 08102555 (\checkmark) Ours: 02314321 $\xrightarrow{\text{member_meronym}^{-1}}$ 02313495 $\xrightarrow{\text{member_meronym}}$ 02314001 $\xrightarrow{\text{hypernym}}$ 08102555 (\checkmark) Ours: 02314321 $\xrightarrow{\text{member_meronym}^{-1}}$ 02313495 $\xrightarrow{\text{hypernym}}$ 08102555 $\xrightarrow{\text{NO_OP}}$ 08102555 (short path)
<i>works for</i>	MINERVA: ceo_brian_moynihan $\xrightarrow{\text{person_leads_org}}$ bank_bank_america $\xrightarrow{\text{org_terminated_person}}$ ceo_kenneth_lewis $\xrightarrow{\text{person_leads_org}}$ country_america (\times) MARLPaR: ceo_brian_moynihan $\xrightarrow{\text{person_leads_org}}$ bank_bank_america $\xrightarrow{\text{org_terminated_person}}$ ceo_brian_moynihan $\xrightarrow{\text{person_leads_org}}$ bank_bank_america (\checkmark) Ours: ceo_brian_moynihan $\xrightarrow{\text{org_terminated_person}^{-1}}$ bank_bank_america $\xrightarrow{\text{org_terminated_person}}$ ceo_brian_moynihan $\xrightarrow{\text{person_leads_org}}$ bank_bank_america (\checkmark) Ours: ceo_donald_graham $\xrightarrow{\text{org_terminated_person}^{-1}}$ website_the_washington_post $\xrightarrow{\text{org_terminated_person}}$ ceo_donald_graham $\xrightarrow{\text{org_hired_person}^{-1}}$ company_tnt_post (recover from mistakes)

5 Conclusion

In this paper, we develop an attention based reinforcement learning with reward shaping model for query answering task, which can resolve the entity selection problem caused by 1-N/N-N relations in KG. More specifically, we extract the hidden information of the current entity’s neighbors through the attention mechanism, and it helps to train agents to find correct action from the actions that have the same relation but different entities. Moreover, we design a novel reward shaping function to enhance the reward from potential and semantic levels, which helps to solve the entity selection and mitigate sparse rewards problem. We conduct extensive experiments to verify the performance of our model compared with state-of-the-art baselines and some cases also show the effect of our model in entity selection problem. In the future, we will explore research on path diversity to encourage agents to find more diverse paths.

Acknowledgement. This work was in part supported by the Major project of IoV , Technological Innovation Projects in Hubei Province (Grant No. 2020AAA001, 2019AAA024) and Sanya Science and Education Innovation Park of Wuhan University of Technology (Grant No. 2020KF0054).

References

1. Qiu, Y., Wang, Y., Jin, X., Zhang, K.: Stepwise reasoning for multi-relation question answering over knowledge graph with weak supervision. In: WSDM, pp. 474–482 (2020)
2. Xian, Y., Fu, Z., et al.: Reinforcement knowledge graph reasoning for explainable recommendation. In: SIGIR, pp. 285–294 (2019)
3. Xiong, W., Hoang, T., Wang, W.Y.: Deeppath: a reinforcement learning method for knowledge graph reasoning. In: EMNLP, pp. 564–573 (2017)
4. Das, R., Dhuliawala, S., Zaheer, M., et al.: Go for a walk and arrive at the answer: reasoning over paths in knowledge bases using reinforcement learning. In: ICLR (2018)
5. Sutton, R.S., Andrew G.B.: Reinforcement learning: an introduction. MIT press (2018)
6. Li, Z., Jin, X., Guan, S., Wang, Y., Cheng, X.: Path reasoning over knowledge graph: a multi-agent and reinforcement learning based method. In: ICDMW, pp. 929–936 (2018)
7. Bordes, A., et al.: Translating embeddings for modeling multi-relational data. In: Advances in neural information processing systems, pp. 2787–2795 (2013)
8. Wang, Z., et al.: Knowledge graph embedding by translating on hyperplanes. In: AAAI, vol. 14, pp. 1112–1119 (2014)
9. Lin, Y., et al.: Learning entity and relation embeddings for knowledge graph completion. In: AAAI, vol. 15, pp. 2181–2187 (2015)
10. Jia, Y., et al.: Locally adaptive translation for knowledge graph embedding. In: AAAI, pp. 992–998 (2016)
11. Yang, B., Yih, W.T., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint [arXiv:1412.6575](https://arxiv.org/abs/1412.6575) (2014)
12. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: ICML, pp. 2071–2080 (2016)
13. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. In: AAAI Conference on Artificial Intelligence. (2018)

14. Lao, N., Mitchell, T., Cohen, W.W.: Random walk inference and learning in a large scale knowledge base. In: EMNLP, pp. 529–539 (2011)
15. Lin, X.V., Socher, R., Xiong, C.: Multi-hop knowledge graph reasoning with reward shaping. In: EMNLP, pp. 3243–3253 (2018)
16. Li, R., Cheng, X.: DIVINE: a generative adversarial imitation learning framework for knowledge graph reasoning. In: EMNLP-IJCNLP, pp. 2642–2651 (2019)
17. Lei, D., Jiang, G., Gu, X., et al.: Learning collaborative agents with rule guidance for knowledge graph reasoning. In: EMNLP, pp. 8541–8547 (2020)
18. Minervini, P., Bošnjak, M., Rocktäschel, T., et al.: Differentiable reasoning on large knowledge bases and natural language. In: AAAI vol. 34, No. 04, pp. 5182–5190 (2020)
19. Ng, A.Y., Harada, D., Russell, S.: Policy invariance under reward transformations: theory and application to reward shaping. In: ICML, pp. 278–287 (1999)
20. Gulrajani, I., Ahmed, F., Arjovsky, M., et al.: Improved training of wasserstein gans. In: Advances in Neural Information Processing Systems, pp. 5767–5777 (2017)
21. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)



Entity-Aware Relation Representation Learning for Open Relation Extraction

Zihao Liu¹, Yan Zhang¹, Huizhen Wang^{1,2}, and Jingbo Zhu^{1,2(✉)}

¹ NLP Lab, School of Computer Science and Engineering, Northeastern University, Shenyang, China

{zihao.liu,zhangyan}@stumail.neu.edu.cn

² NiuTrans Research, Shenyang, China
{wanghuizhen,zhujingbo}@mail.neu.edu.cn

Abstract. Open relation extraction aims at extracting novel relations from open-domain corpora. However, most recent works typically treat entities and tokens equally while encoding sentences, without taking full advantage of the guiding role of entities in representation learning. In this work, we propose the Entity-Aware Relation Representation learning framework for open relation extraction and establish the new state-of-the-art on standard benchmarks. It gives more attention to entities when learning representations by leveraging an entity-aware attention mechanism. And we further propose a pair-wise contrastive loss to learn relation representations effectively in terms of alignment and uniformity. Extensive experimental results show that our framework achieves significant improvements compared to state-of-the-art models.

Keywords: Open relation extraction · Attention mechanism · Contrastive learning

1 Introduction

Open relation extraction (OpenRE) aims at extracting novel relation types from open-domain corpora, where the relation types may not be pre-defined. OpenRE plays an increasingly important role in various Nature Language Processing (NLP) applications such as knowledge base completion.

Most OpenRE methods can be divided into tagging-based methods and clustering based methods. Tagging-based methods [3, 7] directly extract semantical words or phrases from sentences as relation types.

In contrast, clustering-based methods extract novel relations by clustering instances which is based on the semantic distances between them. Essentially, we consider that clustering-base methods can be decomposed into two modules: relation representation learning module and cluster module. The representation learning module is the core of entire framework, and the cluster module relies heavily on the representations learned by it. Previous OpenRE methods proposed different architectures for better representation learning module such as siamese

network [21] and large pretrained language models [10], and they all achieved significant improvements thanks to this.

Einstein was born in the Germany , but moved to Switzerland in 1895.

Einstein was born in the Germany, but moved to Switzerland in 1895.

Fig. 1. Models need to be aware of entity pair. The sentence will appear multiple times in the distant-supervised dataset with a different entity pair at each time. For a sentence, the blues indicate the entity pair, and red indicates their relation’s surface-form. (Color figure online)

However, learning relation representations is challenging due to the numerous surface forms of relations in distant-supervised open-domain corpora. Although recent works [5,11] propose to learn relation representations from knowledge bases, they cannot handle the unstructured and distant-supervised text due to the enormous label noise.

As shown in Fig. 1, the different entity pairs in same sentence may express distinct relations. Both entities and context provide critical information for relation extraction [14], where entities provide most of the information and play a critical guiding role for OpenRE [19]. This makes it necessary for the models to be more aware of entities when learning relation representations. However, most OpenRE methods are not aware of the importance of entities, either only utilizes the contextualized entity pair representation output by BERT to cluster relations [10] or even completely ignore the entities when predicting relations [17].

In this paper, we propose the entity-aware attention mechanism to give more attention to the entities when learning relation representations. And we further propose the pair-wise contrastive loss. It leverage the idea of contrastive learning to pull semantically close neighbors together and push apart semantically non-neighbors, which helps the model learn relation representations effectively in terms of alignment and uni-formity. We summarize our contributions as follows:

- (1) We propose the entity-aware attention mechanism, which essentially makes the model gives more attention to entities when encoding instances.
- (2) We propose the pair-wise contrastive loss to help model learn relation representations effectively in terms of alignment and uni-formity.
- (3) We conduct extensive experiments on three real-world human-curated or distant-supervised datasets (FewRel, T-Rex SPO, and T-Rex DS). Experimental results show that our framework significantly outperform supervised or unsupervised state-of-the-art OpenRE models.

2 Related Work

Open relation extraction aims to discover novel relations from unsupervised open-domain corpora and has recently attracted increasing attention. Compared with traditional RE methods, OpenRE can handle the open-ended growth of new relation types well. Most OpenRE methods can divide into tagging-based methods and clustering-based methods. Tagging-based methods cast OpenRE as a sequence labeling problem, and extract surface forms of relations from plain text in unsupervised [2,3] or supervised [7,18] paradigms. However, tagging-based methods heavily rely on relational words or phrases in sentences and cannot extract implicit relations.

In contrast, traditional clustering-based OpenRE methods extract novel relations by clustering rich features extracted by external linguistic tools [8, 12, 22]. [13] proposes discrete-state variational autoencoder (VAE) that reconstructs arguments relying on predicted relations to optimizes a relation classifier. However, the VAE is hard to train without supervision. To train model stably, [17] introduces a skewness loss which encourages the classifier to predict relations with confidence. To exploits weak, self-supervised signals, [10] leverages pre-trained language models to learn relation representations and cluster in a self-supervised learning framework. Compared with previous unsupervised methods, [21] utilize relational siamese networks to transfer relational similarity knowledge from supervised data to discover novel relations in unlabeled data, and achieves state-of-the-art performance.

However, previous OpenRE methods are not aware of the importance of entities in discovering novel relations. And in this paper, we mainly focus on entity-aware clustering-based OpenRE methods to discover novel relations.

3 Problem Definition

In this section, we formally defined the problem of relation discovery task. Let E be a set of entities. And let $\mathcal{D} = [(S^0, r^0, (e_1^0, e_2^0)) \dots (S^M, r^M, (e_1^M, e_2^M))]$ be a corpus of M instances. Each instance contains sentence S^i , entity pair (e_1^i, e_2^i) and relation r^i . The input of the problem is a sentence S which consists of N tokens and an entity pair (e_1, e_2) appears in it. And we decompose the problem into two sub-tasks: relation representation learning and relation clustering.

For relation representation learning task, let F_θ denote a function of learning relation representation from instance. This sub-task takes sentence S , labeled with two entities e_1, e_2 , as input and output relation representation z^r which is corresponding to relation r .

For relation clustering task, let \mathcal{K} denote a set of relation clusters and the size of \mathcal{K} is unknown in advance. This sub-task is, for the input relation representation z^r , to predict the relation cluster label $y_k \in \mathcal{K}$ where the subscripts k denote the index of cluster.

We aim to build a model which takes sentence S and labeled entity pair (e_1^0, e_2^0) as input and predicts cluster label y_k . During the evaluation, y_k is compared against the ground truth relation type.

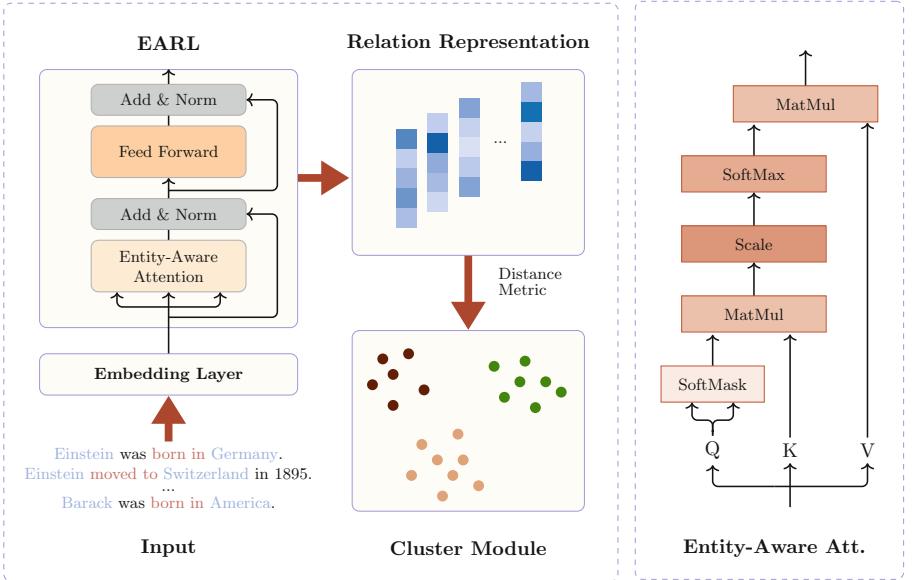


Fig. 2. Left: The architecture of our framework. Our model takes sentences with entity pairs as inputs and clusters them based on the distance between relation representations learned by Entity-Aware Relation Representation Module. **Right:** Our proposed Entity-Aware Attention mechanism.

4 Method

In this section, we will describe our model which consists of an entity-aware relation representation learning module and a relation clustering module. As shown in Fig. 2, our representation module takes sentences as inputs and learns relation representations. Then the clustering module clusters instances to extract novel relations via distance between learned relation representations. We will detail each component below.

4.1 Entity-Aware Relation Representation Learning

The representation learning module is focus on learning mappings from sentence S to relation representation z^i . The sentences will have similar representations if express the same relation, otherwise the learned representations will be dissimilar with each other.

Sentence Embedder. For a sentence S consisting of N words, we insert entity markers to highlight the entities before embedding the sentence. Specifically, we place the entity markers, $\langle e1 \rangle$, $\langle /e1 \rangle$, $\langle e2 \rangle$ and $\langle /e2 \rangle$, before and after the two entities.

Following previous approaches [21, 23], for a sentence S , We use pretrained d_1 -dimensional GloVe vectors [15] to initialize the word embeddings $w \in \mathbb{R}^{N \times d_1}$. We also randomly initialize two d_2 -dimensional position embeddings $p \in \mathbb{R}^{N \times d_2}$ representing the positions of entities. Finally we concatenate the embeddings to form a sequence of vectors $z_0 \in \mathbb{R}^{N \times (d_1+2d_2)}$ as the initial relation representation. The i -th vector corresponds to the i -th tokens of the input sentence.

Entity-Aware Sentence Encoder. Given the strong performance of the attention mechanism, specifically Transformer [20], on encoding instances into a feature space, we adopt it as our relation representation encoder.

Considering the importance of entities in previous OpenRE methods [8, 10], we propose an entity-aware attention mechanism to discover novel relations with rich entity features. We replace the multi-head attention with our proposed entity-aware attention in Transformer architecture and get the relation representation encoder in this paper. In the following, we will detail the mechanism and advantages of the proposed attention mechanism in the OpenRE task.

The entity-aware attention mechanism is a variation of self-attention mechanism, where the input \mathbf{Q} (queries), \mathbf{K} (keys), and \mathbf{V} (values) are exactly the same and essentially the output of previous layer z_{l-1} . The output of our entity-aware attention, z_l , is computed as the weighted sum of the values, where the weight assigned to each value is determined by a compatibility function of the query with all keys as follows:

$$z_l = \text{softmax}\left(\frac{f_W(\mathbf{Q}, \mathbf{K})}{\sqrt{d_k}}\right)\mathbf{V} \quad (1)$$

where $1/\sqrt{d_k}$ is scale factor and

$$d_k = \dim(\mathbf{K}), \mathbf{Q} = \mathbf{K} = \mathbf{V} = z_{l-1}$$

To make the sentence encoder give more attention to the entities when learning relation representations, We design a compatibility function f_{W_i} . Specifically, we, inspired by [24], use a soft mask for \mathbf{Q} to assign low weight to the tokens in sentences which are not entities. For each query Q_i and key K_i , we define f_{W_i} as follows:

$$\begin{aligned} f_{W_i}(Q_i, K_i) &= \text{softmax}(Q_i)K_i^T \\ &= (Q'_i W_i + Q_i(1 - W_i))K_i^T \end{aligned} \quad (2)$$

where Q_i is a token embedding which is obtained by masking the non-tokens of sentence, and Q'_i is the entity embedding which is obtained by masking the non-entities of Q_i . The W_i is a learnable parameter and it measures how much attention the model pays to entities. Equation 2 reduces the weight of non-entities and leaves weight of entities unchanged, which enables the model to pay more attention to entities when encoder sentences. The soft masking is an extension of conventional hard masking in the sense that the former degenerates to the latter iff $W_i = 1$. And the remaining parts of sentence encoder are similar to Transformer.

Pair-Wise Contrastive Loss. Considering the remarkable success of contrastive learning approaches in the representation learning task, we define a contrastive loss to make sure that the relation representations sharing the same relation should be close together, while different relations should be apart.

To take full advantage of relational semantics during training, we first randomly sample N instances with the probability of sharing the same relation γ and construct a pair-wise batch by combining instances with each other from the sampled instances. Then the pair-wise contrastive loss is computed as follows:

$$\mathcal{L}_{pwc} = -\frac{1}{N^2} \sum_{i,j} (1 - \delta(i, j)) d(z^i, z^j) - \delta(i, j) d(z^i, z^j) \quad (3)$$

where $\delta(i, j) \in \{0, 1\}$ is an indicator function that takes the value 1 iff i shares the same relation with j . And the distance metric $d(z^i, z^j)$ between two representations is defined as follows, and we use it to determine whether or not z^i and z^j represent the same relation:

$$d(z^i, z^j) = 1/(1 + \exp(\frac{z^i}{\|z^i\|} \cdot \frac{z^j}{\|z^j\|})) \quad (4)$$

When minimizes the pair-wise contrastive loss, the model can learn effective representations by minimizing the average distance of neighbors with the same relation and maximizing the semantic distance of non-neighbors. Following previous OpenRE method [21], we also adopt a cross-entropy loss \mathcal{L}_{ce} and a virtual adversarial training loss \mathcal{L}_v . The final loss is defined as a weighted sum of the above three losses:

$$\mathcal{L} = \mathcal{L}_{ce} + \mathcal{L}_v + \mathcal{L}_{pwc} \quad (5)$$

4.2 Relation Clustering

After the representation learning module is trained, we consider the outputs as the relation representations of the corresponding instances. For any pair of relation representations (z^i, z^j) , we regard each representation as a node and connect them with an edge if $d(z^i, z^j)$ is grater than threshold ϕ , otherwise not, and the edge indicates that they represent the same relation. Then we get a relation representation graph. Finally, we use louvain algorithm [4] to extract novel relations via clustering representations based on the relation graph.

The K-means and hierarchical agglomerative clustering (HAC) algorithms which are typically used in cluster-based methods [21, 22] both needs the exact number of clusters in advance. In contrast, louvain is a graph-based clustering algorithm used for detecting communities. It will automatically find the proper number of clusters by optimizing community modularity and does not need the number of clusters in advance, which is fully compatible with OpenRE.

Additionally, Louvain might produce clusters with few instances which cannot be regarded as novel relations. To solve this problem, we relabel these instances with the labels of their closest labeled neighbors.

5 Experiments

In this section, we conduct extensive experiments on both human-curated or distant-supervised datasets to show the effectiveness of our model on the OpenRE task. We also give a detailed analysis of experiments to show the advantages and a better understanding of our model.

5.1 Datasets

Following previous works [10, 17, 21], we evaluate our framework on three datasets, namely FewRel [9], T-REx SPO, and T-REx DS dataset.

FewRel is a supervised dataset derived from Wikipedia and annotated by crowd workers. It consists of 100 relations with 700 instances each¹. The relation of each sentence is first annotated by distant supervision methods and then filtered by crowd workers.

Following the dataset settings in [17], We also construct two datasets from T-REx², namely T-REx DS and T-REx SPO. We only consider instances where both entities appear in sentences. And if a sentence contains multiple entity pairs, it will appear multiple times with a different entity pair at each time. The only difference between T-REx DS and T-REx SPO is the corresponding relation surface forms appear in sentences of T-REx SPO.

5.2 Datasets Division

Table 1 presents the details of our datasets based on the above constraints. Considering the goal of OpenRE is novel relation discovery, our datasets division strategy is slightly different from previous OpenRE methods [10, 17] on two T-REx datasets. To best evaluate the performance of our proposed models on relation discovery, we randomly select 64% and 16% of the relations and corresponding instances for training and validation. And the remaining will be used as the test set for evaluation.

Table 1. The details of our datasets division. For FewRel, we do not use validation set. The Rel. and Ins. column represents the number of relations and instances, respectively.

Dataset	Train		Validation		Test	
	#Rel.	#Ins.	#Rel.	#Ins.	#Rel.	#Ins.
FewRel	64	44,800	—	—	16	11,200
T-REx SPO	237	608,043	59	134,545	75	199,026
T-REx DS	417	7,281,837	104	2,203,547	131	4,602,098

¹ The FewRel benchmark provides a training set with 64 relations, a validation set with 16 relations, and a hidden test set with 20 relations which are only for evaluation on <https://thunlp.github.io/1/fewrel1.html>.

² <https://hadyelsahar.github.io/t-rex/>.

5.3 Baseline and Model

We compare our model against several state-of-the-art models on OpenRE benchmarks. Additionally, we evaluate the performance of our proposed framework in a practical, yet more challenging setting: we assume the test set only consist of new relations which will not appear in train set or validation set, and the number of golden relations is not known for models.

March [13] proposes discrete-state variational autoencoder (**VAE**) which reconstructs arguments relying on relation predicted by the encoding component. **PCNN+ $\mathcal{L}_S+\mathcal{L}_D$** [17], a discriminative method differs from March, trains on unsupervised datasets using piece-wise convolutional network (PCNN) and predicts relations with confidence while several relation types are predicted over a mini-batch. **SelfORE** [10] utilizes self-training to iteratively learn relation representations and clusters and archives state-of-the-art on OpenRE task.

Relational Siamese Network augmented with louvain cluster algorithm and virtual adversarial loss(**RSN-LV**, [21]) is the state-of-the-art OpenRE method that transfers relational knowledge from labeled data to discovery relations in unlabeled data.

5.4 Evaluation Metrics

Following the previous OpenRE methods [10, 17, 21], we use the standard evaluation protocol and adopt instance-level evaluation metrics to evaluate relation clustering, including B-Cubed metric (B³, [1]), V-measure [16] and Adjusted Rand Index (ARI, [6]). To properly evaluation, we adopt the majority of ground truth labels in each relation cluster as the prediction label of all samples in the cluster.

5.5 Implementation Details

Hyperparameters. In the embedding layer, we use pre-trained 50-dimensional GloVe word embeddings³ to initialize the word embeddings and randomly initialize two 5-dimensional position embeddings. We put a dropout layer after the sentence embedding layer where the dropout rate is 0.2. During training, the batch size is 64, and the probability of same relations $\gamma = 0.06$. To construct each pair-wise batch, we randomly sample 16 relations and 4 instances each where the seed is 0. For optimization, we use Adam optimizer with a learning rate of 1e-4. For clustering, ϕ is 0.5 selected from {0.4, 0.5, 0.6}. And we use the number of relations in test set when evaluating with HAC.

Dataset Settings. For the FewRel dataset, we follow the settings used in [21] to make a fair comparison with it and do not use the validation set for tuning the hyperparameters. We use 16 relations with 700 instances each for evaluation on FewRel.

³ <https://nlp.stanford.edu/projects/glove/>.

For T-REx SPO and T-REx DS datasets, we also follow the settings in previous works [10, 17] to compare with them faithfully where all models are trained with 10 relations. Consider the goal of novel relation discovery and more challenging settings, we randomly sample 10 relations as the train relation type set for each run which makes the model achieve more generalizability. Considering the size of the test set and cost of the evaluation, we randomly sample 20000 instances from the original test set on T-REx SPO and T-REx DS to report the final scores.

We report the average scores of 10 runs for our model. For each run, we keep the model that achieves the highest averaged B^3 F1, V-measure F1 and ARI on 1000 randomly sampled instances from the validation set, and evaluate and report its score on the test set.

5.6 Results and Analysis

Table 2. The B^3 , V-measure and ARI metrics on FewRel, T-REx SPO, and T-REx DS. ♦: results from [21]; ♠: results from [10]. We highlight the highest numbers among models on same dataset.

Dataset	Model	B^3			V-measure			ARI
		F1	Prec	Rec	F1	Hom	Comp	
FewRel	VAE ♦ [13]	17.9	69.7	28.5	-	-	-	-
	RW-HAC ♦ [8]	31.8	46.0	37.6	-	-	-	-
	RSN-LV ♦ [21]	59.9	77.5	48.9	-	-	-	-
	RSN-LV	63.8	51.8	83.2	74.5	66.7	84.5	50.9
	EARL	71.7	65.1	79.7	79.2	75.4	83.5	61.5
T-REx SPO	VAE ♠ [13]	24.8	20.6	31.3	23.6	19.1	30.6	12.6
	PCNN+ $\mathcal{L}_S+\mathcal{L}_D$ ♦ [17]	36.3	28.4	50.3	41.1	33.7	53.6	21.3
	SelfORE ♠ [10]	41.0	39.4	42.8	41.4	40.3	42.5	33.7
	RSN-LV	58.5	51.7	67.8	37.8	30.9	49.1	27.7
	EARL	67.5	68.4	66.7	63.2	59.7	67.2	47.0
T-REx DS	VAE ♠ [13]	9.0	6.4	15.5	5.7	4.5	7.9	1.9
	PCNN+ $\mathcal{L}_S+\mathcal{L}_D$ ♦ [17]	19.7	14.0	33.4	26.6	20.8	36.8	9.4
	SelfORE ♠ [10]	32.9	29.7	36.8	32.4	30.1	35.1	20.1
	RSN-LV	41.2	38.4	45.0	31.0	27.9	35.2	17.3
	EARL	47.8	44.2	52.4	39.5	34.5	46.4	28.4

Main Results. We report the B^3 , V-measure and ARI metrics on FewRel, T-REx SPO, and T-REx DS, and compares our framework EARL to previous state-of-the-art unsupervised and supervised OpenRE methods. The results are presented in Table 2, from which we can observe that:

- (1) EARL models achieve the best performance and significantly outperform the previous state-of-the-art methods both on the B^3 F1, V-measure F1, and

ARI. For example, compared to RSN-LV, EARL achieves 7.9%, 4.7%, 10.6% improvements in B^3 F1, V-measure F1 and ARI on the FewRel respectively. And compared to previous unsupervised methods, EARL achieves more than 30% improvements, and the performance gap is even greater. It indicates that EARL can effectively leverage the idea of contrastive and entity pairs to learn better semantic representations of novel relations.

- (2) EARL models perform well on all three datasets constructed by crowd workers or auto-labeled by aligning sentences with Freebase. Although T-Rex SPO and T-Rex DS dataset contain a lot of label noise, EARL models achieve significant improvements compared with previous supervised and unsupervised approaches. It will be attributed to our proposed entity-aware relation encoder, and the entity-aware attention mechanism make model treat the entities as “anchors” and it alleviates the misleading effects of multiple occurrences of the same sentence on relation representation learning module.

Ablation Study. As shown in Table 3, We conduct ablations to investigate how different the proposed entity-aware attention mechanism and pair-wise contrastive loss affect EARL’s performance, and we also analyze and verify the impact of different clustering algorithms on OpenRE. All results are evaluated on the test set of FewRel containing 11,200 instances, except for HACs which are time consuming, and they are evaluated with randomly sampled 2000 instances and the same seed. Experimental results show that all components contribute to the final performance.

Table 3. Ablation results on FewRel (%).

Model	B^3			V-measure			ARI
	F1	Prec.	Rec.	F1	Hom.	Comp.	
RSN-HAC	61.8	54.8	70.8	73.1	69.9	76.6	52.4
EARL-HAC	64.5	58.6	71.8	74.7	72.0	77.6	54.6
EARL	71.7	65.1	79.7	79.2	75.4	83.5	61.5
w/o Entity-Aware Enc	67.6	56.4	84.5	77.4	70.3	86.0	55.7
w/o \mathcal{L}_{pwc}	63.8	51.8	83.2	74.5	66.7	84.5	50.9

We find that replacing our entity-aware sentence encoder with CNN like [21] will decrease the B^3 precision and V-measure Homogeneity, and gain improvements in B^3 recall and V-measure completeness. It shows that the proposed entity-aware attention mechanism allows our model to learn more precise semantical relation representations with the help of treating the entities as “anchors”.

And removing pair-wise contrastive loss after replacing the sentence encoder, which is essentially the RSN-LV [21], contributes to a severe performance drop in all the metrics. This means our proposed pair-wise contrastive loss let the model obtain a better distribution of representations in terms of alignment and uniformity.

In addition, we replace the louvain with HAC and the performance of all metrics is severely degraded especially B³. One explanation is that model does not put additional constraints on the prior distribution of relational vectors and therefore the relation clusters might have odd shapes in violation of HAC’s assumption [21].

Model Generalization. We also conduct experiments to investigate the generalizability of EARL. As shown in Table 4, for evaluation on T-REx SPO and T-REx DS, EARL trained on FewRel can easily achieve over 97% and 93% of the best performance. It indicates the generalization ability of EARL and it can effectively discover novel relations from different instance and relation distributions. And it also shows that the quality of labeled instances is much more important than quantity for OpenRE.

Table 4. Model generalization experiments.

Test Set	Train Set	B ³ -F1	V-measure-F1	ARI
T-REx SPO	T-REx SPO	67.5	63.2	47.0
	FewRel	65.7(97%)	59.0(93%)	42.4(90%)
T-REx DS	T-REx DS	47.8	39.5	28.4
	FewRel	44.6(93%)	39.3(99%)	19.5(69%)

6 Conclusions

In this paper, we propose an Entity-Aware Relation Representation learning model for novel relation discovery and establish the new state-of-the-art on standard benchmarks. Our model uses an entity-aware sentence encoder that treats entities as “anchors” to help model obtain precise relation representations during encoding sentences, and also we propose the pair-wise contrastive learning loss to learn relation representations effectively in terms of alignment and uniformity. We conduct extensive experiments and analyses to understand why our models significantly surpass previous OpenRE models and achieve new state-of-the-art performance. For future research, we will try to explore more directions such as better representation encoder and joint open relation extraction.

References

1. Bagga, A., Baldwin, B.: Entity-based cross-document coreferencing using the vector space model. In: COLING-ACL (1998)
2. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O.: Open information extraction from the web (2008)
3. Banko, M., Etzioni, O.: The tradeoffs between open and traditional relation extraction. In: ACL (2008)

4. Blondel, V., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, 10008 (2008)
5. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: NIPS (2013)
6. Cugmas, M., Ferligoj, A.: On comparing partitions. International Federation of Classification Societies (2015)
7. Cui, L., Wei, F., Zhou, M.: Neural open information extraction (2018)
8. Elsayar, H., Demidova, E., Gottschalk, S., Gravier, C., Laforest, F.: Unsupervised open relation extraction. In: Blomqvist, E., Hose, K., Paulheim, H., Lawrynowicz, A., Ciravegna, F., Hartig, O. (eds.) ESWC 2017. LNCS, vol. 10577, pp. 12–16. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70407-4_3
9. Han, X., et al.: FewRel: a large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. ArXiv abs/1810.10147 (2018)
10. Hu, X., Wen, L., Xu, Y., Zhang, C., Yu, P.S.: SelfORE: self-supervised relational feature learning for open relation extraction. ArXiv abs/2004.02438 (2020)
11. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: ACL (2015)
12. Lin, D., Pantel, P.: Dirt @sbt@discovery of inference rules from text. In: KDD 2001 (2001)
13. Marcheggiani, D., Titov, I.: Discrete-state variational autoencoders for joint discovery and factorization of relations. *Trans. Assoc. Comput. Linguist.* **4**, 231–244 (2016)
14. Peng, H., et al.: Learning from context or names? An empirical study on neural relation extraction. In: EMNLP (2020)
15. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: EMNLP (2014)
16. Rosenberg, A., Hirschberg, J.: V-measure: a conditional entropy-based external cluster evaluation measure. In: EMNLP-CoNLL (2007)
17. Simon, É., Guigue, V., Piwowarski, B.: Unsupervised information extraction: regularizing discriminative approaches with relation distribution losses. In: ACL (2019)
18. Stanovsky, G., Michael, J., Zettlemoyer, L., Dagan, I.: Supervised open information extraction. In: NAACL-HLT (2018)
19. Tran, T.T., Le, P., Ananiadou, S.: Revisiting unsupervised relation extraction. ArXiv abs/2005.00087 (2020)
20. Vaswani, A., et al.: Attention is all you need. ArXiv abs/1706.03762 (2017)
21. Wu, R., et al.: Open relation extraction: Relational knowledge transfer from supervised data to unsupervised data. In: EMNLP/IJCNLP (2019)
22. Yao, L., Riedel, S., McCallum, A.: Unsupervised relation discovery with sense disambiguation. In: ACL (2012)
23. Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J.: Relation classification via convolutional deep neural network. In: COLING (2014)
24. Zhang, S., Huang, H., Liu, J., Li, H.: Spelling error correction with soft-masked BERT (2020)



ReMERT: Relational Memory-Based Extraction for Relational Triples

Chongshuai Zhao¹, Xudong Dai², Lin Feng¹, and Peng Liu^{1(✉)}

¹ National Joint Engineering Laboratory of Internet Applied Technology of Mines,
China University of Mining and Technology, Xuzhou, Jiangsu, China

{zhaochs,lynne515,liupeng}@cumt.edu.cn

² IFLYTEK, Hefei, Anhui, China

xddai@iflytek.com

Abstract. Relational triples extraction aims to detect entity pairs (subjects, objects) along with their relations. Previous work failed to deal with complex relationship triples, such as overlapping triples and nested entities, and lacked semantic representation in the process of extracting entity pairs and relationships. To mitigate these issues, we propose a joint extraction model called ReMERT, which first decomposes the joint extraction task into three interrelated subtasks, namely RSE (Relation-specific Subject Extraction), RM (Relational Memory) module construction and OE (Object Extraction). The first subtask is to distinguish all subjects that may be involved with target relations, the second is to retrieve target relational representation from RM module, and the last is to identify corresponding objects for each specific (s, r) pair. Additionally, RSE and OE subtasks are further deconstructed into sequence labeling problems based on the proposed hierarchical binary tagging scheme. Owing to the reasonable decomposition strategy, the proposed model can fully capture the semantic interdependency between different subtasks, as well as reduce noise from irrelevant entity pairs. Experimental results show that the proposed method outperforms previous work by 0.8% (F1 score), achieving a new state-of-the-art on Chinese DuIE datasets. We also adopt sufficient experiments and obtain promising results both in public English NYT and Chinese DuIE datasets.

Keywords: Relation extraction · Hierarchical tagging schema · Overlapping triple · Nested entity · Relation memory

1 Introduction

Relational triple extraction (RE) task plays a critical role in natural language processing (NLP) and knowledge graphs (KGs) construction, which refers to extracting the predefined relation categories between entity pairs from unstructured raw texts. It can be formalized as a relational triple $T = (s, r, o)$, where we call s and o as the subject and object entity of triple T , and r represents a specific relational type between them.

Early work on RE task mainly followed the pipeline ideas [1, 2]. They extracted relational triples in serial NER and RC steps, which suffer from the error propagation problem and neglect the relevance of the two steps. To address these issues, the NLP community began to explore joint models that aim to learn entities and relations simultaneously. For example, Zheng et al. [3] presented a novel tagging schema, which turned the extracting tasks into a sequential tagging problem, bridging the information gap between the NER and RC steps.

Despite the great success of previous work, conventional joint models suffer from two serious restrictions, namely, overlapping triples and nested entities. Figure 1 illustrates these restrictions. Taking the first sentence in Fig. 1 as an example, the RC module in some joint models [4] could predict only one triple (**WuJing**, **Direct_movie**, **Wolf Warrior**) or (**WuJing**, **Act.in**, **Wolf Warrior**), and the tagging scheme of [3] is unable to represent both **Direct_movie** and **Act.in** by a single tag for each word. Some novel schemas have been proposed to handle overlapping triple problem [5] and significant improvement have been achieved. However, due to the problem of nested entities, the prior models cannot hold all the triples in a sentence through entity recognition module, as shown in the lower part of Fig. 1. For example, Wei et al. [6] proposed a new relational triples extraction framework, including subject tagging module and specific relation tagging module. But it cannot address the case of nested entities, that is, **PLA** is part of **PLA Army Air Force University**. To address the problem of nested entities, TPLinker [7] demonstrated an end-to-end sequence labeling model with handshaking tagging schema from the view of module design, which represent each sequence as $l \times l$ (l is the length of input sequence) matrix. However, as the text lengthens, the size of the matrix and the cost of processing long text will undoubtedly increase. Besides, [8] is convinced that these methods lack an valid representation of the relation types, because relation is determined by the entity pairs, so it should be represented as a matrix rather than one-hot vectors. Neither Casrel nor TPLinker use the semantic information of the relation type.

To overcome the aforementioned drawbacks, we design a novel framework called **ReMERT** (**R**elational **M**emory-based **E**xtraction for **R**elational **T**riples). Given a sentence, ReMERT aims to answer three questions: “What are all the possible subjects?”, “What are the involved relations for each subject?”, and “Which object(s) can be obtained according to specific (s, r) pair?”.

Briefly, this work has the following two contributions: (1) An innovative framework is designed to enrich the representation of relation types of complex RE tasks. (2) The proposed model has excellent performance on English NYT dataset, and achieves the new state-of-the-art on the Chinese DuIE dataset.

2 Related Work

Early work could be summarized into two major approaches: the pipeline methods and joint methods. The pipeline methods [1, 2] are divided into two steps: First named entity recognition (NER) to extract all potential entities in the

Overlapping Triples	EPO	<p>Wujing directed and acted in "Wolf Warrior".</p> <p><WuJing, Directed_movie, Wolf Warrior ></p> <p>< WuJing, Act_in, Wolf Warrior ></p>
	SPO	<p>The Tiananmen is a monumental gate in Beijing, the capital city of China.</p> <p><Tiananmen, Locate_in, Beijing></p> <p><Beijing, Capital_of, China></p>
Nested Entities	Inner-NE	<p>"Selected Works of Lan Hongwen" is a book published by Renmin University of China Press in 2007.</p> <p>< Selected Works of Lan Hongwen, Author_is, Lan Hongwen></p> <p>< Selected Works of Lan Hongwen, Publish_by, Renmin University Press of China></p>
	Extra-NE	<p>The PLA Army Air Force University is located in Jilin Province.</p> <p>< PLA Army Air Force University, Locate_in, Jilin Province></p> <p>< PLA, Child_unit, PLA Army Air Force University></p>

Fig. 1. Examples of overlapping triples and nested entities. In overlapping triples, EPO shows at least two relations overlapping in the same entity-pair. SEO means that at least two relations sharing a single entity in sample. In nested entities, Inner-NE as in the same relation, entity pairs are nested. Extra-NE as entity pairs is nested between different relations. Bold texts are the nested parts.

input sequences. Then the entities are combined in pairs, and their relations are classified by the relation classification (RC) module. However, [9] points out that the pipeline methods ignores the interaction between the NER and RC modules, and suffer from the error propagation problem. To address these problems, recent work proposes joint methods [3], which aims to obtain triples simultaneously through the combined module of NER and RC. Compared with the pipeline methods, the joint methods can simultaneously extract and leverage the deep correlations between entities and relations. These models have achieved superior performance than traditional pipeline methods.

Although the above work showed promising results, they completely abandoned overlapping triples. Zeng et al. [5] first studied this problem and classified sentences with overlapping triples into *EPO* and *SEO*. They proposed an end-to-end model that considers relation extraction as a problem of generating triples with copy mechanism. GraphRel [10] adopts graph convolutional networks, which excelled the former methods in solving the problem of overlapping triples by incorporating the regional and sequential dependency features of words. CasRel [6] provided a fresh perspective for revisiting the RE task by modeling the relations as a function that maps subjects to objects. [8] calculates the relational score matrix of entity pairs on all relation types by extracting different entity embeddings from the pretrained language model, yet it costs much longer time to compute a $l \times l$ matrix with label.

In this paper, we propose a new extraction framework called ReMERT to solve three problems: overlapping triples, nested entities and insufficient representation of relation types. Experiments prove that the proposed framework performs better than previous work.

3 Methodology

Relational triples extraction is used to identify all potential relational triples in a sentence, some of which face bottlenecks of overlapping triples and nested entities. In order to break these bottlenecks, we adopted the idea of hierarchical sequence labeling and designed a joint model to directly extract relational triples. As shown in Fig. 2, first, the model encodes the sentences with shared BERT[11] encoder. Second, build a Relation-specific Subject Extractor (RSE) to extract subjects and related relation types. Then, we get the embedding of the relation type from the RM module. Finally, for each extracted (s, r) pair, the Object Extractor (OE) is triggered to detect object(s).

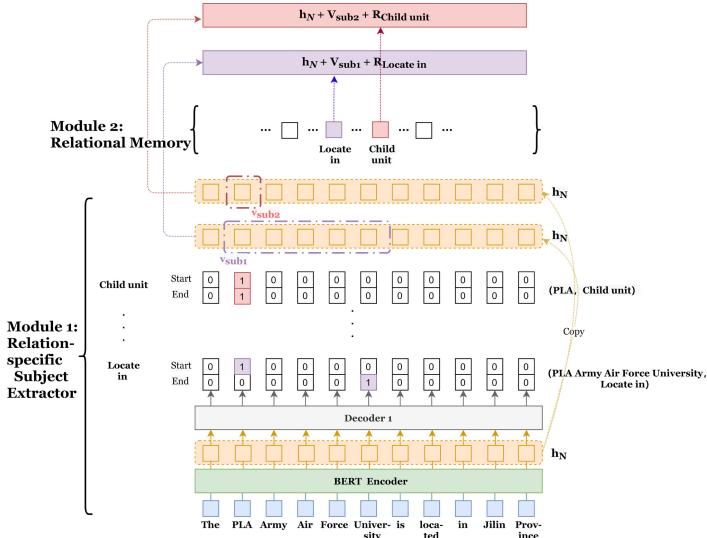


Fig. 2. The framework of RSE module and RM module. In this example, two candidate subjects are detected at the low level, and the presented 0/1 tagger indicates the start and end positions of nested subjects in different relation types. In this way, two different $(\text{subject}, \text{relation})$ pairs (PLA, Child) and (PLA Army Air Force University, Locate in) can be obtained.

We wrap these extractors into a general paradigm called Relational Memory-based Extraction for Relational Triples (abbreviated as ReMERT). Such paradigm can be understood by decomposing the joint probability of triple extraction into condition probability:

$$\prod_{(s, r, o) \in T} P((s, r, o)|S) = \prod_{(s, r) \in T} P((s, r)|S) \prod_{o \in T|(s, r)} P(o|(s, r), S) \quad (1)$$

where given an annotated sentence S and a set of potential triples $T = (s, r, o)$ in S , $(s, r) \in T$ denotes a subject appearing in the triples T . $T|(s, r)$ is the set

of triples led by (s, r) pair in T . $o \in T|(s, r)$ is an object led by subject s and corresponding relation r in T . We aim to maximize the data likelihood of Eq. 1 on training dataset. In this manner, the OE module can consider the semantics of a given (s, r) pair when extracting objects. Besides, this paradigm no longer extracts all entities in the first step, only identifying subjects that may participate in the target triple. Intuitively, the framework provides two benefits. First, we make no assumption on how multiple triples share entities in a sentence, which helps to deal with the problem of overlapping triples. Secondly, due to the hierarchical sequence labeling strategy in the RSE module, it can solve the problem of nested entities in different relation types. We will describe the details in the following sections.

3.1 Bert Encoder

We utilize BERT [11] model to incorporate information from multi-layer bidirectional Transformer [12] encoder based on language representation model. For a given word w , the input representation of BERT encoder is denoted as I_s . The generation formula is:

$$I_s = E_t + E_p \quad (2)$$

where E_t is the token embedding corresponding to w in the WordPiece embeddings with a 30,000+ vocabulary, E_p is the position embedding which indicates the position index of w in the input sentence. Note that in this work the input is a single sentence instead of sentence pair, hence the segmentation embedding as described in original BERT paper was not taken into account in Eq. 2

3.2 ReMERT Decoder

Compared with the generality of the encoder, the superiority of this model is mainly manifested in the decoding layer. The basic idea of the decoder is to extract triples in three ways. First, the RSE module detects relation-specific subjects from the input sentences. Then for each specific relation type, we obtain relational representation from the RM module. Finally, all possible objects are detected by the OE module.

Relation-Specific Subject Extractor. The RSE model takes sequence as input and captures contextual features using BERT. It is designed to distinguish all subjects and the corresponding relation types by directly decoding the encoded vector \mathbf{h}_N , which is produced by the BERT encoder. As shown in Fig. 2, the size of the vector produced by RSE module is $R \times L \times 2$. The detailed operations are as following:

$$p_{r_j,i}^{start_{sub}} = \sigma(\mathbf{W}_{start}^r \mathbf{x}_i + \mathbf{b}_{start}^r) \quad (3)$$

$$p_{r_j,i}^{end_{sub}} = \sigma(\mathbf{W}_{end}^r \mathbf{x}_i + \mathbf{b}_{end}^r) \quad (4)$$

where $p_{r_j,i}^{start_{sub}}$ and $p_{r_j,i}^{end_{sub}}$ represent the probability of identifying the i -th token as the start and end position of a subject in specific relation type r_j , relation type id $j \in (1, 2, \dots, R)$. The corresponding token will be assigned with tag 1 if the probability exceeds a certain threshold or with a tag 0 otherwise. \mathbf{x}_i is the encoder representation of the i -th token in the input sequence, where $\mathbf{W}_{(.)}^r$ is a parameter matrix and $\mathbf{b}_{(.)}$ is a bias vector to be learned during training, σ is the sigmoid activation function.

Relational Memory Module. The previous section provides all possible subjects in the input sentence and their corresponding relation types, marked as (s, r) pairs. The RM module aims to obtain the semantic representation of the relation type according to the relation id, and integrate the semantic information into the object prediction stage.

We first create a simple lookup table to store all relation embeddings. Notably, the parameters of these relation embeddings are randomly initialized and updated in the training progress. Secondly, we retrieve the corresponding relation embedding by a specific relation id, which can be expressed by the following formula:

$$\mathbf{l}_{r_j} = \mathbf{RET}(rel_j) \quad (5)$$

where rel_j is the relation id j , \mathbf{l}_{r_j} is the vector of j -th in relation embeddings table (RET). The embedding module contains R vectors with size h , where R is the number of relation types and h is the dimension of encoded hidden states.

Through the above two steps, we obtain the start and end positions of all subjects in the input sentence and their token embeddings, which are denoted as \mathbf{v}_{sub}^{start} and \mathbf{v}_{sub}^{end} respectively. The representation of the subject \mathbf{v}_{sub} is the sum of the start and end embeddings.

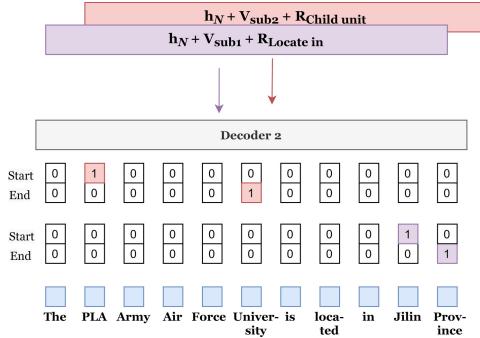
Object Extractor. After we get the subject representation \mathbf{v}_{sub_k} and its corresponding relation embedding \mathbf{R}_{r_j} , the object extractor aims to identify all objects. As the Fig. 3 shown, we utilize binary classification to determine whether a token belongs to the start or the end of an object. The detailed operation of the object extractor for each token is as following:

$$p_i^{start_{obj}} = \sigma(\mathbf{W}_{start}^{r_j}(\mathbf{x}_i + \mathbf{v}_{sub_k} + \mathbf{R}_{r_j}) + \mathbf{b}_{start}) \quad (6)$$

$$p_i^{end_{obj}} = \sigma(\mathbf{W}_{end}^{r_j}(\mathbf{x}_i + \mathbf{v}_{sub_k} + \mathbf{R}_{r_j}) + \mathbf{b}_{end}) \quad (7)$$

where $p_i^{start_{obj}}$ and $p_i^{end_{obj}}$ represent the probability of identifying the i -th token in the input sequence as the start and end position of an object, respectively. $\mathbf{W}_{(.)}^r$ represents the trainable weight, $\mathbf{b}_{(.)}$ is the bias and σ is the sigmoid activation function. For each (s, r) pair, we iteratively apply the same OE process on it. The advantage is that the overlapping objects between different triples can be solved, as long as these triples contain different subjects or relations.

Loss Function. We define the training loss (to be minimized) of ReMERT as the sum of the negative log probabilities of the true start and end labels by the predicted distributions:

**Fig. 3.** Object Extractor

$$L = -\frac{1}{n} \sum_{i=1}^n (\log P(y_i^{start} = \hat{y}_i^{start}) + \log P(y_i^{end} = \hat{y}_i^{end})) \quad (8)$$

Two learning signals are provided to train the model: L_{RSE} for RSE module and L_{OE} for OE module, both of which are formulated as Eq. 8. To share input utterance across tasks and train them jointly, for each training instance, we randomly select a relation-specific subject from the golden subject set as the specified input of the OE module. Finally, the joint loss is given by:

$$L_{loss} = L_{RSE} + L_{OE} \quad (9)$$

Then, the model is trained with gradient descent. The optimization Eq. 9 enables the extraction of subject, object, and relation influence each other, so that the loss in each component can be mutually constrained.

4 Experiments

This section describes the experimental process and best results on the public datasets. The overall experiment is compared with the baseline methods, and fine-grained analysis is performed on different complex relation types.

4.1 Experimental Settings

We use Tesla V100 GB as training station. Part of hyper-parameters are shown in Table 1. The proposed model is implemented with Pytorch and the network weights are optimized with Adam [13]. Based on the problem formulation described in Sect. 3, the proposed model actually fits a binary classifier to predict the existence of triples, which actually gives a probability for each possible token. Therefore, we set the threshold to 0.5 to differentiate the positive and negative classes.

Table 1. Hyper-parameters used for training on each dataset

Hyper-parameters	NYT	DuIE
Pretrained model	BERT-Base, cased	BERT-Base, Chinese
Batch size	8	8
Learning rate	3×10^{-5}	5×10^{-5}
Max train epochs	100	20
Max sequence length	512	256

4.2 Datasets and Metrics

We evaluate the proposed framework on two public datasets, NYT [14] and DuIE¹. NYT is an English dataset, which was originally produced by the distant supervision method. It consists of 1.18M sentences with 24 predefined relation types. In this article, we use the NYT datasets released by [5], in which contains 56195 sentences for training, 5000 sentences for validation, and 5000 sentence for test. DuIE is a Chinese dataset released by Baidu Inc. for information extraction, consisting of 210k Chinese sentences with 49 pre-specific relation categories. According to the different overlapping and nested entities patterns of relational triples, we split the sentences into four categories, namely EntityPairOverlap (EPO), SingleEntityOverlap (SEO), InnerNestedEntities (Inner-NE) and ExtraNestedEentities (Extra-NE). The statistics of the two datasets are described in Table 2. Following previous work [10], the extracted relational triple (s, r, o) is regarded as correct only if the relation and (s, o) entity-pair are all correct. For fair comparison, we report the standard Precision (Prec), Recall (Rec) and F1 score as in line with baselines.

Table 2. Statistics of the data set. Please note that a sentence can belong to both the EPO category and the SEO category. In addition, there are few examples of nested entities in NYT, and there are some in DuIE.

Category	NYT		DuIE	
	Train	Test	Train	Test
EPO	9782	978	173,084	1925
SEO	14735	1297	21,639	13331
Inner-NE	63	2	2360	287
Extra-NE	2	0	2367	290
All sentences	56195	5000	173084	21639

¹ Available at <http://ai.baidu.com/broad/download>.

4.3 Baselines

As described above, pretrained language model especially BERT, are very powerful and may cause unfair comparisons between the proposed method and the traditional. Therefore, we select some recent work, and reproduced with BERT.

NovelTagging [3] was the first framework to extract relational triples by a novel sequential tagging scheme. **GraphRel** [10] utilized graph convolutional network to extract overlapping relations by splitting entity mention pairs into several word pairs and considering all pairs for prediction. **BiTT** [15] transferred the triples with the same relation category in a sentence are especially represented as two binary trees, each of which is converted into a word-level tags sequence to label each word. **CasRel** [6] introduced a novel cascade binary tagging framework based on BERT model, which first extracted all possible subjects in a sentence then identified all possible relations and corresponding objects for each subject. **TPLinker** is the first one-stage joint extraction model that can extract all kinds of overlapping relations without the influence of exposure bias [7]. The experiment results of different baselines for relational triple extraction on two datasets are shown in Table 3.

Table 3. The main result. Bold indicates the highest score. The ♣ is directly quoted from the result of [15]. The ♠ marked result is reproduced through the official implementation.

Method	NYT			DuIE		
	Prec	Rec	F1	Prec	Rec	F1
♣ NovelTagging	89.0	55.6	69.3	75.0	38.0	50.4
♣ GraphRel	82.5	57.9	68.1	41.1	25.8	31.8
♣ BiTT	89.7	88.0	88.9	75.7	80.6	78.0
♠ CasRel	89.6	88.8	89.2	81.5	77.8	79.6
♠ TPLinker	90.3	90.6	90.4	80.8	80.4	80.6
Ours	89.2	90.2	89.7	81.5	81.3	81.4

4.4 Results and Analysis

We compared the proposed method with the baseline model in terms of quality and efficiency.

Quality. Table 3 shows the Prec, Rec, and F1 of our framework and the baseline models. In particular, the performance of ReMERT on NYT is close to state-of-the-art, and it is 0.8% higher than the state-of-the-art model on DuIE. We can also observe that ReMERT achieves a similar F1 score to CasRel and TPLinker on NYT dataset. We consider it is because: 1) Although CasRel cannot address nested entities problem, but there are almost no nested entities in

NYT dataset, so the superiority of proposed model cannot be proved. 2) The methods of exceeding 90% F1 score have already surpassed human-level performance. In other words, the room for boosting is too limited. For the comparison of their performance in dealing with nested entities, please refer to Table 4.

Table 4. The performance of the three methods on the Inner-NE and Extra-NE test sets. It is worth noting that since NYT contains almost no examples of nested entities, we only verify them on DuIE. It can be seen that whether it is Inner-NE or Extra-NE, the F1 score of ReMERT is higher than that of CasRel and TPLinker.

Method	Inner-NE			Extra-NE		
	Prec	Rec	F1	Prec	Rec	F1
CasRel	74.4	59.3	66.0	73.7	58.0	64.9
TPLinker	75.0	61.5	67.6	74.1	60.0	66.3
Ours	74.2	63.8	68.6	73.4	60.6	66.4

Efficiency. The efficiency comparison results of the three methods are shown in Table 5. We compare the efficiency of different methods in terms of the total parameters of the model, the average training time of an epoch and average inference time. Compared with CasRel, ReMERT utilizes a similar entity labeling strategy, so the two perform similar in terms of parameters and calculation time. TPLinker modeling the entity span as $L \times L$ matrix, while ReMERT is only represented by $R \times L \times 2$ matrix. The size of L is usually much bigger than R , so ReMERT has fewer parameters, and more efficient operation.

Table 5. Comparison of computational efficiency. Params all (millions) represents the number of parameters of the entire model. Training time represents the average time (h) of the model cost for each epoch. Inference time (h) represents the average time it takes for the model to predict a sample.

Model	Params all (million)	Training time (h)	Inference time (h)
Casrel	108.3	1.1	0.35
TPLinker	109.6	1.2	0.43
ReMERT	108.4	3.6	0.32

5 Conclusion

In this paper, we proposed a new extraction framework called ReMERT to solve the problem of overlapping triples, nested entities and insufficient representation of relation types. The experimental results show that the proposed model outperforms all baselines and achieves a new state-of-the-art on the DuIE datasets.

In the future, we plan to build a large Chinese RE dataset containing substantial complex relational triples to better verify the superiority of our model and explore its performance on other information extraction tasks.

Acknowledgements. We thanks anonymous reviewers for their precious comments. This research is supported by the Postgraduate Research & Practice Innovation Program of Jiangsu Province (Grant SJCX21_0989) and Smart Mining Open Funding Project of Shandong Energy Zibo Mining Group & China University of Mining and Technology (Grant 2019LH10).

References

1. Zhou, G., Su, J., Zhang, J., Zhang, M.: Exploring various knowledge in relation extraction. In: ACL, Meeting of the Association for Computational Linguistics, Conference, June, University of Michigan, USA (2002)
2. Chan, Y.S., Roth, D.: Exploiting syntactico-semantic structures for relation extraction. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 551–560 (2011)
3. Zheng, S., Wang, F., Bao, H., Hao, Y., Zhou, P., Xu, B.: Joint extraction of entities and relations based on a novel tagging scheme. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1227–1236. Association for Computational Linguistics, Vancouver, July 2017
4. Miwa, M., Bansal, M.: End-to-end relation extraction using LSTMs on sequences and tree structures (2016)
5. Zeng, X., Zeng, D., He, S., Kang, L., Zhao, J.: Extracting relational facts by an end-to-end neural model with copy mechanism. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (2018)
6. Wei, Z., Su, J., Wang, Y., Tian, Y., Chang, Y.: A novel cascade binary tagging framework for relational triple extraction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 1476–1488. Association for Computational Linguistics, July 2020
7. Wang, Y., Yu, B., Zhang, Y., Liu, T., Zhu, H., Sun, L.: TPLinker: single-stage joint extraction of entities and relations through token pair linking. arXiv preprint [arXiv:2010.13415](https://arxiv.org/abs/2010.13415), 2020
8. Li, C., Tian, Y.: Downstream model design of pre-trained language model for relation extraction task (2020)
9. Li, Q., Ji, H.: Incremental joint extraction of entity mentions and relations. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 402–412 (2014)
10. Fu, T.-J., Li, P.-H., Ma, W.-Y.: GraphRel: modeling text as relational graphs for joint entity and relation extraction. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1409–1418. Association for Computational Linguistics, Florence, July 2019
11. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Pre-training of deep bidirectional transformers for language understanding, BERT (2018)
12. Vaswani, A.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30, pp. 5998–6008 (2017)

13. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. *Comput. Sci.* (2014)
14. Riedel, S., Yao, L., Mccallum, A.K.: Modeling relations and their mentions without labeled text. In: Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD, Barcelona, Spain, 20–24 September 2010, Proceedings. Part II **I**, 2010 (2010)
15. Luo, X., Liu, W., Ma, M., Wang, P.: A bidirectional tree tagging scheme for jointly extracting overlapping entities and relations (2020)



Recognition of Nested Entity with Dependency Information

Yu Xia^{1,2} and Fang Kong^{1,2(✉)}

¹ Institute of Artificial Intelligence, Soochow University, Suzhou, China
20205227072@stu.suda.edu.cn, kongfang@suda.edu.cn

² School of Computer Science and Technology, Soochow University, Suzhou, China

Abstract. Named entity recognition (NER) is a basic task in natural language processing. However, most existing models are hard to detect entities with nested structure which means that an entity contains one or more entities. In this paper, we propose a boundary-aware approach for nested NER. First, word information is incorporated in the same dimension via Lexicon, in which characters are feed into LSTM to learn internal structure of words and obtain character representation. To augment word representation, Graph Convolutional Network (GCN) is applied to extract dependency information between entities. Second, our model can detect boundaries to locate entity by using Star-Transformer, which is suitable for small-scale corpus and unstructured texts because of its star structure. Based on predicted boundaries, our model utilizes boundary-aware regions to predict entity categorical labels, which can reduce the number of candidate entities and decrease computation cost. In our experiment, it shows an impressive improvement on forum corpus and that our model can perform well on a small-scale corpus.

Keywords: Nested NER · Dependency information · Star-transformer

1 Introduction

Named entity recognition (Luo et al. [7]) aims to identify entities with specific meanings from texts such as Person, Organization, etc. The previous solution is to regard NER as a sequence labeling task where each token is tagged with a boundary label and a categorical label. For example, a token can be tagged with *E-Location*, where *E* indicates the end of an entity and *Location* indicates the corresponding entity categorical label.

LSTM-CRF model (Huang et al. [3], Ma et al. [9]) includes bidirectional LSTM and a Conditional Random Field layer, which achieves promising results. However, it is difficult for LSTM to have efficient parallel computing capabilities owing to its own sequence-dependent structure. Transformer (Vaswani et al. [12]) makes up for the shortcomings of low parallelism of LSTM by attention mechanism. Fully connected attention mechanism means that each token establishes a direct connection with other tokens, thus capture long-range features. However,

What fully-connected structure brings is that there are many parameters and a large amount of training data is required. Owing to small scale of forum corpus, it is easy to overfit to the training data by utilizing Transformer.

Jia et al. [4] propose a character-level model for NER. Compared to NER in English, Chinese NER is more difficult since sentences have vague boundaries, thus leading to that segmentation is a little rough and can't find more detailed semantic information. Therefore, some approaches resort to performing Chinese NER at character level. Models based on fine-grained entity recognition can reduce memory and time complexity. However, using characters only distorts the meaning of words and has a decrease in model performance. Additionally, the word-level model (He et al. [2]) splits sentences into words, which can make semantic information relatively complete. However, the size of vocabulary is very large, thus increasing the memory and time complexity.

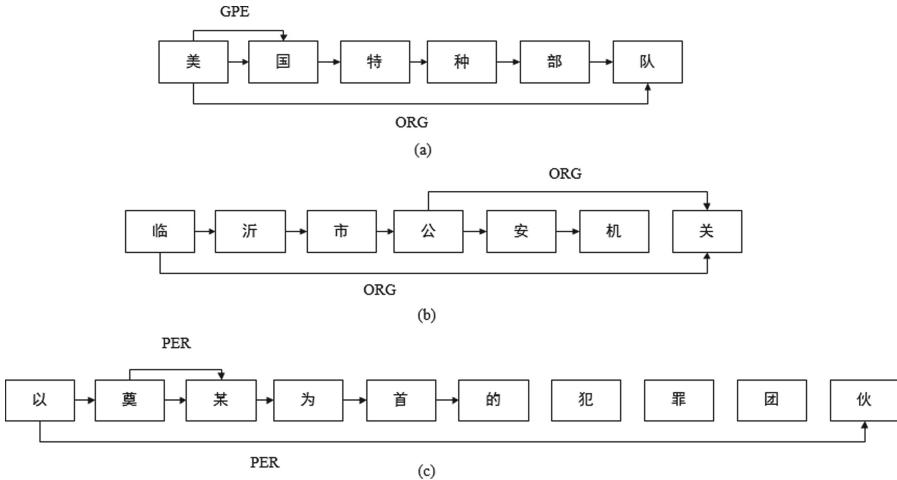
2 Motivation

In this paper, we perform Named entity recognition (NER) on the forum dataset. First, the scale of this dataset is small. The statistics are shown in Table 1. We can see that forum corpus contains 157 documents and 4808 sentences. Entity types consist of '*PER*', '*ORG*', '*GPE*', '*LOC*' and '*TITLE*'. Second, some entities in this dataset are nested and percentage of nested entities is high, even the percentage of nested entities belong to '*TITLE*' is up to 80%. Third, Examples shown in Fig. 1 denote that there are different kinds of ways to nest. Some entities share the same beginning and some entities share the same end. However, the traditional sequence labeling task can't recognize nested entities, because it does not support assigning multiple tags to a token.

To handle this problem, we introduce a boundary-aware neural model (Zheng et al. [13]) as our baseline system, which leverages entity boundaries to predict entity categorical labels. Based on it, we propose an approach to capture non-local and local information and incorporate dependency information. First, motivated by Lexico (Ma et al. [8]), bigram embedding is used for augmenting character representation and incorporating word information. Second, we utilize GCN (Kipf et al. [6]) to model dependency information between entities. Third, owing to the small-scale dataset, Star-Transformer (Guo et al. [1]) is used for encoder instead of LSTM which can prevent overfitting and achieves good performance in unstructured texts. Last, in terms of decoder, we apply a single-layer sequence labeling model to identify entity boundaries because tokens in nested entities can share the same boundary labels which can be seen in Fig. 1. Based on the detected boundaries, we match each tokens with label *B* to tokens with label *E*. The spans between them are considered as candidate entities. The representation extracted by Star-Transformer of candidate entities are used to classify categorical label. Also, considering that two tasks share the same entity boundaries, we use joint learning for training these tasks simultaneously, which can capture the connection of boundary detection and label classification. Star-Transformer with GCN based on character and word information proposed in

Table 1. The statistics of forum dataset.

	Train	Dev	Test	All	Nested	Percentage
Documents	97	40	20	157	–	–
Sentences	3048	1145	615	4808	–	–
PER	5500	1869	1127	8496	2032	23.92%
GPE	3284	1418	606	5308	2226	41.94%
ORG	1167	369	150	1686	891	52.85%
LOC	755	314	112	1181	499	42.25%
TITLE	202	100	23	325	261	80.31%
All	10908	4070	2018	16996	5909	34.77%

**Fig. 1.** Examples of nested entities in forum texts. *B*, *I*, *E* represent the start, middle and end of the entity. *PER*, *ORG* and *GPE* are categories of entities.

this paper shows an impressive improvement on forum corpus and can recognize different kinds of nested entities.

3 Model

Following the overview in Fig. 2, our model consists of two parts as boundary detection and categorical labels prediction. The boundary detection part aims to predict whether one token is first or last word of an entity to locate candidate entities. Another task aims to predict the categorical labels of candidate entities.

3.1 Word Representation

For a character-level NER model, the input sentence is seen as a character sequence $s = [c_1, c_2, \dots, c_n]$, where c_i represents the i -th character and n is the

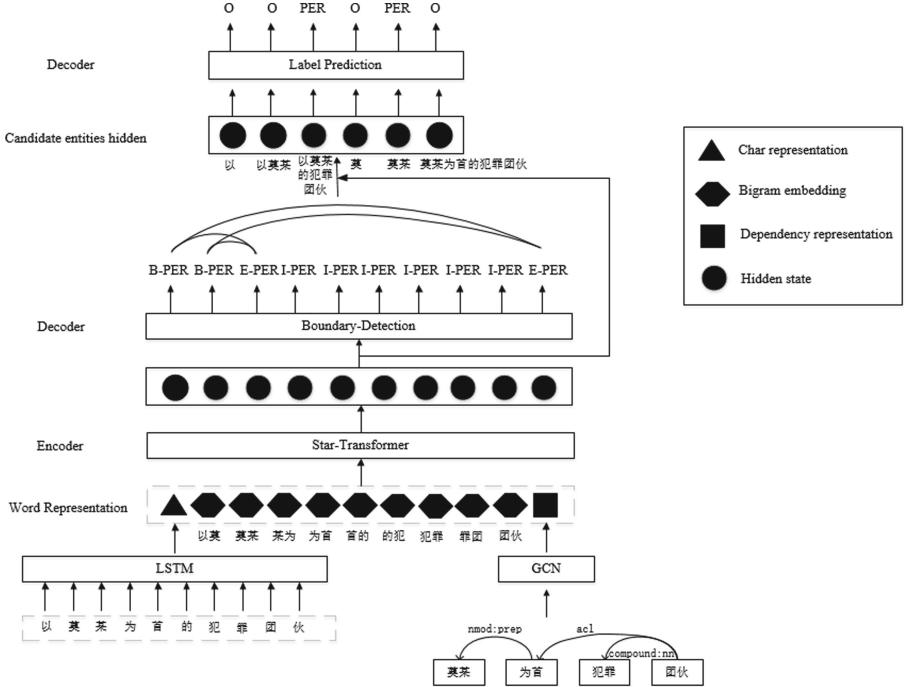


Fig. 2. The general structure of our model. Char embedding is feed into LSTM layer. We concentrate char embedding, word embedding and dependency information modeled by GCN and feed them into Star-Transformer to extract shared feature for detecting the boundaries of entities and categorical prediction.

source length. We use a character embedding layer to represent each character, where d_c is the dimension of character embedding vector. However, this model is often accompanied by a decrease in model performance. The useful of character bigrams has been proved for representing characters especially for methods not using word information. Therefore, we concentrate character embedding and bigram embedding as Eq. (1):

$$x_c^i = [E_c(c_i); E_b(c_i, c_{i+1})] \quad (1)$$

where d_b is the dimension of bigram embedding vector, E_b denotes the bigram embedding lookup table, E_c denotes the character embedding lookup table.

To augment word representation, GCN is applied to model dependency information between entities. First, a sentence can be parsed to a dependency tree. In Fig. 2, we can see that dependency relation of the sentence contains ‘*nmod:prep*’, ‘*acl*’ and ‘*compound:nn*’. We can regard dependency trees as a kind of directed graph. Given a graph with n nodes, one node represents one bigram, the dependency graph can be represented with an $n \times n$ adjacency matrix A where $A_{ij} = 1$ if there is a dependency relation between bigram i and bigram j . In an L -layer

GCN, if we denote by h_i^{l-1} the input vector and h_i^l the output vector of node i at the l -th layer, a graph convolution operation can be written as Eq. (2)

$$h_i^l = \sigma\left(\sum_{j=1}^n A_{ij} W^l h_j^{l-1} + b^l\right) \quad (2)$$

where W^l is a linear transformation, b^l is a bias term, and σ is a nonlinear function, h_i^l is output of dependency representations. Intuitively, during each graph convolution, each node gathers and summarizes information from its neighboring nodes.

After applying an L -layer GCN over word vectors, we obtain dependency representations, the final word representation can be constructed by concentrating character embedding, bigram embedding and dependency representations.

3.2 Encoder

Our encoder uses a Star-Transformer to obtain context-sensitive hidden state. As shown in Fig. 3, the fully-connected attention-based model like Transformer is not suitable for small-scale corpus because this kind of model needs lots of parameters and is easy to overfit to small-scale dataset. Therefore, Star-Transformer is utilized in which complexity can be reduced from quadratic to linear. Star-Transformer consists of a relay node and n satellite nodes, where a relay node is used for gathering and absorbing knowledge from all the satellite nodes. The state of i -th satellite node represents the features of i -th character in a text sequence. Each radical connection links a satellite node to a relay node. Thus, every two non-adjacent satellite nodes can receive non-local information with a two-step update. Each ring connection links a satellite node to its adjacent satellite nodes, thus making it easy to capture local information. With two kinds of connections, Star-Transformer can capture local and long-range compositions simultaneously. The implementation of Star-Transformer is also based on the attention mechanism.

Given a sequence of word representation H , we can use a query vector q to select the relevant information with attention as Eq. (3):

$$\text{Attention}(q, K, V) = \text{softmax}\left(\frac{qK^T}{\sqrt{d}}\right)V \quad (3)$$

Where $K = HW^K$, $V = HW^V$ and W^K, W^V are learnable parameters.

To gather more useful information from H , we can use multi-head attention with k heads as Eq. (4).

$$\text{Attention}(q, K, V) = (a_1 \oplus \dots \oplus a_k)W^o \quad (4)$$

$$a_i = \text{Attention}(qW_i^Q, HW_i^K, HW_i^V) \quad (5)$$

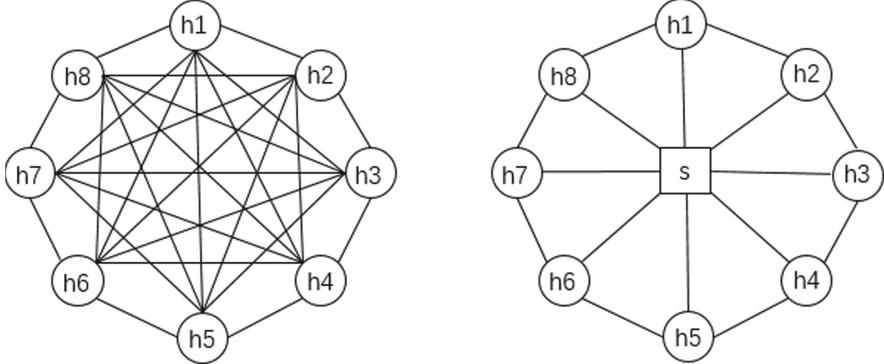


Fig. 3. The method of node connection of Transformer and Star-Transformer

Where \oplus denotes concatenation operation, and $W_i^Q, W_i^K, W_i^V, W_i^O$ are learnable parameters.

Let $s^t \in R^{1 \times d}$ and $H^t \in R^{n \times d}$ denote the states of the relay node and all the n satellite nodes at time step t . During encoding word representation, we start from initializing the state with $H^0 = e$ and $s^0 = \text{average}(e)$, where e denotes word representation.

The update can be divided into the update of satellite node and the update of relay node. First, the update of each satellite node is updated from its adjacent nodes h_{i-1}, h_{i+1} , previous state of relay node s^{t-1} and its correspond embedding e^i as Eq. (6), (7), (8).

$$C_i^t = [h_{i-1}^{t-1}, h_i^{t-1}, h_{(i+1)}^{(t-1)}, e^i, s^{t-1}] \quad (6)$$

$$h_i^t = \text{MultiAttention}(h_i^{t-1}, C_i^t) \quad (7)$$

$$h_i^t = \text{LayerNorm}(\text{ReLU}(h_i^t)) \quad (8)$$

Second, the update of relay node s^t gathers the information of all satellite nodes and its previous relay node state as Eq. (9), (10).

$$s^t = \text{MultiAttention}(s^{t-1}, [s^{t-1}; H^t]) \quad (9)$$

$$s^t = \text{LayerNorm}(\text{ReLU}(s^t)) \quad (10)$$

3.3 Decoder

The decoder of our model consists of two subtasks as boundary detection and categorical labels prediction. Different from assigning an entity categorical label to each token, we predict the boundary first.

Entity Boundary Detection. Boundary detection aims to predict start and end location of an entity. The above hidden state extracted by Star-Transformer is feed into a classifier:

$$O_i = W^{start} \cdot h_i + b^{start} \quad (11)$$

$$P_i = \text{softmax}(O_i) \quad (12)$$

Where P_i denotes the probability of character being boundary of an entity. Based on the classifier, we can predict boundary labels. Boundary labels consist of ‘B’ denoting the beginning of the entity, ‘I’ denoting the middle of the entity, ‘E’ denoting the end of the entity and ‘O’ denoting non-entity.

Boundary-Aware Entity Categorical Label Prediction. Our assembling strategy is that every start boundary is matched with all end boundaries after it. Based on P_i , each token can be assigned with labels ‘B’, ‘I’, ‘E’ and ‘O’. Given hidden state extracted by Star-Transformer $H = (h_1, h_2, \dots, h_n)$ and its corresponding predicted boundary label sequence $L = (l_1, l_2, \dots, l_n)$, we match each token ‘B’ with label ‘E’ to construct candidate entity regions. Especially, considering that there are entities containing one single token, we match tokens with label ‘B’ to themselves firstly.

The representation of candidate entities $R_{i,j}$ is obtained as following:

$$R_{i,j} = \frac{1}{j-i+1} \sum_{k=i}^j h_k \quad (13)$$

where h_k denotes the output of encoder. For example, the predicted boundary label sequence is $L = (B, B, E, I, I, I, I, I, I, E)$, the candidate entities are $R_{0,0}, R_{0,2}, R_{0,9}, R_{1,1}, R_{1,2}, R_{1,9}$.

Our model only classifies boundary-relevance regions $R_{i,j}$, thus reducing the number of possible regions. $R_{i,j}$ which represents features of candidate entity is feed into a classifier. The number of categorical labels is six, which contains ‘PER’, ‘ORG’, ‘LOC’, ‘TITLE’, ‘GPE’ and ‘None’.

$$type_{i,j} = U \cdot R_{i,j} + b \quad (14)$$

$$t_{i,j} = \text{softmax}(type_{i,j}) \quad (15)$$

3.4 Loss Function

In our model, we predict categorical labels based on predicted entity boundaries. Therefore, two tasks share the same entity boundaries, we apply a multitask loss for training the two tasks simultaneously. The multitask loss is defined as follows:

$$Loss_{boundary} = - \sum (\hat{P}_i) \log(P_i) \quad (16)$$

$$Loss_{labels} = - \sum (\hat{t}_{i,j}) \log(t_{i,j}) \quad (17)$$

$$Loss_{multi} = \alpha \sum Loss_{boundary} + (1 - \alpha) \sum Loss_{labels} \quad (18)$$

where \hat{P}_i and P_i denote predicted entity boundaries and ground-truth boundary labels, $\hat{t}_{i,j}$ and $t_{i,j}$ denote true distribution and predicted distribution of entity categorical labels. α is a hyper-parameter which is assigned to control the degree of importance for each task.

4 Experiment

4.1 Experimental Settings

In this paper, we use Star-Transformer with GCN based on character and word information on the forum corpus which is a publicly available data for Nested Named Entity Recognition. We set different parameters to conduct the experiment. Also, we discuss and analyze results, and precision, recall, F1 are used to evaluate results.

- **Parameter Setting:** Boundary detection and label prediction share the same entity boundaries, we apply a multitask loss for training these tasks simultaneously. α is a hyper-parameter used for controlling the degree of importance for each task. In this mode, we set α to 0.3. In terms of clip norm, gradient clipping is generally used to solve the problem of gradient explosion which occurs frequently in the process of training LSTM, so set clip norm for gradient clipping. The parameters of our model are shown in Table 2.
- **Baseline:** We compare our model with the following methods: Sohrab and Miwa (2018) propose an exhaustive region classification model which enumerates all the possible regions or spans in the sentence to map to a given set of tags. Ju et al. (2018) propose a hierarchical approach to predict the representation of the token. Zheng et al. (2019) propose LSTM for extracting shared features for entity boundary detection and categorical label prediction. The results are shown in Table 3.
- **Evaluation Metrics:** Only when entity boundary and categorical label are correct simultaneously, an entity is confirmed correct. The metrics are precision, recall and F-score to evaluate the performance.

No matter what granularity, three metrics of Star-Transformer which acts as encoder are higher than LSTM. In compared to LSTM which can not capture long distance features, Star-Transformer can capture non-local and local compositions based on attention mechanism, which leads to identifying more entities. Additionally, overfitting will not happen on this small-scale dataset because structure of Star-Transformer is star topology, whose complexity is reduced from quadratic to linear.

Forum texts have vague word boundary and belong to unstructured texts. Thus, the model on this dataset needs to incorporate word information to make semantic information relatively complete. By concentrating character embedding and bigram embedding, vocabulary is relatively reasonable and our model is able to learn meaningful and context representation without high memory and time complexity during calculation of embedding. In Table 3, we can observe that performances of three character-level models are better by incorporating into word information.

Table 2. The parameters of our model.

Parameter	Value	Parameter	Value
Char embedding size	200	Dropout	0.5
Word embedding size	200	Learning rate	0.0005
Hidden size	200	Batch size	20
Multi head	5	Clip norm	5
Star layer	4	∞	0.3
GCN layer	2	Optimizer	Adam

Table 3. The results of our model.

Baseline	Level	Precision	Recall	F1
Sohrab and Miwa (2018)	Char	75.03%	61.94%	67.86%
Ju et al. (2018)	Char	75.77%	64.05%	69.41%
Zheng et al. (2019)	Char	73.78%	66.50%	69.95%
LSTM	Char	73.78%	66.50%	69.95%
Transformer	Char	69.52%	68.58%	69.08%
Star-transformer	Char	75.40%	69.87%	72.53%
Star-transformer + GCN	Char	71.71%	65.06%	68.23%
LSTM	Word	69.73%	65.06%	67.32%
Transformer	Word	61.41%	61.60%	61.50%
Star-transformer	Word	75.50%	65.91%	70.46%
Star-transformer + GCN	Word	75.20%	64.17%	69.25%
LSTM	Char + word	71.76%	70.27%	71.01%
Transformer	Char + word	70.68%	63.18%	66.72%
Star-transformer	Char + word	74.04%	70.52%	72.23%
Star-transformer + GCN	Char + word	74.44%	71.16%	72.76%

By comparison to Transformer, three metrics of Star-Transformer which acts as encoder are higher than Transformer. Although Transformer has a fully connected attention mechanism and captures long-range features, Transformers needs more parameters and training data than Star-Transformer. Therefore, performance on small datasets of Star-Transformer is better than Transformer instead owing to overfitting to the training data by utilizing Transformer.

Additionally, our model applies GCN to model dependency between entities, thus augmenting word representation and incorporating dependency information. By concentrating bigram embedding, character embedding and dependency representation modeled by GCN, final performance of our model is better than other models.

4.2 Performance of Decoder

Our decoder consists of two subtasks: boundary detection and categorical label prediction. Table 4 shows performance of boundary detection on forum dataset.

Table 4. The Performance of Our Decoder.

Model	Decoder	Precision	Recall	F1
Baseline	Boundary detection	80.62%	76.73%	78.63%
	Categorical prediction	83.32%	83.89%	83.60%
Our model	Boundary detection	82.88%	78.05%	80.39%
	Categorical prediction	85.49%	82.90%	84.18%

Table 5. The performance of entity recognition.

TYPE	Model	Gold	Predict	Correct	Accuracy
PER	Baseline	1127	1064	747	66.28%
	Our model	1127	1081	807	71.61%
ORG	Baseline	150	110	61	40.67%
	Our model	150	146	81	54%
GPE	Baseline	606	521	452	74.59%
	Our model	606	564	471	77.72%
LOC	Baseline	112	112	72	64.29%
	Our model	112	124	73	65.16%
TITLE	Baseline	23	26	8	34.78%
	Our model	23	14	4	17.39%

Our model locates entities more precisely than compared methods. The performance of boundary detection has a great influence on our model. Improving performance of this task impacts on our model: it generates more true candidate entities including nested entities, and reduces influence of false entities. Meanwhile, Star-Transformer is suitable for encoding on this forum dataset, thus being effective to support boundary detection task.

Based on predicted entity boundaries, we assemble boundaries into candidate entities. Every start boundary is matched with all end boundaries after it. We use a single-layer sequence labeling model to recognize entity type. We can observe that categorical label classifier improves precision but hurts recall, even so, final performance is increased.

4.3 Discussion

Table 5 shows the performances of our model on the five categories on the test dataset. *Predict* denotes the number of entities predicted and *Correct* denotes the number of entities predicted correctly. By comparing the number of these metrics, for example, in terms of entity type *LOC*, the number of entities predicted is 124 and the number of true entities is 112, we can observe that Star-Transformer leads to over-identification, which means that it will recognize words that are not entities as entities, thus reducing precision.

Table 6 shows the recognition of flat entities which are not nested. *Non-Nested* denotes the number of true flat entities and *Correct* denotes the number of entities predicted correctly. We can observe that the precision of most categorical labels prediction is higher than baseline. Table 6 also shows the performance of recognizing nested entities. *Nested* denotes the number of true nested entities. We also observe better performance of our model. However, the correct number of nested entities accounts for 42.6% on average of all nested entities.

Table 6. The Comparison of Entity Recognition.

TYPE	Model	Nested	Correct	Accuracy	Non-nested	Correct	Accuracy
PER	Baseline	279	113	40.50%	848	634	74.76%
	Our model	279	120	43.01%	848	687	81.01%
ORG	Baseline	72	20	27.78%	78	41	52.56%
	Our model	72	30	41.67%	78	51	65.38%
GPE	Baseline	236	120	50.85%	379	332	87.60%
	Our model	236	138	58.47%	379	333	87.86%
LOC	Baseline	49	21	42.86%	63	51	80.95%
	Our model	49	25	51.02%	63	48	76.19%
TITLE	Baseline	15	2	13.33%	8	5	62.5%
	Our model	15	3	20%	8	1	12.5%

If an entity has the following nesting condition: one character is both the beginning of one entity and the end of another entity, the performance of our model on this condition needs improved. Because sequence labeling model is applied to detect boundary and can not assign multiple tags to one character.

5 Related Work

A layered sequence labeling model (Ju et al. [5]) is proposed by using a hierarchical approach to predict the representation of the token. However, it has higher learning requirements for the decoder. If there is a misjudgment in the previous iteration process, this problem may be passed on to the subsequent iteration process. Sohrab et al. [10] propose an exhaustive region classification model which enumerates all the possible regions or spans in the sentence to map to a given set of tags. However, It brings about the following problems: high time and space complexity, difficult classifier training, lots of negative samples and so on.

Despite their shortcomings, these problems can be mitigated by some manual rules or settings. Before classification, one or more classifier can be trained to filter out batches of negative samples. Our model inspired by the boundary-aware neural model [11,13]. Our model only predicts entity categorical based on detected entity boundaries, thus decreasing the number of possible regions and time cost.

6 Conclusion

In this paper, we propose a boundary-aware model for nested NER on the forum dataset. We use Star-Transformer with GCN based on character and word information as our model. First, word information is incorporated by concentrating bigram embedding and character embedding. In order to enhance word representation, GCN is applied to extract dependency information between entities. Second, owing to the small-scale dataset, Star-Transformer is used for encoder instead of LSTM which can prevent overfitting and achieves good performance in unstructured texts. Last, we utilize shared features extracted by Star-Transformer to predict entity boundaries. Based on predicted boundaries, our model matches every start boundary with all end boundaries to predict entity categories. We train two tasks simultaneously by a multitask loss.

Acknowledgements. This work was supported by Project 61876118 under the National Natural Science Foundation of China and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

1. Guo, Q., Qiu, X., Liu, P., Shao, Y., Xue, X., Zhang, Z.: Star-transformer. arXiv preprint [arXiv:1902.09113](https://arxiv.org/abs/1902.09113) (2019)
2. He, J., Wang, H.: Chinese named entity recognition and word segmentation based on character. In: Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing, pp. 128–132 (2008)
3. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint [arXiv:1508.01991](https://arxiv.org/abs/1508.01991) (2015)
4. Jia, Y., Ma, X.: Attention in character-based BiLSTM-CRF for Chinese named entity recognition. In: ICMAI 2019 (2019)
5. Ju, M., Miwa, M., Ananiadou, S.: A neural layered model for nested named entity recognition. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 1446–1459 (2018)
6. Kipf, N.T., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint [arXiv:1609.02907](https://arxiv.org/abs/1609.02907) (2016)
7. Li, J., Sun, A., Han, J., Li, C.: A survey on deep learning for named entity recognition. arXiv preprint [arXiv:1812.09449](https://arxiv.org/abs/1812.09449) (2020)
8. Ma, R., Peng, M., Zhang, Q., Wei, Z., Huang, X.: Simplify the usage of lexicon in Chinese NER. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 5951–5960 (2020)
9. Ma, X., Hovy, E.: End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1064–1074 (2016)
10. Sohrab, M.G., Miwa, M.: Deep exhaustive model for nested named entity recognition. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2843–2849 (2018)
11. Tan, C., Qiu, W., Chen, M., Wang, R., Huang, F.: Boundary enhanced neural span classification for nested named entity recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 9016–9023 (2020)

12. Vaswani, A., et al.: Attention is all you need. arXiv preprint [arXiv:1706.03762](https://arxiv.org/abs/1706.03762) (2017)
13. Zheng, C., Cai, Y., Xu, J., Leung, H.F., Xu, G.: A boundary-aware neural model for nested named entity recognition. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 357–366 (2019)



HAIN: Hierarchical Aggregation and Inference Network for Document-Level Relation Extraction

Nan Hu, Taolin Zhang, Shuangji Yang, Wei Nong, and Xiaofeng He^(✉)

School of Computer Science and Technology, East China Normal University,
Shanghai, China

{51194501046, 52184501016, 51194501201, 51194501160}@stu.ecnu.edu.cn,
hexf@cs.ecnu.edu.cn

Abstract. Document-level relation extraction (RE) aims to extract relations between entities within a document. Unlike sentence-level RE, it requires integrating evidences across multiple sentences. However, current models still lack the ability to effectively obtain relevant evidences for relation inference from multi-granularity information in the document. In this paper, we propose **Hierarchical Aggregation and Inference Network** (HAIN), performing the model to effectively predict relations by using global and local information from the document. Specifically, HAIN first constructs a *meta dependency graph* (mDG) to capture rich long distance global dependency information across the document. It also constructs a *mention interaction graph* (MG) to model complex local interactions among different mentions. Finally, it creates an *entity inference graph* (EG), based on which we design a novel hybrid attention mechanism to integrate relevant global and local information for entities. Experimental results demonstrate that our model achieves superior performance on a large-scale document-level dataset (DocRED). Extensive analyses also show that the model is particularly effective in extracting relations between entities across multiple sentences and mentions.

Keywords: Document-level relation extraction · Graph neural network

1 Introduction

Relation extraction (RE) aims to identify semantic relations between entities from plain text. With the growing demand for structured knowledge, RE has attracted much attention in natural language processing. Prior works have made great progress in extracting relations within a sentence (sentence-level RE). However, in real world scenarios, a large number of relation instances appear across sentences. Compared with sentence, a document often contains many entities, and some entities have multiple mentions under the same phrase of alias. Hence, document-level RE is a more complex relation extraction problem.

Figure 1 shows an example of document-level RE. Early studies [10,12] defined document-level RE to short text spans (e.g., document only contains two sentences). Some other studies were limited to specific domain (e.g., biomedicine). It's obviously that they are incapable of dealing with the example in Fig. 1. Recent works [1,7,9] used graph-based neural approaches, since graph has proven useful in encoding long distance, cross-sentential information. They mainly put different types of nodes in a same graph and then applied vanilla GCNs [6] to jointly update nodes. However, current models do not in-depth explore a reasonable graph aggregation and inference structure which is critical to model's understanding of the entire document.

<p>[1] <i>Michael Helm</i> is a <u>Canadian</u> novelist . [2] <i>He</i> was born in <u>Eston</u> , <u>Saskatchewan</u> , and received degrees in literature from <u>the University of Saskatchewan</u> and <u>the University of Toronto</u> . [3] His debut novel , <u><i>The Projectionist (1997)</i></u> , was nominated for <u>the Giller Prize</u> and the <u><i>Trillium Book Award</i></u> . [4] His second novel , <u><i>In the Place of Last Things (2004)</i></u> was a finalist for regional <u>Commonwealth Prize</u> for Best Book and the Rogers Writers ' Trust Fiction Prize ... [7] Helm currently teaches in the ...</p>	
Head Entity: <i>Michael Helm</i>	Reasoning type: Logical reasoning
Tail Entity: <i>Trillium Book Award</i>	Relation type: Inter-sentence relation
Relation: Award Received	Supporting sentence: 1, 3

Fig. 1. An example from the DocRED [20] dataset. Entities and mentions involved in the relation instance (*Michael Helm*, *Award Received*, *Trillium Book Award*) are colored. Other irrelevant mentions are underlined for clarity (best viewed in color).

From our point of view, as Fig. 1 shows, in order to extract the relation between *Michael Helm* and *Trillium Book Award*. Firstly, we should identify sentence 1 and 3 are supporting sentences that contain the global context information about *Michael Helm* and *Trillium Book Award*. Then, identify *Michael Helm* is a novelist from sentence 1, *The Projectionist(1997)* is a novel written by *Michael Helm* and nominated for *Trillium Book Award* from sentence 3. Finally, we can infer that *Michael Helm* received *Trillium Book Award*. Obviously, it's a step by step inference behavior, multi-granularity information is aggregated from coarse to fine (document → mention → entity). But the supporting sentences are scattered in the document, relevant mentions usually don't appear in the same sentence, and entities need long distance dependency information.

In this paper, we propose a novel graph-based network for document-level RE. Our primary motivation is to design a hierarchical aggregation and inference structure that can do document-level RE as the above intuitive example. Towards this goal, we address three challenges: (1) *how to capture long distance dependency information of a document?* Syntactic dependency tree conveys rich structural information that is proven useful for many sentence-level RE models [4,23]. We extend it to document-level, and build a meta dependency graph (mDG) that can utilize structural knowledge to capture long distance global dependency information of a document. (2) *how to model complex local information of mentions?* We construct a mention interaction graph (MG) to capture

local information by mention interactions. Concretely, we merge the initial representations of mentions from mDG, build MG by self-attention mechanism [17] and then apply GCN [6] to encode MG. (3) *how to learn entity representations effectively?* We build an entity inference graph (EG) and design a novel hybrid attention mechanism to encode global and local information from mDG and MG into entities.

Our main contributions can be summarized as follows:

1. We propose a Hierarchical Aggregation and Inference Network (**HAIN**), which features a hierarchical graph design, to better cope with document-level RE task.
2. We introduce three different graphs to meet the needs of different granularity information. A novel hybrid attention mechanism is proposed to effectively aggregate global and local information for entities.
3. HAIN achieves new state-of-the-art performance on DocRED dataset. Our detailed analysis further shows its superior advantage in extracting relations between entities of long distance.

2 Methodology

2.1 Model Overview

Given a document $D = [x_1, x_2, \dots, x_n]$, where $i \in [1, n]$ and x_i is the i -th word in document. Sentences, entities and their corresponding textual mentions are annotated in the document. The set of relation types is pre-defined. Our goal is to identify the relations of all entity pairs in the document. Obviously, it is a multi-label classification problem.

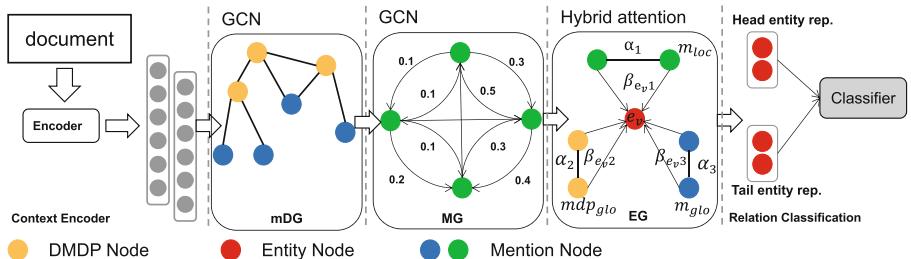


Fig. 2. Architecture of HAIN. Some nodes are omitted for simplicity. MG is a fully connected graph with learned edge weight from 0.0 to 1.0. In EG, $\alpha_i, \beta_{e_v i}$ are type and node attention scores of entity e_v calculated by hybrid attention mechanism. m_{loc} is mention nodes representations learned from MG, m_{glo}, mdp_{glo} are mention and DMDP nodes representations learned from mDG.

Figure 2 depicts the architecture of our HAIN. (1) First, it uses LSTM [14] or BERT [2] as encoder to receive an entire document with annotations as input

and output the contextual representation of each word. (2) Next, it constructs a *meta dependency graph* (mDG) by using the dependencies of the syntactic dependency tree. It also creates a *mention interaction graph* (MG) by self-attention mechanism [17]. mDG and MG graphs are encoded by using stacked GCN [6] to respectively capture global and local information of the document. (3) Then, a novel hybrid attention mechanism is designed to integrate relevant global and local relation inference information into entities in entity inference graph (EG). (4) Finally, it uses entities representations learned from EG to predict relations.

2.2 Context Encoder

To obtain the contextual representation of each word, we feed a document D into a contextual encoder. The context encoder can be a bidirectional LSTM [14] or BERT [2]. Here we use the BiLSTM as an example:

$$\overleftarrow{h_{w_j}} = \text{LSTM}(\overleftarrow{h_{w_{j+1}}}, \gamma_j) \quad (1)$$

$$\overrightarrow{h_{w_j}} = \text{LSTM}(\overrightarrow{h_{w_{j-1}}}, \gamma_j) \quad (2)$$

where $\overleftarrow{h_{w_j}}$ and $\overrightarrow{h_{w_j}}$ represent the hidden representations of the j -th word in the document of two directions, γ_j indicates the word embedding of the j -th word. Finally, the contextual representation of each word in the document is represented as $h_{w_j} = [\overleftarrow{h_{w_j}}; \overrightarrow{h_{w_j}}]$.

2.3 Meta Dependency Graph

Based on the contextual representation of each word, we extract document meta dependency path nodes (DMDP) and mention nodes to construct *meta dependency graph*. The initial representation of a mention node \mathbf{m}_i is calculated by averaging the representations of contained words (e.g., $h_{m_i} = [avg_{w_j \in m_i}(h_{w_j})]$). Early approaches [4, 13] used all nodes in the syntactic dependency tree of a sentence. Nan et al., [9] just extracted nodes on the shortest dependency path (SMDP) between mentions in the sentence, as it is able to make full use of relevant information while ignoring irrelevant information. We extend it to DMDP by connecting root nodes of each sentence dependency tree in a document.

As Fig. 3 shows, given four mentions m_1, m_2, m_3, m_4 in two sentences s_1, s_2 of document D, and $m_1, m_2 \in s_1, m_3, m_4 \in s_2$. SMDP just extracts MDP_{m_1, m_2} and MDP_{m_3, m_4} as nodes. But our DMDP extracts MDP_{m_i, m_j} , $i, j \in 1, 2, 3, 4$ and $i \neq j$ as nodes, which will contain more inter-sentential information.

We define an adjacency matrix \mathbf{A}_D to represent the *meta dependency graph*, where $\mathbf{A}_{D_{i,j}} = 1$ when there is an edge connects node i and node j in dependency tree. Then we employ a L-layer stacked GCN [6] to convolute the *meta dependency graph*. Given node \mathbf{u} at the l -th layer, the graph convolutional operation can be defined as:

$$h_{\mathbf{u}}^{(l+1)} = \text{RELU} \left(\sum_{j=1}^n \mathbf{A}_{D_{i,j}} \mathbf{W}^{(l)} h_{\mathbf{u}_j}^{(l)} + b^{(l)} \right) \quad (3)$$

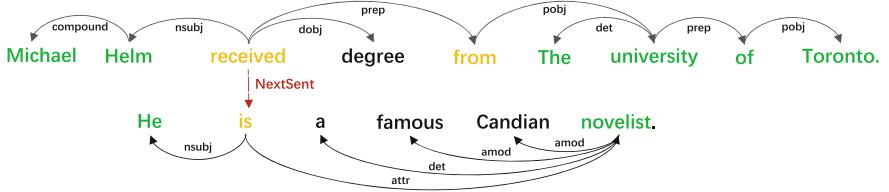


Fig. 3. An example of document meta dependency path nodes (DMDP). Mention and DMDP nodes are respectively colored in green and yellow. (Color figure online)

where $\mathbf{W}^{(l)} \in \mathbb{R}^{d_n \times d_n}$ and $b^{(l)} \in \mathbb{R}^{d_n}$ are trainable parameters, d_n is the dimension of node representations.

After the graph information propagation in *meta dependency graph*, we can obtain new representations of mention and DMDP nodes, we respectively denote them by \mathbf{m}_{glo} and \mathbf{mdp}_{glo} which encode the semantic information of the whole document.

2.4 Mention Interaction Graph

Past works [4, 9, 18] showed that local information is also important for relation classification, which can be captured by mention interactions. But the local context of different mentions is complex, it is hard to create a graph by explicit rules (e.g., co-references, syntactic trees or heuristics). Hence, we employ soft-attention mechanism [17] to construct an implicit graph. The key idea is to use attention for inducing interactions between mention nodes, especially for those connected by indirect, multi-hop paths.

We first compute an adjacency matrix \mathbf{A}_M for *mention interaction graph* by using self attention mechanism [17]. Then similar to previous steps in mDG, we apply graph convolutional operation to aggregate mention interactions.

$$\mathbf{A}_M = \text{softmax}\left(\frac{QW_t^Q \times (KW_t^K)^T}{\sqrt{d_n}}\right) \quad (4)$$

$$h_m^{(l+1)} = \text{RELU}\left(\sum_{j=1}^n \mathbf{A}_{M_{ij}} \mathbf{W}^{(l)} h_{m_j}^{(l)} + b^{(l)}\right) \quad (5)$$

where $W^Q \in \mathbb{R}^{d_n \times d_n}$, $W^K \in \mathbb{R}^{d_n \times d_n}$ are trainable projection matrices. Q and K are both equal to \mathbf{m}_{glo} which is from mDG. $\mathbf{W}^{(l)} \in \mathbb{R}^{d_n \times d_n}$ and $b^{(l)} \in \mathbb{R}^{d_n}$ are trainable parameters. After the operation of mutual reasoning between mentions, we get the mention representations \mathbf{m}_{loc} , which contain local information of mentions.

2.5 Entity Inference Graph

The goal of *entity inference graph* is to integrate long distance global information from mDG, and local interaction information from MG into entities. Therefore,

we generate a fully connect weighted graph with \mathbf{mdp}_{glo} , \mathbf{m}_{glo} , \mathbf{m}_{loc} and \mathbf{e} nodes. The initial representation of an entity node \mathbf{e}_i is calculated by averaging of its mention representations (e.g., $h_{e_i} = [avg_{m_j \in e_i}(h_{m_j})]$).

Given a specific entity \mathbf{e}_i , different types of neighboring nodes may have different impacts on it. For example, the \mathbf{mdp}_{glo} may contain more inter-sentential global information than \mathbf{m}_{loc} . But when \mathbf{e}_i needs fine-grained information, \mathbf{m}_{loc} is more useful. Additionally, different neighboring entities could also have different importance. To capture both the different importance at neighboring node level and neighboring type level for entities, we design a novel hybrid attention mechanism which can learn the graph connection weights in end to end fashion.

Neighboring Type Attention. For an entity node \mathbf{e}_v , the neighboring type attention learns the weights of different types of neighboring nodes. Specifically, we first represent the embedding of the type τ as $h_\tau = \sum_{v' \in \mathcal{N}_{e_v}} h_{v'}$, which is the sum of the neighboring node features $h_{v'}$, where the nodes $v' \in \mathcal{N}_{e_v}$ and are with the type τ . Then, we calculate the type attention scores based on the current node embedding h_{e_v} and the type embedding h_τ :

$$a_\tau = \text{LeakyReLU}(\mu_\tau^T \cdot [h_{e_v} || h_\tau]) \quad (6)$$

where μ_τ is the trainable attention vector for the type τ .

Then we obtain the type attention weights by normalizing the attention scores across all the types with the softmax function:

$$\alpha_\tau = \frac{\exp(a_\tau)}{\sum_{\tau' \in \mathcal{T}} \exp(a_{\tau'})} \quad (7)$$

Neighboring Node Attention. We design the neighboring node attention to capture the importance of different neighboring nodes and reduce the weights of noisy nodes. Formally, for entity node e_v and its neighboring node $v' \in \mathcal{N}_{e_v}$ with the type τ' , we compute the node attention scores based on the node embeddings h_{e_v} and $h_{v'}$ with the type attention weight $\alpha_{\tau'}$ for the node v' :

$$\beta_{e_v v'} = \sigma(v^T \cdot \alpha_{\tau'} [h_{e_v} || h_{v'}]) \quad (8)$$

where v is the trainable attention vector. Then we normalize the node attention scores similar to above:

$$\beta'_{e_v v'} = \frac{\exp(\beta_{e_v v'})}{\sum_{u \in \mathcal{N}_{e_v}} \exp(\beta_{e_v u})} \quad (9)$$

After the computation of type attention and node attention, the representations of all neighboring nodes h_u in \mathcal{N}_{e_v} are aggregated to \bar{h}'_{e_v} :

$$\bar{h}_{e_v} = \text{RELU}(\sum_{u \in \mathcal{N}_{e_v}} \beta'_{e_v u} (h_u \mathbf{W}_v + b_v)) \quad (10)$$

$$\bar{h}'_{e_v} = \mathcal{LN}(\bar{h}_{e_v} + (\sigma(\bar{h}_{e_v} \mathbf{W}_{l1} + b_{l1}) \mathbf{W}_{l2})) \quad (11)$$

where $\mathbf{W}_v \in \mathbb{R}^{d_n \times d_n}$, $\mathbf{W}_{l1} \in \mathbb{R}^{d_n \times 4d_n}$, $\mathbf{W}_{l2} \in \mathbb{R}^{4d_n \times d_n}$. $b_v \in \mathbb{R}^{d_n}$ and $b_{l1} \in \mathbb{R}^{4d_n}$ are the bias vectors. \mathcal{LN} is the LayerNorm function and $\sigma(\cdot)$ is activation function GELU. \bar{h}'_{e_v} is the v -th entity representation from EG. We get the final representation \mathbf{e} , which contains a vast amount of relation inference information.

2.6 Relation Classification

To classify the relations for an entity pair $(\mathbf{e}^{head}, \mathbf{e}^{tail})$, we first concatenate entity representations and relative distance representations as follows:

$$\hat{\mathbf{e}}^{head} = [\mathbf{e}^{head}; \mathbf{Dist}(\delta_{ht})] \quad (12)$$

$$\hat{\mathbf{e}}^{tail} = [\mathbf{e}^{tail}; \mathbf{Dist}(\delta_{th})] \quad (13)$$

where δ_{ht} means the relative distance of the head entity to tail entity, δ_{th} is similarly defined. **Dist** is a trainable relative distance embedding matrix. Then, we use a bilinear function to compute the probability for each relation type:

$$P(r|\mathbf{e}^{head}, \mathbf{e}^{tail}) = \text{sigmoid}(\mathbf{W}_{r2}\sigma(\mathbf{e}^{head}\mathbf{W}_{r1}\mathbf{e}^{tail} + b_{r1}) + b_{r2}) \quad (14)$$

where $\mathbf{W}_{r1}, \mathbf{W}_{r2} \in \mathbb{R}^{d_n \times d_n \times d_r}$, $b_{r1}, b_{r2} \in \mathbb{R}^{d_r}$ are relation type dependent trainable parameters, d_r is the number of relation types. We use binary cross entropy as the classification loss to train HAIN:

$$\text{loss} = - \sum_{r=1}^{d_r} y_r \log P(r|\mathbf{e}^{head}, \mathbf{e}^{tail}) + (1 - y_r) \log(1 - P(r|\mathbf{e}^{head}, \mathbf{e}^{tail})) \quad (15)$$

where $y_r \in \{0, 1\}$ is the true value on relation r .

3 Experiments

3.1 Dataset

We evaluate HAIN on DocRED [20] builted from Wikipedia and Wikidata, which is the largest document-level RE dataset. Both human-annotated and distantly-supervised data are offered. We only use the human-annotated data.

3.2 Baseline Models

We compare our HAIN with the following models.

- **Sequence-based Models.** Yao et al. [20] proposed several baseline models which used CNN/LSTM as encoder and predicted relations between entities by a bilinear function. Context-Aware [15] incorporated context relation information by attention, and Yao et al. [20] adapted it for document-level RE. HIN [16] aggregated the inference information of different granularity to predict relations.

- **Graph-based Models.** LSR [9] induced a latent document graph by maximum tree theory and used GCN for multi-hop reasoning. Nan et al. [9] also adopted GCNN [13] and AGGCN [23] for DocRED, while these are state-of-the-art sentence-level RE models. GEDA [7] characterized the complex interaction between sentences via a dual attention network. GAIN [22] proposed a novel path reasoning mechanism to infer relations between entities.
- **PLM-based Models.** BERT-RE [19] simply used BERT [2] as encoder to get a contextual entity representations. CorefBERT [21] designed a mention reference prediction task to enhance the coreferential reasoning ability of the pre-trained language model explicitly.

Table 1. Main results of different models on DocRED. Results with \dagger are implemented and published by Nan et al. [9]. Other results are reported in their original papers.

Model	Dev		Test	
	IgnF1	F1	IgnF1	F1
CNN [20]	41.58	43.45	40.33	42.26
LSTM [20]	48.44	50.68	47.71	50.07
Context-Aware [20]	48.94	51.09	48.40	50.70
HIN-GloVe [16]	51.06	52.95	51.15	53.30
GCNN \dagger [13]	46.22	51.52	49.57	51.62
AGGCN \dagger [3]	46.29	52.47	48.89	51.45
GEDA [7]	51.03	53.60	51.22	52.97
LSR-GloVe [9]	48.82	55.17	52.15	54.18
GAIN-GloVe [22]	53.05	55.29	52.66	55.08
HAIN-GloVe	54.98	56.03	54.73	55.76
BERT-RE _{base} [19]	–	54.16	–	53.20
GEDA-BERT _{base} [7]	54.52	56.16	53.71	55.74
HIN-BERT _{base} [16]	54.29	56.31	53.70	55.60
CorefBERT _{base} [21]	55.32	57.51	54.54	56.96
LSR-BERT _{base} [9]	52.43	59.00	56.97	59.05
GAIN-BERT _{base} [22]	59.14	61.22	59.00	61.24
HAIN-BERT_{base}	59.77	62.31	59.43	61.41
CorefBERT _{large} [21]	56.73	58.88	56.48	58.70
GAIN-BERT _{large} [22]	60.87	63.09	60.31	62.76
HAIN-BERT_{large}	61.27	63.91	61.23	63.01

3.3 Experimental Setup

Following Yao et al. [20], we use the GloVe [11] embedding with BiLSTM, and BERT [2] as the context encoder. We use spaCy¹ to get syntactic dependency

¹ <https://spacy.io/>.

parse tree for each sentence. Then we use NetWorkX² to represent the dependency parse tree. In our HAIN implementation, we use 3 layers of GCN and set the dropout rate to 0.4, learning rate to 0.001. We train HAIN using Adam [5] as optimizer. All hyper-parameters are tuned on the development set.

We use F1 as the evaluation metric. Due to some relation instances are present in both training and dev/test sets, to avoid introducing evaluation bias, we also report Ign F1 which denotes F1 scores excluding relation instances shared by the training and dev/test sets.

3.4 Main Results

Table 1 lists the results of different models in DocRED [20] dev and test set. We can find that:

(1) The graph-based models [3,9] obtain comparable results, and the best graph-based model LSR [9] outperforms the best sequence-based model HIN [16]. We owe it to the graph structure can better encode long distance, cross-sentential information. (2) BERT [2] can further boost the performance of our model, which indicates the importance of prior knowledge. For example, HAIN-BERT_{base} outperforms HAIN-GloVe 6.28/5.65 in F1 scores. (3) HAIN-BERT_{large} has achieved the best results compared with all the models. We attribute it to the hierarchical graph structure and hybrid attention mechanism, the former can model global and local information from the document, the latter can effectively synthesize them.

Table 2. Intra- and inter-sentence experimental results. (Models with ♠ are reported in Nan et al., [9]. Model with † is re-trained based on their open implementation.)

Model	Intra-F1	Inter-F1
LSTM ♠ [20]	56.57	41.47
LSR-GloVe ♠ [9]	60.83	48.35
GAIN-GloVe [22]	61.67	48.77
HAIN-GloVe	62.72	49.87
BERT-RE _{base} ♠ [19]	61.61	47.15
GLRE [†] [18]	63.63	51.56
LSR-BERT _{base} ♠ [9]	65.26	52.05
GAIN-BERT _{base} [22]	67.10	53.90
HAIN-BERT_{base}	68.34	54.70

² <https://networkx.org/>.

3.5 Detail Analysis

Intra- and Inter-sentence Performance. An entity pair requires inter-sentence reasoning if the two entities from the same document have no mentions in the same sentence. We report the Intra-F1 and Inter-F1 scores in Table 2, which only consider intra- or inter-sentence relations respectively.

Under the same setting, our HAIN outperforms all the other models in both intra- and inter- sentence setting. In particular, the differences in Inter-F1 scores between HAIN and other models tend to be larger than the differences in the Intra-F1 scores. For example HAIN-BERT_{base} improves 2.65 Inter-F1 scores compared with LSR-BERT_{base}. The results suggest that the hierarchical aggregation and inference structure of our model is capable of integrating the information across long distance, multiple sentences of a document.

Ablation Study. To further analyze HAIN, we conduct some ablation studies to verify the effectiveness of different modules and mechanisms of HAIN. Results are shown in Table 3. We can observe that: (1) When we remove DMDP nodes, and use SMDP nodes as Nan et al., [9], Inter-F1 drops by 1.26 scores. It means that DMDP nodes can capture richer inter-sentential information than traditional SMDP nodes. (2) F1 and Inter-F1 drops when we remove *meta dependency graph*, it shows that mDG can capture long distance dependency information. (3) Taking away *mention interaction graph*, Intra-F1 sharply drops by 4.59 scores. This drop shows that MG plays a vital role in capturing local information. (4) We remove the Hybrid attention mechanism. To be specific, we directly use the original GCN [6] to convolute the *entity inference graph*, ignoring the different importance of multi-granularity information. The Hybrid attention mechanism’s removal results in poor performance across all metrics. It suggests that our hybrid attention mechanism helps aggregate global and local information, therefore, improve the overall performance of document-level RE.

Table 3. Ablation Study of HAIN-BERT_{base} on DocRED dev set.

	F1	Ign F1	Intra-F1	Inter-F1
Full model	62.31	59.77	68.34	54.70
- DMDP Node	59.33	58.97	67.46	53.44
- Meta dependency graph	58.40	59.66	67.01	53.87
- Mention interaction graph	58.23	59.07	63.75	53.90
- Hybrid attention mechanism	57.89	56.23	60.77	51.10

Case Study. We list a few examples from DocRED dev set in Table 4, and use HAIN-GloVe in comparison with GAIN-GloVe [22] which is one of the most powerful graph-based model recently. We can observe that: (1) From example 1, we can find that long distance dependency information is necessary. The

head entity *William Earl Barber* and tail entity *Marines* cross five sentences, which need the model to be robust enough to tackle long distance cross sentence information. HAIN can capture long distance dependency information by meta dependency graph (mDG) to correctly identify the relation *military branch*. (2) From example 2, we can observe that logical reasoning is vital. We know *Dany Morin* is a *Canadian* in sentence 1, *Dany Morin* is a member of *New Democratic Party* in sentence 2. Extracting the relation between *Canadian* and *New Democratic Party* needs the bridge entity *Dany Morin*. HAIN handled this problem by reasoning in the entity inference graph (EG), which can fuse global and local important information to capture the logical relations. (3) Commonsense knowledge is required in example 3. Models must know that *M* is the code name of a person ahead of time, then identify the relation of *Miss Moneypenny* and *Bond* is *present in work*. Both HAIN and GAIN can not solve this issue, due to lack of the commonsense knowledge. We leave it as our future work.

Table 4. Case study on the DocRED. **Head entities** and **Tail entities** are colored accordingly. Other relevant entities are colored in **blue**.

[1] William Earl Barber (November 30 , 1919 April 19 , 2002) was a United States Marine Corps colonel. [2] He fought on Iwo Jima during World War II and was awarded the Medal of Honor for his actions in the Battle of Chosin Reservoir during the Korean War ... [4] Despite the extreme cold weather conditions and a bullet wound to the leg, Barber refused evacuation and an order for his company to ... [5] Barber, aware that leaving would cause 8,000 Marines of his division to be trapped in North Korea , held on to the position with his men ...	Relation Label: military branch	HAIN: military branch	GAIN: N/A
[1] Dany Morin (born December 19, 1985) is a Canadian businessman and former politician. [2] He represented the electoral district of Chicoutimi: Le Fjord as a member of the New Democratic Party ... [3] He served as the NDP associate critic for lesbian, gay, bisexual, transgender, and transsexual issues, alongside lead critic Randall Garrison ...	Relation Label: country	HAIN: country	GAIN: N/A
[1] Miss Moneypenny , later assigned the first names of Eve or Jane , is a fictional character in the James Bond novels and films. [2] She is secretary to M , who is Bond 's superior officer and head of the British Secret Intelligence Service (MI6) . [3] Although she has a small part in most of the films, it is always highlighted by the underscored romantic tension between her and Bond ...	Relation Label: present in work	HAIN: N/A	GAIN: N/A

4 Related Work

In practice, many real world relation instances can only be extracted across sentences. For example, Yao et al., [20] made an analysis on Wikipedia corpus, at least 40.7% of relations can only be extracted on the document level. Therefore, natural language processing community has gradually pay much attention to document-level RE. To accelerate the research on document-level RE, Yao et al. [20] introduced DocRED, constructed from Wikipedia and Wikidata. At present, DocRED

is the largest document-level RE dataset. Quirk et al., [12] incorporated both standard dependencies and discourse relations in RE. Peng et al., [10] explored different LSTM approaches with various dependencies, such as syntactic and sequential. But they both captured document specific features, ignored relational inference in document. Recently, many graph-based models are designed to handle this problem. Sahu et al., [13] utilized syntactic parsing and coreference resolution to build a document-level graph for graph inference. Christopoulou et al., [1] constructed a document graph with heterogeneous types of nodes and edges, and proposed edge-oriented model for global relation inference. Li et al., [7] proposed a dual attention network to characterize the interactions in document. Nan et al., [9] treated the graph structure as a latent variable and constructed it by utilizing structured attention [8]. Zeng et al. [22] proposed a novel path reasoning mechanism to enhance the reasoning abilities for RE. Different from the previous works, we construct a hierarchical graph which can utilize the structural information from syntactic trees to capture long-distance dependency. Moreover, we propose a novel hybrid attention mechanism to effectively aggregate global and local information to reason logical relations between entities.

5 Conclusion

In this paper, we proposed a hierarchical aggregation and inference network (HAIN) for document-level RE. It respectively establishes three different information granularity graphs which can effectively integrate relevant relation inference evidences from coarse to fine. Experiments show that our HAIN achieves state-of-the-art performance on the widely used dataset DocRED. In the future, we plan to utilize extra commonsense knowledge to help train more efficient models for solving the commonsense relation inference problem.

References

1. Christopoulou, F., Miwa, M., Ananiadou, S.: Connecting the dots: document-level neural relation extraction with edge-oriented graphs. In: EMNLP (2019)
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL (2019)
3. Guo, Z., Zhang, Y., Lu, W.: Attention guided graph convolutional networks for relation extraction. In: ACL (2019)
4. Gupta, P., Rajaram, S., Schütze, H., Runkler, T.: Neural relation extraction within and across sentence boundaries. In: AAAI (2019)
5. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: ICLR (2015)
6. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)
7. Li, B., Ye, W., Sheng, Z., Xie, R., Xi, X., Zhang, S.: Graph enhanced dual attention network for document-level relation extraction. In: Coling (2020)
8. Liu, Y., Lapata, M.: Learning structured text representations. In: TACL (2018)
9. Nan, G., Guo, Z., Sekulić, I., Lu, W.: Reasoning with latent structure refinement for document-level relation extraction. In: ACL (2020)

10. Peng, N., Poon, H., Quirk, C., Toutanova, K., Yih, W.T.: Cross-sentence N-ary relation extraction with graph LSTMs. In: TACL (2017)
11. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: EMNLP, pp. 1532–1543 (2014)
12. Quirk, C., Poon, H.: Distant supervision for relation extraction beyond the sentence boundary. In: EACL (2016)
13. Sahu, S.K., Christopoulou, F., Miwa, M., Ananiadou, S.: Inter-sentence relation extraction with document-level graph convolutional neural network. In: ACL (2019)
14. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Sig. Process. **45**(11), 2673–2681 (1997)
15. Sorokin, D., Gurevych, I.: Context-aware representations for knowledge base relation extraction. In: EMNLP (2017)
16. Tang, H., et al.: HIN: hierarchical inference network for document-level relation extraction. In: PAKDD (2020)
17. Vaswani, A., et al.: Attention is all you need. In: NIPS (2017)
18. Wang, D., Hu, W., Cao, E., Sun, W.: Global-to-local neural networks for document-level relation extraction. In: EMNLP (2020)
19. Wang, H., Focke, C., Sylvester, R., Mishra, N., Wang, W.: Fine-tune BERT for DocRED with two-step process. arXiv preprint [arXiv:1909.11898](https://arxiv.org/abs/1909.11898) (2019)
20. Yao, Y., et al.: DocRED: a large-scale document-level relation extraction dataset. In: ACL (2019)
21. Ye, D., Lin, Y., Du, J., Liu, Z., Sun, M., Liu, Z.: Coreferential reasoning learning for language representation. In: EMNLP (2020)
22. Zeng, S., Xu, R., Chang, B., Li, L.: Double graph based reasoning for document-level relation extraction. In: EMNLP (2020)
23. Zhang, Y., Guo, Z., Lu, W.: Attention guided graph convolutional networks for relation extraction. In: ACL (2019)



Incorporate Lexicon into Self-training: A Distantly Supervised Chinese Medical NER

Zhen Gan^{1,3}, Zhucong Li^{1,2}, Baoli Zhang¹, Jing Wan³, Yubo Chen^{1,2(✉)},
Kang Liu^{1,2}, Jun Zhao^{1,2(✉)}, Yafei Shi⁴, and Shengping Liu⁴

¹ National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China

{zhucong.li,baoli.zhang,yubo.chen,kliu,jzhao}@nlpr.ia.ac.cn

² School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China

³ Beijing University of Chemical Technology, Beijing, China

{ganzen,wanj}@mail.buct.edu.cn

⁴ Beijing Unisound Information Technology Co., Ltd., Beijing, China

{shiyafei,liushengping}@unisound.com

Abstract. Medical named entity recognition (NER) tasks usually lack sufficient annotation data. Distant supervision is often used to alleviate this problem, which can quickly and automatically generate annotated training datasets through dictionaries. However, the current distantly supervised method suffers from noisy labeling due to limited coverage of the dictionary, which will cause a large number of unlabeled entities. We call this phenomenon an incomplete annotation problem. To tackle the incomplete annotation problem, we propose a novel distantly supervised method for Chinese medical NER. Specifically, we propose a high recall self-training mechanism to recall potential unlabeled entities in the distant supervision dataset. To reduce error in the high recall self-training, we propose a fine-grained lexicon enhanced scoring and ranking mechanism. Our method improves 3.2% and 5.03% compared to the baseline models on the dataset we proposed and a benchmark dataset for Chinese medical NER.

Keywords: Medical named entity recognition · Fine-grained lexicon · Distantly supervised · Self-training

1 Introduction

Medical named entity recognition (NER) is a classic task in medical natural language processing [8, 11, 14]. It is the basic technology of medical information extraction and medical knowledge graph. A major problem with the medical NER task is lacking labeled data. Since medical entities are highly specialized

Z. Gan and Z. Li—Equal contribution.

and require professionals to label them, there are very few publicly released medical NER datasets. Therefore, the way to obtain sufficient labeled data is a huge challenge.

Sample of Chinese medical text	Dictionary
Span	Label
行结肠镜发现 直肠息肉 0.3×0.6cm, 未予特殊处置。	盲肠(cecum) Anatomy
Colonoscopy revealed a rectal polyp of 0.3×0.6 cm, and no special treatment was given.	培美曲塞(Pemetrexed) Drug
结肠镜示: 升结肠: 近盲肠见多处瘢痕改变。	
Colonoscopy showed: Ascending colon Multiple scar changes were seen near the cecum	

Fig. 1. The incomplete annotation problem of the distantly supervised NER method. The red font indicates an unlabeled entity, green font indicates the correct labeled entity. The left side is the text to be labeled, and the right side is the dictionary. Since the dictionary does not contain the two entities of “直肠息肉” (a rectal polyp) and “升结肠” (ascending colon), these two entities are missing in the labeled data. (Color figure online)

Recently, the distantly supervised method has been applied to automatically generate labeled data based on domain-specific dictionaries. This method first identifies entity mentions by exact string matching with the dictionary, and then assigns corresponding types to the entity mentions. Although this method is effective to label data automatically, it suffers from noisy labeling due to limited coverage of the dictionary, which will cause a large number of entities unlabeled. We call this phenomenon as the incomplete annotation problem [4, 7, 9, 10, 18] and show it in Fig. 1.

In this work, we propose a novel distantly supervised method for Chinese medical NER to tackle the incomplete annotation problem. We constructed a Chinese medical NER dataset named *CDD* to verify the effectiveness of our method. The training set of the dataset is annotated using distantly supervised methods, and the development set and test set are annotated by professional doctors. In addition, we are also conducting experiments on *CCKS 2019* [2] which is a public Chinese medical NER dataset. The training sets of *CCKS 2019* are constructed by randomly removing the annotated named entities in well-annotated datasets.

Specifically, our approach is divided into two parts:

High Recall Self-training Mechanism: We propose a high recall self-training mechanism to recall potential unlabeled entities in the distant supervision dataset. Self-training is an effective method to tackle the incomplete annotation problem. Here, self-training specifically refers to iteratively predict the training set by itself. Therefore we propose a high recall strategy inspired by the K -fold cross-validation method. We divide the training set into K -fold data and take the $K-1$ fold data as the new training set to train independent K NER models with the same development set. We recall a large number of entities through the high recall strategy and all entities predicted by the K models. After a round of iteration, we verify whether the training data meets the requirements.

Fine-grained Lexicon Enhanced Scoring and Ranking mechanism: To reduce error in the high recall self-training, we propose a fine-grained lexicon enhanced scoring and ranking mechanism. Each entity has its unique features and some common features with other entities. For example, the “胸部CT” (Chest CT) and the “腹部CT” (Abdomen CT) are medical examinations. They also have the same structure of “身体部位+ CT” (body parts + CT) and the same composition as “CT”. In medical examination named entities, this situation is very common. The fine-grained lexicon is obtained from the entity in the distant supervision data for word segmentation. For entities recalled by self-training, having more fine-grained lexicon labels is more likely to be the correct entity. We use the fine-grained lexicon to label the entities of the training set construct positive samples and create appropriate negative samples. We use a fine-grained lexicon to label the entities in the training set as positive samples and construct appropriate negative samples. Then we combine the positive and negative samples to train the scoring and ranking model.

The main contributions of this paper can be summarized as follows:

- To tackle the incomplete annotation problem, we propose a novel distantly supervised method for Chinese medical NER and construct a Chinese medical dataset.
- To recall potential unlabeled entities in the distant supervision dataset, we propose a high recall self-training mechanism. To reduce error in the high recall self-training, we propose a fine-grained lexicon enhanced scoring and ranking mechanism.
- Experiments prove the effectiveness of our method, which improves **3.2%** and **5.03%** compared to the baseline models on the dataset we proposed and a benchmark dataset for Chinese medical NER.

2 Related Work

2.1 Distantly Supervised NER

Distant supervision methods for NER generally cause data incomplete annotation problems. Several approaches to this issue have been proposed. Fuzzy CRF and AutoNER [12] allow learning from high-quality phrases. However, since these phrases are obtained through distant supervision, the unlabeled entities in the corpora may still be missed. PU learning [9, 10] unbiasedly and consistently estimates the training loss. Partial CRF [4, 18] supports learning from incomplete annotations. Different from the above method, our method applies a high recall self-training mechanism and fine-grained lexicon enhanced scoring and ranking mechanism to recall potential unlabeled entities in the distant supervision dataset and reduce error in the high recall self-training. So training data noise is reduced and the performance of the model will improve. The fine-grained lexicon required by our model is very easy to obtain and the method has good transferability.

2.2 Self-training

Self-training is an effective method to tackle the incomplete annotation problem, which specifically refers to repeatedly predicting the training set itself, and this method can capture noise pattern information well. Jie et al. [4] made a bold attempt to treat each location label outside the entity as a hidden variable. Their model calculates the probability of potential label paths as training loss weights in the self-training process. Their method essentially aims to detect and add potentially lost entities. In the same year, Mayhew et al. [9] and Wang et al. [17] aims to generate a weighted training set based on self-training to detect and reduce false label weights. Then they use this set to train a weighted NER model. The self-training of our method is to recall potential unlabeled entities in the distant supervision dataset. Self-training is to prepare for the following work of scoring and ranking entities. The higher the quality of the entity recalled by self-training, the more conducive to the next work.

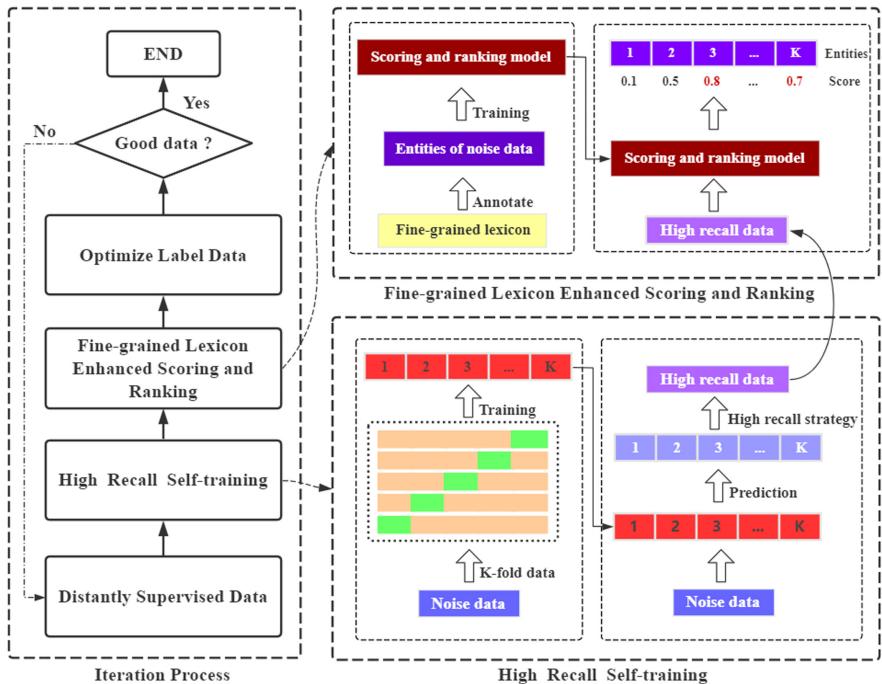


Fig. 2. Our basic model structure. On the left is the iterative process of semi-supervised data optimization and denoising. On the right is two main modules of the framework, namely High Recall Self-training (lower right) and Fine-grained Lexicon Enhanced Scoring and Ranking (upper right).

3 Our Method

Our method is to recall the unlabeled entities of training data, which achieves the purpose of removing the unlabeled entity noise. Our basic model structure is shown in the Fig. 2. This section will introduce our method from four subsections: Distantly Supervised Method Annotating Data, High Recall Self-training, Fine-grained Lexicon Enhanced Scoring and Ranking, and Iteration Process.

3.1 Distantly Supervised Method Annotating Data

There is a small amount of well-annotated data. We use remote supervision methods to automatically annotate a large amount of incomplete annotation data. Our method can recall unlabeled entities from annotation data. The following sections will introduce them in detail.

3.2 High Recall Self-training

To obtain the training data of high recall entities, we propose the high recall model with Self-training. Inspired by the K -fold cross-validation method, we divide the training set into K -fold data and take the $K-1$ fold data as the training set to train independent K NER models with the same development set. Due to the different distributions of unlabeled entities of the training data, the K models have the feature of large differences. Then all the training data is predicted with the K models, and we obtain the training set with higher recall through the high recall strategy. The main process of this section is shown in the lower right of Fig. 1.

Definition Description. We denote an input sentence as $s = \{x_1, x_2, \dots, x_n\}$ and the annotated named entity set as $E_s = \{e_0, e_1, e_2, \dots, e_m\}$. n is the sentence length and m is the amount of entities. Each member of e_k of set E_s is represented as $e_k = (pos_{start}, pos_{end}, label)$, which are the start position, end position and label of e_k respectively. The goal of our high recall model with self-training is to recall as many unlabeled entities as possible through self-training and high-recall strategies, so that the E_s set and the fully labeled entity set have more overlapping entities.

Model Training. In our self-training model, dividing the training set into K -fold data, we take the $K-1$ fold data as the training set and train independent K NER models with the same development set of no unlabeled entity. For the basic NER model framework of this part, we have selected two commonly used model structures: biLSTM [8]+CRF [6, 15] and BERT [1]+biLSTM+CRF.

High Recall Strategy. We designed a high recall strategy, which can combine multiple training data prediction entities with the entities from the previous round of data to obtain high recall entity training data. The set of prediction result entities E_{pre} for each fold, our high recall strategy is represented by a mathematical notation as follows:

$$E_{new} = E_{last} + \hat{E}_{extra} \quad (1)$$

$$E_{extra} = \sum_{0 \leq i < k} E_{pre_i} - E_{last} \quad (2)$$

$$\hat{E}_{extra} = \{e_{extra} \mid \sum e_{extra} > \alpha, e_{extra} \in E_{pre_i}, 0 \leq i < k\} \quad (3)$$

E_{new} is the entity set of the new round of data. E_{last} is the entity set of the previous round of data. E_{extra} is all new entities predicted by models. \hat{E}_{extra} means to select the entity e_{extra} that meets the requirements from E_{extra} to keep it. The conditions: the entity belongs to E_{pre} and the frequency of occurrence reaches the threshold α , then it is retained, otherwise the entity is removed.

3.3 Fine-Grained Lexicon Enhanced Scoring and Ranking

The high recall model with self-training obtains training data with high recall entities but it also recalls a large number of error entities, causing unnecessary noise. Therefore, we propose a fine-grained lexicon Enhanced scoring and ranking mechanism to rank the recalled new entities. This mechanism only retains entities whose scores reach the threshold, greatly reducing redundant entities and obtaining less noisy training data. This section will introduce our model in detail from the following four subsections: overall, data construction, model training and Entity Scoring, and Ranking Strategy.

The fine-grained lexicon enhanced scoring and ranking mechanism is mainly to score and rank the new entities in the high recall training data, retain the entities that meet the requirements, and delete the entities that do not meet the requirements. Guided by this idea, we obtain new training data with more correct new entities and redundant new entities, which is to improve the quality of training data.

Each entity has its unique features and some common features with other entities. The common features include the same structure, the same component, etc. For example, “白细胞” (White blood cells) and “红细胞” (red blood cells) are two inspection entities that respectively indicate the number of white blood cells and red blood cells in the blood. These two entities have the same structure “* + 细胞” (* + cells) and the same component “细胞” (cells). Another example is more obvious. The “胸部CT” (Chest CT) and the “腹部CT” (Abdomen CT) are medical examinations. They also have the same structure of “身体部位+CT” (body parts + CT), and have the same composition as “CT”. In medical examination named entities, this situation is very common.

For this reason, we propose the fine-grained lexicon enhanced scoring and ranking mechanism to construct a sequence labeling task to learn the entity features within and between entities, thereby scoring and ranking the new entities. Define a entity sequence as $e = \{x_1, x_2, \dots, x_n\}$, the length of the entity e is n , this entity contains a fine-grained lexicon entity set as $E_{\bar{e}} = \{\bar{e}_0, \bar{e}_1, \dots, \bar{e}_m\}$, where $\bar{e}_i = (\text{pos}_{start}, \text{pos}_{end}, \text{label}) (0 \leq i \leq m)$ represents a fine-grained lexicon entity. Constructing new data with a fine-grained lexicon and entities of training data, we train the ranking model.

Data Construction. **Step 1**, Obtain the entity of the training data. **Step 2**, Obtain a fine-grained dictionary. Use the jieba word segmentation tool to

segment the entities of the training data obtained in the first step. After Removing duplicate entities and deleting unreasonable segmentation results, we get a fine-grained dictionary. **Step 3**, Use the data from the first and second steps to generate training data for the new task. First, we use a fine-grained dictionary to back-label the original training data. The inspection entity part is extracted as a positive sample, and in addition, we construct a negative sample of the data. Negative samples are mainly divided into two parts, one is difficult negative samples, and the other is ordinary negative samples. Difficult negative samples are generated using pre-defined N-gram feature templates around positive samples, and ordinary negative samples are generated using pre-defined N-gram feature templates around non-entities (parts with fine-grained labels), which are used in our experiments. The template is shown in Table 1. Among them, $E_{(i,j)}$ represents the entity in the sentence, and the beginning and ending positions of the entity span are the i -th and j -th positions of the sentence, respectively. The negative sample composition is to add one, two, or three characters before and after such an entity. After obtaining all the negative samples, to avoid the imbalance of positive and negative samples, we randomly selected a certain proportion of negative samples and added them to the positive sample data to form the training data of the final scoring model.

Table 1. N-gram feature template diagram

Type	Templates
1-gram	$x_{i-1}E_{(i,j)}, E_{(i,j)}x_{j+1}, x_{i-1}E_{(i,j)}x_{j+1}$
2-gram	$x_{i-2}x_{i-1}E_{(i,j)}, E_{(i,j)}x_{j+1}x_{j+2}, x_{i-2}x_{i-1}E_{(i,j)}x_{j+1}x_{j+2}$
3-gram	$x_{i-3}x_{i-2}x_{i-1}E_{(i,j)}, E_{(i,j)}x_{j+1}x_{j+2}x_{j+3}, x_{i-3}x_{i-2}x_{i-1}E_{(i,j)}x_{j+1}x_{j+2}x_{j+3}$

Model Training. In theory, we can use all sequence labeling models as our scoring model. In this article, we selected two models of biLSTM+CRF and BERT+biLSTM+CRF, and compared them. The latter has better results, but the former is more efficient. Using the aforementioned method to build data, we can perform model training and get our key scoring model.

Entity Scoring and Ranking Strategy. After we get the ranking model, such as biLSTM+CRF, we can rely on the model to score the new entity predicted. First, we use the ranking model to predict all new entities, and use the ranking model to make fine-grained label predictions for these new entities. It then scores the number of fine-grained tags contained in each new entity. We tried a simple unified scoring method, and also tried to choose different scoring methods according to the length of the new entity. The experimental results show that the latter has better stability and accuracy. The specific method is as follows. For the newly predicted entity e_{new} , the two scoring functions are as follows:

$$Score_1 = \frac{H(L_{\bar{e}}, e_{new})}{S(e_{new})} \quad (4)$$

$$Score_2 = \alpha \cdot \frac{H(L_{\bar{e}}, e_{new})}{S(e_{new})}, 0 < \alpha < 1 \quad (5)$$

$L_{\bar{e}}$ represents fine-grained lexicon label set, and $H(L_{\bar{e}}, e_{new})$ represents the number of fine-grained labels appearing in the new entity e_{new} . $S(e_{new})$ represents the number of characters of e_{new} . α indicates that a coefficient is given according to the length of the new entity e_{new} .

3.4 Iteration Process

After the training data is iterated, a large number of unlabeled entities are recalled, we will verify whether the training data meets the requirements. If the requirements are met, we stop the iteration, otherwise continue to iterate. The best result of our experiment in two datasets appeared in the second and third iterations.

4 Experiments

In this section, we conduct a series of experiments on two Chinese medical NER datasets to prove the effectiveness of our method.¹

4.1 Dataset

There are two datasets, both of which are Chinese medical NER datasets.

The first dataset named *CDD* is from the laboratory examination text and other auxiliary examinations text of the China Disease Resource Database. 1857 samples of them are annotated according to the Medical Named Entity Recognition labeling standard established by professional doctors. And 5574 samples of them are annotated by the distantly supervised method. We spent more than a month completing this difficult data collation and iterative annotation work.

The second dataset is the *CCKS 2019*, all samples of which are clinical text. There are 6 categories of it that are defined as follows: Disease and diagnose (Dis), Imaging examination (ImgExam), Laboratory examination (LabExam), Operation, Drug, and Anatomy. In particular, since the *CCKS 2019* dataset does not have a development set, we randomly selected 20% samples from the training data as the development set. The statistical details of the two original datasets are shown in Table 2.

4.2 Evaluation

For the fairness of model comparison, we refer to the standard-setting and use the micro-average F1 score to evaluate all methods, and report the precision (Pre) and recall (Rec) as percentages.

¹ Code is available at <https://github.com/ganzhenj/2021NLPCC>.

Table 2. The statistics details of two original datasets.

Dataset	Type	Train	Dev	Test
<i>CDD</i>	Sentence	5.6K	0.9K	0.9K
	Char	527.7K	83.0K	90.3K
<i>CCKS 2019</i>	Sentence	1.0K	–	0.4K
	Char	418.4K	–	132.7K

4.3 Experiment Setting

- **Character Embedding:** In our experiments, We use the same character embeddings as [19], which is pre-trained on Chinese Giga-Word.
- **BERT Enhanced Character Embedding:** Since pre-trained language models have been proven to be effective on several tasks, we also experiment with employing BERT to augment our model via BERT enhanced embedding.
- **Hyper-parameter Setting:** The biLSTM+CRF is trained for 50 epochs with the learning rate of 0.01 using Adam [5] optimizer and the dropout [13] is 0.3. The BERT module of BERT+biLSTM+CRF is trained for 50 epochs with the learning rate of 5e-5 using Adam optimizer. For the BiLSTM+CRF module of it, the learning rate is 5e-4, the batch size is 16, and the dropout is 0.3.

4.4 Main Results

The main results are shown in Table 3. The “Gold” represents that the Bilstm+CRF model trains with a noise-free label dataset, and all other models train with a noisy label dataset. The “Base” represents the Bilstm+CRF model. Compared with the baseline, our method improves 3.2% and 5.03% in F1 on two benchmark datasets. Our model greatly improves 9.08% in Recall on the *CCKS 2019*. Our model can effectively recall the correct entity.

The effects of other models have not achieved good results, which reflects the special difficulty of medical NER dataset. It cannot achieve the expected effect in the medical NER field for the general model of solving incompletely labeled NER. Therefore, we need to propose effective solutions to the specific difficulties and challenges of medical NER dataset. Our model starts from the medical text itself and uses the medical fine-grained dictionary inside the text to effectively remove the noise of the data. This is the advantage of our model.

Our method has good stability and robustness, and the effect is better on *CCKS 2019* with more categories, indicating that our model can cope with complex medical NER tasks. Our method can also easily integrate other methods, such as BERT and other pre-trained models, because we remove noise from the data level and optimize the training data. Our model has the effect of stable operation steps and perfection in the case of processing a small amount of annotated data.

Table 3. Main results.

Method	CDD			CCKS 2019		
	Pre	Rec	F1	Pre	Rec	F1
Gold	–	–	–	81.12	79.33	80.22
Base	66.51	52.23	58.51	80.85	62.25	70.34
Veit et al. [16]	37.41	35.54	36.45	45.33	43.44	44.36
Luo et al. [7]	41.01	32.67	36.37	46.97	44.75	45.84
Hedderich et al. [3]	36.43	36.56	36.50	48.06	47.85	47.95
Jie et al. [4]	55.50	42.60	48.20	78.50	54.32	64.21
Mayhew et al. [9]	69.39	47.18	56.17	75.17	60.27	66.90
Yang et al. [18]	–	–	60.90	–	–	66.53
Our method	69.01	55.81	61.71	79.89	71.33	75.37
Our+BERT	69.43	59.92	64.33	81.69	76.94	79.24

4.5 Ablation Study

To verify whether the various modules of our model affect the performance of the model, we designed an ablation experiment, the results of which are shown in Table 4. The “FLM” is the abbreviation of the fine-grained lexicon enhanced scoring and ranking mechanism. The “IP” is the abbreviation of the iteration process. The “-IP” means only one round of iteration and the best result of *CCKS 2019* is iterated for three rounds.

Table 4. Results of ablation experiments. The “BERT” represents BERT+Bilstm+CRF model. All number is the percentage of the F1 score.

Method	CCKS 2019						
	Dis	ImgExam	LabExam	Operation	Drug	Anatomy	average
Baseline	68.41	62.18	52.74	75.23	61.34	75.74	70.34
Our method	74.17	71.45	57.66	74.58	73.71	79.61	75.37
– FLM	70.33	63.33	60.40	74.36	68.45	77.93	73.28
– IP	72.70	68.96	60.72	70.59	70.97	79.46	74.80
BERT	77.13	69.76	54.72	79.08	87.07	80.14	77.39
Our+BERT	79.52	70.15	61.90	79.87	86.66	81.99	79.25

For *CCKS 2019*, there are a total of 6 categories of data, of which only one category has a decrease in F1 compared to Baseline, and the other five categories have a significant increase in F1. For the category “Drug”, there is an increase of 12.37% in F1. This is surprising. In addition, when we remove “FLM”, the F1 of the five categories has decreased, and the “LabExam” category has

increased. This shows that our “FLM” may have a certain destructive effect on this category, and the overall F1 has dropped by 2.09%. In general, the effectiveness of “FLM” has been affirmed. When we remove the “IP”, the overall impact is slightly weaker than that of “FLM”, but the “Operation” category is more affected, and the overall F1 drops by 0.57%. Finally, we experimented on BERT. The F1 of the “Drug” category has declined, and the others have increased, and the overall F1 has increased by 1.86%.

5 Conclusions

In this paper, we introduce a novel distantly supervised method for Chinese medical NER to tackle the incomplete annotation problem and construct a Chinese medical dataset. The key of our method is to use a high recall self-training mechanism and fine-grained lexicon enhanced scoring and ranking mechanism to recall potential unlabeled entities in the distant supervision dataset and reduce error in the high recall self-training. Experiments prove the effectiveness of our method. In the future, our work will extend the study to additional domains and languages.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (No. 61976211, No. 61922085). This work is supported by Beijing Academy of Artificial Intelligence (BAAI2019QN0301) and the Key Research Program of the Chinese Academy of Sciences (Grant NO. ZDBS-SSW-JSC006). This work is also supported by a grant from Beijing Unisound Information Technology Co., Ltd.

References

1. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) (2018)
2. Han, X., et al.: Overview of the CCKS 2019 knowledge graph evaluation track: entity, relation, event and QA. arXiv preprint [arXiv:2003.03875](https://arxiv.org/abs/2003.03875) (2020)
3. Hedderich, M.A., Klakow, D.: Training a neural network in a low-resource setting on automatically annotated noisy data. In: Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP, pp. 12–18 (2018)
4. Jie, Z., Xie, P., Lu, W., Ding, R., Li, L.: Better modeling of incomplete annotations for named entity recognition. 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), pp. 729–734 (2019)
5. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
6. Lafferty, J.D., McCallum, A., Pereira, F.C.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning, pp. 282–289 (2001)
7. Luo, B., et al.: Learning with noise: enhance distantly supervised relation extraction with dynamic transition matrix. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 430–439 (2017)

8. Ma, X., Hovy, E.: End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1064–1074 (2016)
9. Mayhew, S., Chaturvedi, S., Tsai, C.T., Roth, D.: Named entity recognition with partially annotated training data. In: Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), pp. 645–655 (2019)
10. Peng, M., Xing, X., Zhang, Q., Fu, J., Huang, X.J.: Distantly supervised named entity recognition using positive-unlabeled learning. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 2409–2419 (2019)
11. Peters, M., et al.: Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 2227–2237 (2018)
12. Shang, J., Liu, L., Gu, X., Ren, X., Ren, T., Han, J.: Learning named entity tagger using domain-specific dictionary. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2054–2064 (2018)
13. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014)
14. Strubell, E., Verga, P., Belanger, D., McCallum, A.: Fast and accurate entity recognition with iterated dilated convolutions. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 2670–2680 (2017)
15. Sutton, C., McCallum, A.: An introduction to conditional random fields for relational learning. In: Introduction to Statistical Relational Learning, vol. 2, pp. 93–128 (2006)
16. Veit, A., Alldrin, N., Chechik, G., Krasin, I., Gupta, A., Belongie, S.: Learning from noisy large-scale datasets with minimal supervision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 839–847 (2017)
17. Wang, Z., Shang, J., Liu, L., Lu, L., Liu, J., Han, J.: CrossWeigh: training named entity tagger from imperfect annotations. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 5157–5166 (2019)
18. Yang, Y., Chen, W., Li, Z., He, Z., Zhang, M.: Distantly supervised NER with partial annotation learning and reinforcement learning. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 2159–2169 (2018)
19. Zhang, Y., Yang, J.: Chinese NER using lattice LSTM. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1554–1564 (2018)

Summarization and Generation



Diversified Paraphrase Generation with Commonsense Knowledge Graph

Xinyao Shen^(✉), Jiangjie Chen, and Yanghua Xiao

School of Computer Science, Fudan University, Shanghai, China
{xinyaoshen19,jjchen19,shawyh}@fudan.edu.cn

Abstract. Paraphrases refer to text with different expressions conveying the same meaning, which is usually modeled as a sequence-to-sequence (Seq2Seq) learning problem. Traditional Seq2Seq models mainly concentrate on fidelity while ignoring the diversity of paraphrases. Although recent studies begin to focus on the diversity of generated paraphrases, they either adopt inflexible control mechanisms or restrict to synonyms and topic knowledge. In this paper, we propose **KnowledgeE-Enhanced Paraphraser (KEEP)** for diversified paraphrase generation, which leverages a commonsense knowledge graph to explicitly enrich the expressions of paraphrases. Specifically, KEEP retrieves word-level and phrase-level knowledge from an external knowledge graph, and learns to choose more related ones using graph attention mechanism. Extensive experiments on benchmarks of paraphrase generation show the strengths especially in the diversity of our proposed model compared with several strong baselines.

Keywords: Paraphrase generation · Knowledge graph · Diversified generation

1 Introduction

Paraphrases are texts conveying the same meaning while using different words, and the generation of paraphrases is a fundamental task in natural language processing (NLP). The technique has been widely used in many downstream applications, such as text summarization, question answering, semantic parsing, and so on [1].

Early studies on paraphrase generation include rule-based, grammar-based, lexicon-based, and statistical machine translation (SMT)-based approaches [17, 30]. Recently, sequence-to-sequence (Seq2Seq) models have become the dominant technique in the task of paraphrase generation [9, 21], especially since its great success in machine translation [25]. Although Seq2Seq models for paraphrase generation have shown promising results, they tend to generate highly similar outputs with inputs.

We argue that paraphrases should be diversified in nature since an input sentence corresponds to multiple possible paraphrases. To solve this problem, some studies [5, 19] introduce control mechanisms on the Seq2Seq model to produce a variety of paraphrases. However, the template or exemplars in the control mechanism does not cover all the possibilities of paraphrasing, and the introduction of the control mechanism is inflexible.

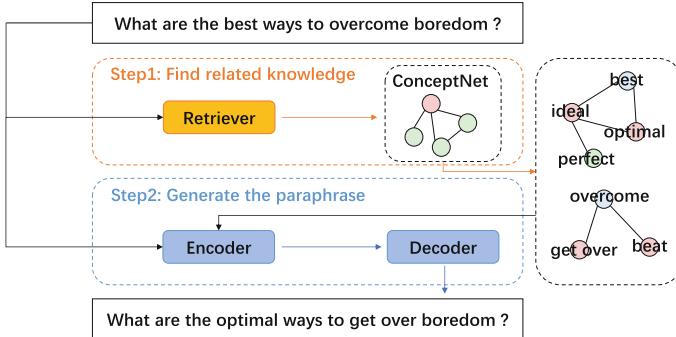


Fig. 1. The knowledge-enhanced model first retrieves a group of optional words or phrases and then generates a paraphrase using the original sentence as a prototype.

The main reason behind this challenge is that the available training data for paraphrasing is scarce and domain-specific [26]. One possible solution is to introduce *external knowledge* to increase the semantic richness of data. There are also efforts to exploit external knowledge in paraphrasing. Huang et al. [10] employ an external synonym dictionary to guide the rewriting of sentences. Liu et al. [16] extend the Seq2Seq structure to incorporate extra topic words for paraphrase generation. Restricting the utilization of knowledge only to those synonyms and topic words, effective as they are, does not exploit the full semantics of knowledge in paraphrase generation.

In this paper, we present an effective **KnowledgE-Enhanced Paraphraser (KEEP)**, which utilizes an external knowledge graph (KG) for diversified paraphrasing. We argue that the rich semantics within a KG can greatly benefit paraphrasing for concepts in the sentences through the semantic neighbors. KEEP first extracts a set of concepts from the paraphrase sentences annotated by entity linking systems. Then, we leverage the extracted words or phrases in paraphrase sentences as the start point to guide the traverses in the graph by graph attention mechanism, which derives from graph neural networks to attend on more appropriate concepts. Finally, we use an attention-based decoder to generate diversified paraphrases from inputs and retrieved knowledge. For instance, as shown in Fig. 1, we wish the related concepts “*optimal*”, “*ideal*”, “*get over*”, “*beat*” can be generated in outputs to improve the diversity of expression forms.

The contributions can be summarized as follows:

- We propose a **KnowledgE-Enhanced Paraphraser (KEEP)** to generate diversified paraphrases.
- We guide the information propagation in the knowledge graph with graph attention by scattering current paraphrases focuses to other related concepts.
- Extensive experiments demonstrate that our proposed model can generate more diversified paraphrases compared with baselines while retaining the same semantics.

2 Related Works

2.1 Neural Paraphrase Generation

Seq2Seq models have been widely used in the task of paraphrase generation. Prakash et al. [21] first adapt a neural approach to paraphrase generation with a residual stacked LSTM network. Gupta et al. [9] combine a variational auto-encoder with a Seq2Seq model to generate multiple paraphrases for a given sentence. Kajiwara [11] proposes a neural model for paraphrase generation that first identifies words in the source sentence that should be paraphrased and then conducts the negative lexically constrained decoding that avoids outputting these words. Kazemnejad et al. [12] propose a novel retrieval-based method by editing inputs using the extracted relations between the retrieved pair of sentences for diversified paraphrases. There are also some translation-based methods for paraphrase generation [8]. The main principle of these methods is to translate the text into another language and back to the source language. The above methods mainly focus on fidelity while ignoring the diversity of outputs. Although some works [11, 12] can improve the diversity of paraphrases, they are still based on the scarce corpus data.

2.2 Knowledge-Enhanced Generation

Recently, pre-trained language models (PLMs) such as BERT [6], GPT-2 [22] and BART [14] have further promoted the study on natural language generation (NLG). However, implicit knowledge in PLMs is not enough to help us generate diversified outputs. Incorporating explicit knowledge in Natural Language Generation (NLG) beyond input text is seen as a promising direction in both academia and industry [28]. The introduction of knowledge has also been studied in many NLG tasks, e.g., question generation [2, 23], abstractive text summarization [7], story generation [27] and so on. There are also efforts to exploit external knowledge in paraphrase generation. Huang et al. [10] employ an external synonym dictionary to conduct rewriting on the source sentence to generate paraphrase sentences. Liu et al. [16] incorporate topic words into the Seq2Seq framework to provide auxiliary guidance for paraphrase generation. Different from previous research, our model introduces richer knowledge explicitly with the commonsense knowledge graph and presents a novel attention mechanism on all concepts in the latent concept space for diversified paraphrase generation.

3 Our Approach

In this section, we present the proposed model KEEP (Fig. 2). We first retrieve related concepts in the knowledge graph to construct the *one-hop concept graph* and the *two-hop concept graph*. Then we encode the input sentence, the *one-hop concept graph*, and the *two-hop concept graph* into hidden representations respectively. Finally, we use an attention-based decoder to generate diversified paraphrases. The task can be formulated as: given an input sentence $x = \{x_1, x_2, \dots, x_n\}$, we seek to generate a set of k paraphrase sentences $Y = \{y^{(1)}, y^{(2)}, \dots, y^{(k)}\}$, that all $y \in Y$ have the same meaning with x , but are different in expression forms.

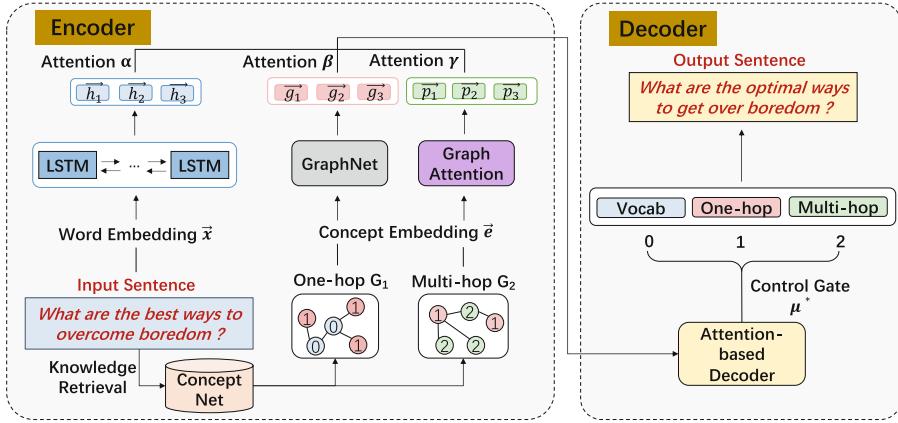


Fig. 2. Architecture of KEEP. Our model consists of an Encoder (Left) and a Decoder (Right). The Encoder encodes the input sentence, the one-hop concept graph and the two-hop concept graph into hidden representations respectively.

3.1 Knowledge Retrieval

Our model relies on the observation that humans usually write paraphrase sentences by replacing words or phrases in the original sentence with their corresponding synonyms or other related words. Therefore, the first step of our method is to retrieve some lexical or phrasal knowledge relevant to the original sentence. We extract a one-hop concept graph and a two-hop graph from a large knowledge graph to guide the paraphrase generation. We grow zero-hop concepts V^0 , which appear in the input sentence and are annotated by entity linking systems, with one-hop concepts V^1 and two-hop concepts V^2 . The concepts in $V^0 \cup V^1$ and relations between them form the one-hop concept graph \mathbb{G}_1 . Also, the two-hop concept graph \mathbb{G}_2 is the knowledge sub-graph induced by $V^1 \cup V^2$.

3.2 Paraphrases and Latent Concept Space Encoding

In this section, we introduce how to encode the input sentence and the KG sub-graphs retrieved in Sect. 3.1.

We use the Bidirectional Long Short Term Memory (Bi-LSTM) as the basic building blocks for Seq2Seq model. Given an input sentence $\{x_1, x_2, \dots, x_n\}$, the LSTM encoder converts it into a set of hidden embeddings $\mathbf{H} = \{h_1, h_2, \dots, h_n\}$. The one-hop concept graph \mathbb{G}_1 is encoded by a graph neural network that propagates information from the input sentence \mathbf{H} to the one-hop concept graph. We choose GraphNet [24] here, since it shows strong effectiveness in encoding knowledge graphs. The l -th layer representation $\mathbf{g}_{e_i}^l$ of concept e_i is calculated by a single-layer feed-forward network (FFN):

$$\mathbf{g}_{e_i}^l = FFN(\mathbf{g}_{e_i}^{l-1} \circ \mathbf{h}^{l-1} \circ \sum_r \sum_{e_j} f_r^{e_j \rightarrow e_i}(\mathbf{g}_{e_j}^{l-1})) \quad (1)$$

where \circ is a concatenation operator and $\mathbf{g}_{e_i}^{l-1}$ is the $(l-1)$ -th layer representation of concept e_i . $f_r^{e_j \rightarrow e_i}(\mathbf{g}_{e_j}^{l-1})$ aggregates the concept semantics of each neighbor concept e_j with relation r . \mathbf{h}^{l-1} is the $(l-1)$ -th layer representation of the input, which is updated with the zero-hop concepts V^0 :

$$\mathbf{h}^{l-1} = FFN\left(\sum_{e_i \in V^0} \mathbf{g}_{e_i}^{l-1}\right) \quad (2)$$

$\mathbf{g}_{e_i}^0$ is initialized with the pre-trained concept embedding \mathbf{e}_i . The input representation \mathbf{h}^0 is initialized with the n -th hidden state \mathbf{h}_n from the input representation set H .

For the two-hop concept graph \mathbb{G}_2 , it is hard to utilize all the concepts and we hope to pay more attention to the more related concepts. To this end, we adopt a novel graph attention mechanism to aggregate concept information. The representation $\mathbf{p}_{\mathbf{e}_q}$, hopping from $e_q \in V^1$ to its connected two-hop concepts e_k , is encoded by an attention mechanism:

$$\mathbf{p}_{\mathbf{e}_q} = \sum_{e_k} \eta_r^{e_k} \cdot [\mathbf{e}_q \circ \mathbf{e}_k] \quad (3)$$

where \mathbf{r} is the relation embedding between the concept $e_q \in V^1$ and its neighbor concept $e_k \in V^2$. \mathbf{e}_q and \mathbf{e}_k are embeddings for concept e_q and concept e_k . The attention $\eta_r^{e_k}$ is calculated as:

$$\eta_r^{e_k} = softmax((\mathbf{W}_r \cdot \mathbf{r})^T \cdot \tanh(\mathbf{W}_q \cdot \mathbf{e}_q + \mathbf{W}_k \cdot \mathbf{e}_k)) \quad (4)$$

where $\mathbf{W}_r, \mathbf{W}_q, \mathbf{W}_k$ are training parameters.

3.3 Diversified Generation

In this section, we use an attention-based decoder to generate diversified paraphrases based on the hidden representations of the input and KG sub-graphs encoded in Sect. 3.2.

We use an attention-based LSTM decoder. The t -step decoder state \mathbf{s}_t is updated by \mathbf{s}_{t-1} , the context representation \mathbf{c}_{t-1} and the word embedding \mathbf{y}_{t-1} of the previous token y_{t-1} :

$$\mathbf{s}_t = LSTM(\mathbf{s}_{t-1}, [\mathbf{c}_{t-1} \circ \mathbf{y}_{t-1}]) \quad (5)$$

where \circ is a concatenation operator.

The context representation \mathbf{c}_{t-1} reads the hidden representations of the input, the one-hop concept graph and the two-hop concept graph with a standard attention mechanism respectively:

$$\mathbf{c}_{t-1} = FFN\left(\left(\sum_{i=1}^n \alpha_{t-1}^i \cdot \mathbf{h}_i\right) \circ \left(\sum_{e_i \in \mathbb{G}_1} \beta_{t-1}^{e_i} \cdot \mathbf{g}_{e_i}\right) \circ \left(\sum_{e_q \in \mathbb{G}_2 \cap V^1} \gamma_{t-1}^{e_q} \cdot \mathbf{p}_{\mathbf{e}_q}\right)\right) \quad (6)$$

The attention weights are calculated over the hidden embedding \mathbf{h}_i of the input, the one-hop concept graph representation \mathbf{g}_{e_i} and the two-hop graph representation $\mathbf{p}_{\mathbf{e}_q}$ of $e_q \in \mathbb{G}_2 \cap V^1$ aggregating two-hop neighbor concepts e_k :

$$\begin{aligned}\alpha_{t-1}^i &= \text{softmax}(\mathbf{s}_{t-1} \cdot \mathbf{h}_i) \\ \beta_{t-1}^{e_i} &= \text{softmax}(\mathbf{s}_{t-1} \cdot \mathbf{g}_{e_i}) \\ \gamma_{t-1}^{e_q} &= \text{softmax}(\mathbf{s}_{t-1} \cdot \mathbf{p}_{e_q})\end{aligned}\quad (7)$$

Finally, we hope that the outputs include tokens from different sources. So we use a control gate μ^* to control the generation by choosing words from vocabulary ($\mu^* = 0$), the one-hop concept graph ($\mu^* = 1, V^0 \cup V^1$) and the two-hop concept graph ($\mu^* = 2, V^2$).

$$\mu^* = \arg \max_{\mu \in \{0,1,2\}} \text{FFN}_\mu(\mathbf{s}_t) \quad (8)$$

The generation probabilities of words w , concepts e_i in \mathbb{G}_1 and multi-hop concepts e_k are computed as follows:

$$y_t = \begin{cases} \text{softmax}(\mathbf{s}_t \cdot \mathbf{w}), & \mu^* = 0 \\ \text{softmax}(\mathbf{s}_t \cdot \mathbf{g}_{e_i}), & \mu^* = 1 \\ \text{softmax}(\mathbf{s}_t \cdot \mathbf{e}_k), & \mu^* = 2 \end{cases} \quad (9)$$

where \mathbf{w} is the word embedding of word w , \mathbf{g}_{e_i} is the one-hop concept graph representation of $e_i \in \mathbb{G}_1$ and \mathbf{e}_k is the concept embedding of the two-hop neighbor concept e_k . We then train our model using standard cross-entropy loss defined in Eq. 10:

$$\mathcal{L} = - \sum_t \log p(y_t^* | y_{<t}, X) \quad (10)$$

where y^* is the actual target sequence.

4 Experiments

4.1 Dataset

We conduct experiments on two of the most frequently used datasets for paraphrase generation: Quora¹ and MSCOCO [15]. We use ConceptNet as the knowledge graph, which contains 120,850 triples, 21,471 concepts and 44 relation types.

Quora. Quora dataset consists of over 400k potential question duplicate pairs. We use true examples of duplicate pairs as paraphrase generation dataset (150K such questions). We sample 100k, 30k, 3k instances for train, test, and validation sets, respectively.

MSCOCO. MSCOCO is a large-scale captioning dataset. This dataset contains over 82k training and 42k validation images, and each image has five captions from five different annotators. We consider different captions of the same image as paraphrases. 20k instances are randomly selected from the data for testing, 10k instances for validation, and remaining data over 320k instances for training.

¹ <https://www.kaggle.com/c/quora-question-pairs>.

4.2 Experimental Setup

Implement Details. We take the top 100k most frequent words as vocabulary from the paraphrases. Glove [20] embedding and TransE [3] embedding are used to initialize the representations of the words and concepts in KG. We use the embedding size of 128 and the batch size of 32. Word embeddings are shared between encoder and decoder. The hidden size is set to 128. We use Adam optimizer [13] with a learning rate of 0.001 to train the parameters and train for 10 epochs on an RTX3090 GPU.

Evaluation Metric. We adopt **BLEU** [18] metric, which is widely used in generation tasks. Considering the limitations of this metric in evaluating the quality of generation, we use more metrics for diversity evaluation. We calculate **Self-BLEU** and **P-BLEU** of results regarding one generated paraphrase as the hypothesis and the others as references. We also calculate the **BERTScore** [29] between the generated paraphrase and the source sentence. We use the BLEU-4 score to compute. For the human evaluation metric, we ask 10 raters to score on 200 generation results, and each result will be evaluated by 5 raters. We ask the human annotators to score the outputs individually based on the following three criteria by using a 5-scale rating for each criterion.: 1) **Fluency**, 2) **Coherency**, 3) **Diversity**. The inter-annotator agreement measured by Spearman’s rank score of around 0.7 shows a good correlation between the raters.

Baselines. We compare our model with the following baselines:

- **Transformer** [25] is a generative model based solely on attention mechanisms.
- **Transformer + KG** joins knowledge and the input sentence together as the input of the model.
- **DicEdit** [10] is a novel approach to model the process with dictionary-guided editing networks.
- **VAE-SVG** [9] is based on a combination of deep generative models (VAE) with sequence-to-sequence models (LSTM) to generate paraphrases.
- **DivGAN** [4] proposes a diversity loss term to make the generator sensitive to the change of latent codes for diversified paraphrase generation.
- **BART** [14] is a denoising autoencoder for pre-trained Seq2Seq models. **BART+KG** incorporates concepts as additional inputs after the input sentence.
- **FSET** [12] a novel retrieval-based method for paraphrase generation by editing inputs using the extracted relations between the retrieved pair of sentences.

4.3 Results

The results of different models on Quora and MSCOCO datasets are shown in Table 1. Our proposed model KEEP outperforms all generative models on most metrics. In terms of BLEU score, KEEP increases 3.53 points compared to Transformer. This indicates our model can generate fluent and accurate paraphrases. What’s more, our model demonstrates a strong ability for diversified paraphrase generation. The Self-BLEU and P-BLEU scores significantly decrease in our model. Although DivGAN

Table 1. Automatic evaluation results from different models. **BL** is short for BLEU. Significant improvements over the best baseline are marked with * (Wilcoxon signed-rank test, $p < 0.01$).

Model	Quora				MSCOCO			
	BL	Self-BL	P-BL	BERTScore	BL	Self-BL	P-BL	BERTScore
Transformer [25]	30.59	42.30	49.69	80.69	22.06	9.44	49.26	66.86
Transformer+KG [25]	31.02	40.15	47.84	79.87	23.54	9.32	44.13	65.73
VAE-SVG [9]	32.00	37.53	44.42	79.44	23.90	9.28	35.10	61.74
DivGAN [4]	31.56	34.31	43.88	81.08	24.06	10.51	34.98	66.70
DicEdit [10]	31.24	36.85	43.68	77.55	24.61	9.11	34.67	60.12
BART [14]	33.36	38.06	45.71	81.12	25.87	9.36	46.78	66.98
BART+KG [14]	33.58	37.45	44.23	80.61	26.03	9.12	40.67	65.72
FSET [12]	33.46	32.89	41.96	75.94	25.24	9.01	34.62	59.87
KEEP (Ours)	34.12	30.69*	40.25	78.23	26.58	8.55	32.58*	64.08

Table 2. Human evaluation results. Our model performs better than other baseline models.

Model	Quora			MSCOCO		
	Fluency	Coherency	Diversity	Fluency	Coherency	Diversity
Transformer+KG [25]	4.12	4.58	2.68	4.27	4.33	2.98
VAE-SVG [9]	4.08	4.52	3.04	4.25	4.27	3.25
DivGAN [4]	4.11	4.46	3.10	4.28	4.28	3.28
DicEdit [10]	4.13	4.45	3.12	4.28	4.25	3.36
BART+KG [14]	4.15	4.61	3.03	4.30	4.38	3.26
FSET [12]	4.18	4.48	3.26	4.31	4.28	3.38
KEEP (Ours)	4.21	4.55	3.67	4.33	4.35	3.77

and FSET also adopt special mechanisms to generate various outputs, KEEP achieves lower Self-BLEU and P-BLEU than DivGAN and FSET. KEEP also performs better than Transformer+KG and BART+KG, which means our model can better incorporate knowledge to improve the diversity of outputs. In terms of BERTScore, it can be seen that our model achieves higher scores than other diversity-based models (e.g., FSET, DicEdit). Although the paraphrases generated by our model are more different from input sentences than BART, the quality of these paraphrases is still good. Furthermore, paraphrase generation means that the morphology is different from the original sentence while maintaining the same meanings.

Human evaluation results are illustrated in Table 2. Generally, our model KEEP achieves high scores on almost all the metrics. Specially, we observe that our model greatly improves the diversity of the generated paraphrases. Comparing KEEP with FSET, the p -value of Wilcoxon signed-rank testing at 95% confidence level is $3.2e-3$, which means the improvements achieved by our approach are statistically significant. Furthermore, to better evaluate the quality and diversity of outputs, we ask five human annotators to make one-on-one comparisons on the groups of generated paraphrases (100 sentences randomly from the test set of the Quora dataset). As shown in Fig. 3, our

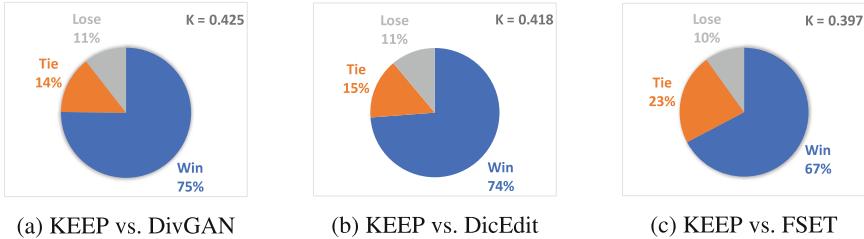


Fig. 3. Results of the one-on-one human evaluation, where KEP clearly wins compared with other models.

Table 3. Ablation study of KEP on the Quora dataset.

Ablation	BL	Self-BL	P-BL
KEP	34.12	30.69	40.25
w/o Two-hop concepts	34.01	33.94	43.27
w/o Concept knowledge	30.62	41.56	48.93
w/o Control gate	33.43	34.81	43.68

model wins in most cases, which means our model KEP can generate higher quality and more diversified paraphrases. Moreover, the inter-annotator agreement measured by Cohen’s kappa K shows fair agreement between raters assessing the models.

4.4 Ablation Study

In order to further evaluate the role of each module in our model, we train and assess different variants: *w/o Two-hop Concepts*: The variant removes the two-hop concept graph and only uses one-hop concepts. *w/o Concept Knowledge*: The variant removes the incorporation of knowledge, including the one-hop concept graph and the two-hop concept graph. *w/o Control Gate*: The variant removes the control gate mechanism which can generate words from different sources.

Table 3 presents the performance comparison. We can see that removing two-hop concepts decreases the performance, especially reduces the diversity of the outputs. This indicates the necessity of integrating two-hop concepts. Furthermore, the model which removes knowledge significantly affects the performance of our model, which further verifies the usefulness of KG data. Finally, removing the control gate mechanism also gives a worse result, which implies the model needs this mechanism to generate tokens from different sources for diversified generation.

Table 4. Case Study. These are paraphrases generated by different models from the Quora dataset. Some unique expressions are marked blue.

Model	Paraphrases
Transformer+KG	1) Can you dream while awake? 2) Can you dream while you are awake? 3) Do you dream while awake?
VAE-SVG	1) Can you dream when you are awake? 2) Do you dream while awake? 3) Can you dream when you wake up ?
DivGAN	1) How do you dream while you are awake? 2) Is it possible to dream while you have awake? 3) Do you dream while awake?
BART+KG	1) Can you dream while you are awake? 2) How do you dream while awake? 3) What are some ways to dream while awake?
FSET	1) how can you dream while awake? 2) Are there some ways for you to dream while awake? 3) How do you dream while you are awake?
KEEP	1) Can humans dream while they are awake? 2) Are there some methods for you to dream when you wake up ? 3) How do you dream while opening your eyes ?

4.5 Case Study

Table 4 shows some examples of the paraphrases. The source text is “*can you dream while awake?*” and the reference is “*can people dream while they are awake?*”. We observe that the paraphrases generated by Transformer+KG are highly similar with minor modifications. What’s more, VAE-SVG, DivGAN and BART+KG can produce more diverse outputs. FSET is able to change the syntactic forms of sentences correctly (replacing “*can you*” with “*are there some ways for you*”). Finally, we find that KEEP can generate high-quality and diversified outputs, which can replace words with their related knowledge (replacing “*awake*” with “*wake up*” or “*open your eyes*”). Especially, it can generate “*can human dream while they are awake*” that is of high similarity to the reference. Note that “*human*” is the two-hop concept of “*you*” in the knowledge graph. Furthermore, since we bring rich knowledge into our model, KEEP can generate more diversified expression forms at the syntactic level, such as “*are there some methods*”.

5 Conclusion

In this paper, we target diversified paraphrasing with the help of the knowledge graph and propose KEEP for this task. To improve the diversity of expression forms in outputs, we introduce related knowledge to enrich the token choices in generated paraphrases. The graph attention mechanism can effectively utilize highly related concepts. Experimental results demonstrate the effectiveness of the proposed knowledge-enhanced paraphrase generation. Detailed analysis shows that our model can better incorporate knowledge, which greatly increases the diversity of generated paraphrases. Future work can adapt this knowledge-enhanced method for other learning tasks or explore how to better combine knowledge with pre-trained generative language models for this task.

Acknowledgements. We thank the anonymous reviewers for their valuable comments and suggestions. This work is supported by Shanghai Science and Technology Innovation Action Plan (No. 19511120400).

References

- Berant, J., Liang, P.: Semantic parsing via paraphrasing. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1415–1425 (2014)
- Bi, S., Cheng, X., Li, Y.F., Wang, Y., Qi, G.: Knowledge-enriched, type-constrained and grammar-guided question generation over knowledge bases. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 2776–2786 (2020)
- Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Neural Information Processing Systems (NIPS), pp. 1–9 (2013)
- Cao, Y., Wan, X.: DivGAN: towards diverse paraphrase generation via diversified generative adversarial network. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pp. 2411–2421 (2020)
- Chen, M., Tang, Q., Wiseman, S., Gimpel, K.: Controllable paraphrase generation with a syntactic exemplar. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 5972–5984 (2019)
- Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) (2018)
- Du, Q.: A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization. In: IJCAI (2018)
- Guo, Y., Liao, Y., Jiang, X., Zhang, Q., Zhang, Y., Liu, Q.: Zero-shot paraphrase generation with multilingual language models. arXiv preprint [arXiv:1911.03597](https://arxiv.org/abs/1911.03597) (2019)
- Gupta, A., Agarwal, A., Singh, P., Rai, P.: A deep generative framework for paraphrase generation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
- Huang, S., Wu, Y., Wei, F., Luan, Z.: Dictionary-guided editing networks for paraphrase generation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 6546–6553 (2019)
- Kajiwara, T.: Negative lexically constrained decoding for paraphrase generation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 6047–6052 (2019)

12. Kazemnejad, A., Salehi, M., Baghshah, M.S.: Paraphrase generation by learning how to edit from samples. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6010–6021 (2020)
13. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: ICLR (Poster) (2015)
14. Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7871–7880 (2020)
15. Lin, T.Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
16. Liu, Y., Lin, Z., Liu, F., Dai, Q., Wang, W.: Generating paraphrase with topic as prior knowledge. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 2381–2384 (2019)
17. Narayan, S., Reddy, S., Cohen, S.B.: Paraphrase generation from latent-variable PCFGs for semantic parsing. arXiv preprint [arXiv:1601.06068](https://arxiv.org/abs/1601.06068) (2016)
18. 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 of the Association for Computational Linguistics, pp. 311–318 (2002)
19. Park, S., et al.: Paraphrase diversification using counterfactual debiasing. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 6883–6891 (2019)
20. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543 (2014)
21. Prakash, A., et al.: Neural paraphrase generation with stacked residual LSTM networks. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 2923–2934 (2016)
22. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI blog 1(8), 9 (2019)
23. Saxena, A., Tripathi, A., Talukdar, P.: Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 4498–4507 (2020)
24. Sun, H., Dhingra, B., Zaheer, M., Mazaitis, K., Salakhutdinov, R., Cohen, W.: Open domain question answering using early fusion of knowledge bases and text. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 4231–4242 (2018)
25. Vaswani, A., et al.: Attention is all you need. In: NIPS (2017)
26. Wang, S., Gupta, R., Chang, N., Baldridge, J.: A task in a suit and a tie: paraphrase generation with semantic augmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 7176–7183 (2019)
27. Xu, P., et al.: MEGATRON-CNTRL: controllable story generation with external knowledge using large-scale language models. arXiv preprint [arXiv:2010.00840](https://arxiv.org/abs/2010.00840) (2020)
28. Yu, W., et al.: A survey of knowledge-enhanced text generation. arXiv preprint [arXiv:2010.04389](https://arxiv.org/abs/2010.04389) (2020)
29. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: BERTScore: evaluating text generation with BERT. arXiv preprint [arXiv:1904.09675](https://arxiv.org/abs/1904.09675) (2019)
30. Zhao, S., Lan, X., Liu, T., Li, S.: Application-driven statistical paraphrase generation. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, pp. 834–842 (2009)



Explore Coarse-Grained Structures for Syntactically Controllable Paraphrase Generation

Erguang Yang¹, Mingtong Liu¹, Deyi Xiong², Yujie Zhang^{1(✉)}, Yao Meng³, Changjian Hu³, Jinan Xu¹, and Yufeng Chen¹

¹ School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China
`{egyang,mingtongliu,yjzhang,jaxu,yfchen}@bjtu.edu.cn`

² College of Intelligence and Computing, Tianjin University, Tianjin, China
`dyxiong@tju.edu.cn`

³ Lenovo Research AI Lab, Beijing, China
`{mengyao1,hucj1}@lenovo.com`

Abstract. Syntactically controlled paraphrase generation can produce diverse paraphrases by exposing syntactic control, where both semantic preservation and syntactic variations are two important factors. Previous works mainly focus on using fine-grained syntactic structures (e.g., full parse tree) as syntactic control. While these methods can achieve excellent syntactic controllability, leads to failing to preserve the semantics of the input sentence. The main reason is that it is difficult to retrieve perfectly compatible syntactic structures with the input sentences. In this paper, we explore coarse-grained syntactic structures to trade-off semantic preservation and syntactic variations. Furthermore, to improve semantic preservation and syntactic controllability, we propose a **Syntax Attention-Guided Paraphrase** (SAGP) model that can correctly select syntactic information according to the current state for surface realization. Experiment results show that SAGP outperforms the previous state-of-the-art method under the same setting. Additionally, we validate that using coarse-grained structures can generate more semantically reasonable text without affecting the syntactic controllability.

Keywords: Coarse-grained structure · Syntactic controllability · Semantic preservation

1 Introduction

Paraphrases are defined as sentences conveying the same meaning but with different surface realization. Paraphrase generation is of great importance to many downstream tasks in natural language processing, such as question answering [6], machine translation [22] and text summarization [21]. Since most existing state-of-the-art paraphrase generation models fail to produce paraphrases with syntactic diversity [7, 10, 16], recent works have started exploring syntactically controllable paraphrase generation [5], i.e., given an input sentence and an exemplar, produce a sentence which follows the syntax of the exemplar and the meaning of the input sentence.

Source	It is hard for me to imagine where they could be hiding it underground .
Exemplar	They ca n't imagine when he 'll be able to walk . PRP MD RB VB WRB PRP MD VB JJ TO VB .
Reference	I ca n't imagine where they could be hiding it underground .
SCPN(2018)	You ca n't imagine where they might be hidden to underground . PRP MD RB VB WRB PRP MD VB VBN TO NN .
CHEN(2019)	It could really be where it could be dangerous to hide . PRP MD RB VB WRB PRP MD VB JJ TO VB .
LIU(2020)	You ca n't imagine where he might be able to hide . PRP MD RB VB WRB PRP MD VB JJ TO VB .
SGCP(2020)	It can n't imagine where they might be able to hide . PRP MD RB VB WRB PRP MD VB JJ TO VB .

Fig. 1. Examples of syntactic paraphrases generated by previous works. SCPN [8] and SGCP [9] use the parse tree of the exemplar as syntactic control, and CHEN [5] and LIU [12] use the exemplar itself. Part-of-speech tags of the exemplar and generated sentences are obtained using the Stanford CoreNLP toolkit [14]. The part-of-speech sequence can reflect the similarity of syntactic structure. Blue indicates that the tag is the same as the exemplar, while red is the opposite.

In this aspect, Iyyer et al. [8] use an attentional seq2seq network to encode the input sentence and a linearized full parse tree to generate paraphrases. Chen et al. [5] and Liu et al. [12] directly use the exemplar itself as syntactic control, where a latent variable is designed to capture the syntactic style of the exemplar, and then the latent variable is fed into the decoder to guide the generation of paraphrases. Kumar et al. [9] encode the parse tree in a top-down manner and generate paraphrases through a queue-based decoding mechanism.

However, existing methods often use fine-grained syntactic structure (e.g., full parse tree) as syntactic control, which leads to the generated sentences not keeping their original semantics. We can observe in Fig. 1 that the generated sentences follow the length and part-of-speech sequence of the exemplar, but do not preserve the semantics of the input sentence. Intuitively, the fine-grained syntactic structure is relatively specific, which will sharply reduce the space of semantic adjustment. When the exemplar and the input sentence are not compatible, it will lead to failing to preserve the semantics of the input sentence.

Additionally, using fine-grained syntactic structure poses the problem of how to effectively retrieve compatible exemplars. In the application scenario, it is difficult to find such an exemplar that is syntactically perfectly compatible with the input sentence.

In this work, we explore coarse-grained syntactic structures to trade-off semantic preservation and syntactic variations. We remove the part-of-speech nodes and then extract the high-level sub-tree of the constituency parse tree to obtain relatively coarse-grained syntactic structures. Furthermore, to improve the semantic preservation and syntactic controllability, we propose a **Syntax Attention-Guided Paraphrase** (SAGP) model which mainly contains as follows: a content enhanced sentence encoder that can make the generated sentence better preserve the key information of input sentence; a syntactic encoder that can effectively encode syntactic structure from both top-down and left-to-right directions; and a regularized syntax attention that can select an appropriate syntactic constituent to control the generation of words.

The main contributions of this work are as follows:

1. We explore coarse-grained syntactic structures to trade-off semantic preservation and syntactic variations, which can alleviate the issue of incompatibility between the syntactic exemplar and the input sentence.
2. To improve the semantic preservation and syntactic controllability, we propose a **Syntax Attention Guided Paraphrase** (SAGP) model.
3. Experiments show that our proposed model substantially outperforms previous state-of-the-art approaches under the same setting. Additionally, we demonstrate that the proposed method can produce diverse paraphrases that conform to the syntax of exemplars and preserve the semantics of the input sentence at the same time.

2 Related Work

We focus primarily on the task of syntactically controllable paraphrase generation, which has received significant recent attention [5, 8, 9, 12]. To address this task, Iyyer et al. [8] use an extended pointer-generator network [18] to encode input sentences and linearized parse trees to generate paraphrases. Our model encodes the tree structure from top-down and left-to-right directions, which can more effectively encode structure information. Moreover, to achieve better syntactic controllability, we propose a regularized syntax attention that can correctly select syntactic constituent.

Chen et al. [5] and Liu et al. [12] use sentential exemplar as syntactic control. They design a latent variable to learn the syntax by encoding the exemplar itself, and then the syntactic variable is used to guide the generation of paraphrases. Word noising and mask scheme are respectively proposed to effectively train their model. However, there are two common problems: (1) It is difficult for the syntactic variable to capture the syntactic structure of the exemplar. (2) Word noising or mask schemes make their model only learn the ability of lexical substitution based on exemplar. Compared to these two approaches, we use an explicit parse tree as syntactic control instead of exemplar itself, which can avoid the above two problems.

Kumar et al. [9] propose using the full or low-level parse tree of the exemplar as syntactic control. They encode the parse tree in a top-down manner based on the RNN network. They also propose a queue-based decoding mechanism to incorporate syntactic information. Their work is different from ours in at least three aspects: (1) Our work explores primarily coarse-grained structures to trade-off semantic preservation and syntactic variations. (2) The top-down encoding manner makes their model only encode parent-child relations ignore the sibling relations that are essential for syntactic controllability. In contrast, our method makes up for this deficiency by using left-to-right encoding. (3) The queue-based decoding mechanism only uses the head node to predict the control node. We use the regularized attention mechanism which can select a more correct syntactic node by comparing attention weights of all syntactic nodes.

Another related line of works produces syntactic paraphrases via the unsupervised method [3, 20]. Their models use two variational autoencoders [4] to introduce two latent variables which are designed to capture semantics and syntax, respectively. These methods cannot effectively model structure information of parse tree.

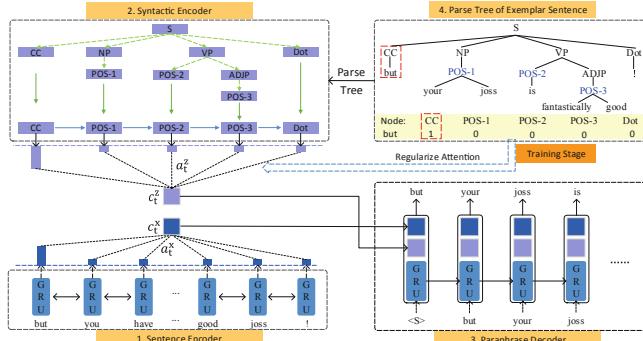


Fig. 2. The architecture of SAGP. For a detailed description, please refer to Sect. 3. (Color figure online)

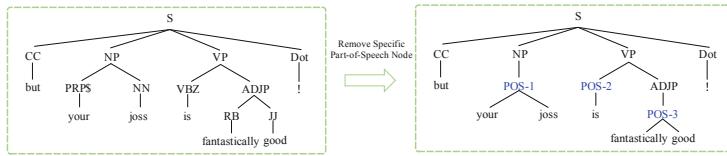


Fig. 3. Remove the part-of-speech nodes below level 2. Note that the numbers are just for differentiating POS in writing.

3 Our Approach

3.1 Problem Formalization

We formulate the problem of syntactically controllable paraphrase generation as follows. Given two sentences (x, z) as input, we would like to generate sentence y to convey the meaning of sentence x and conform to the syntactic structure of sentence z . The x and z are called the semantic input and syntactic exemplar, respectively.

3.2 Overall Architecture

As shown in Fig. 2, the proposed model is built on the standard encoder-decoder architecture with an attention mechanism [1], which mainly contains a sentence encoder, a tree encoder and a paraphrase decoder. As for the attention mechanism, we use the global attention method proposed by [13] with the general alignment function.

Concretely, we use the paraphrase parallel sentence pair (x, y) to train our model. At the training stage, we set $z = y$ to avoid constructing the additional syntactic exemplars. Syntactic control is provided by full parse tree¹ without leaf node (i.e., words) of syntactic exemplar. To obtain a coarse-grained structure, we first remove the part-of-speech nodes below level 2, which means that POS nodes higher than level 2 will be

¹ Obtained using the Stanford CoreNLP [14].

kept (e.g., the CC node), as shown in Fig. 3, and then prune the parse tree to height as $\{3, 4, 5, 6, 7\}$. Then given (x, p_z) , we train the model to predict y , where p_z denotes the processed parse tree of syntactic exemplar, x and p_z are input to the sentence encoder and the syntactic encoder.

3.3 Sentence Encoder

The sentence encoder is a Bi-GRU network. Take the sentence $x = \{x_1, x_2, \dots, x_n\}$ as input, the encoder computes the sentence-side hidden state sequence as $\mathbf{h}_i^x = \text{GRU}(e(x_i), \mathbf{h}_{i-1}^x)$, where the $e(x_i)$ and \mathbf{h}_i^x denote the embedding vector and the hidden state of the word x_i , respectively.

Additionally, we propose a content enhanced method to improve the ability to preserve the semantics of content words that express more important meanings than other words. We recognize a fixed percentage N (40% in the experiment) of words with high TF-IDF scores in the sentence as content words sequence $c = \{c_1, c_2, \dots\}$, and then feed the sequence into the shared sentence encoder to obtain the content-side hidden state \mathbf{h}_i^c in the same manner, where \mathbf{h}_i^c denotes the hidden state of the word c_i .

3.4 Syntactic Encoder

We also use the GRU network to build the syntactic encoder. As shown in Fig. 2(2), we traverse the given parse tree in a top-down (green line) and left-to-right (blue line) to obtain and model parent-child and sibling relationships, respectively.

For the top-down (TD) direction, we encode the parse tree in a depth-first manner. Specifically, the representation \mathbf{h}_v of each node v is calculated by the following:

$$\mathbf{h}_v = \text{GRU}(e(v), \mathbf{h}_{pa(v)}) \quad (1)$$

where the $e(v)$ and $pa(v)$ denote the embedding vector and the parent node of v , respectively. Although we can obtain TD representations of all nodes, only the TD representations of leaf nodes will be used for left-to-right encoding. For the particular example given in Fig. 2(2), the TD representations of all leaf nodes $H_{leaf}^{TD} = [\mathbf{h}_{CC}^{TD}, \mathbf{h}_{POS-1}^{TD}, \mathbf{h}_{POS-2}^{TD}, \mathbf{h}_{POS-3}^{TD}, \mathbf{h}_{Dot}^{TD}]$.

For the left-to-right (LR) encoding, the encoder is a forward GRU network. We take the leaf nodes sequence $Leaf_{seq} = \{CC, POS - 1, POS - 2, POS - 3, Dot\}$ as input, and compute the LR representations of all leaf nodes $H_{leaf}^{LR} = [\mathbf{h}_{CC}^{LR}, \mathbf{h}_{POS-1}^{LR}, \mathbf{h}_{POS-2}^{LR}, \mathbf{h}_{POS-3}^{LR}, \mathbf{h}_{Dot}^{LR}]$. Particularly, take the $POS - 1$ node as an example:

$$\mathbf{h}_{POS-1}^{LR} = \text{GRU}(\mathbf{h}_{POS-1}^{TD}, \mathbf{h}_{CC}^{LR}) \quad (2)$$

then we use the LR representations of all leaf nodes to provide syntactic information for the paraphrase decoder. In the following description, we use \mathbf{h}^z to represent H_{leaf}^{LR} .

3.5 Paraphrase Decoder

The paraphrase decoder is a forward GRU network. Having obtained the sentence, content, and syntax representations, we introduce multiple attention modules to incorporate these representations.

Regularized Syntax Attention. We first enhance the decoder’s hidden state using the syntactic context representation. Specifically, given the target hidden state \mathbf{h}_t and the syntax-side hidden state sequence \mathbf{h}^z , the calculation of syntax attention as follows:

$$\begin{aligned}\mathbf{a}_t^z &= \text{softmax}(\mathbf{h}_t^\top \mathbf{W}_z \mathbf{h}^z) \\ \mathbf{c}_t^z &= \sum_j \mathbf{a}_t^z(j) \mathbf{h}_j^z\end{aligned}\quad (3)$$

where the \mathbf{W}_z is the trainable parameter, \mathbf{a}_t^z denotes the attention weight that can select the appropriate node to control the generation according to the current hidden state. \mathbf{c}_t^z denotes the weighted sum of the syntax-side hidden states.

Then we employ a concatenation layer to obtain the enhanced hidden state as follows:

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_{cz}[\mathbf{c}_t^z; \mathbf{h}_t]) \quad (4)$$

where the \mathbf{W}_{cz} is the trainable parameter, the enhanced hidden state $\tilde{\mathbf{h}}_t$ will be used for the calculations of the following sentence and content attention.

Attention weight \mathbf{a}_t^z is essential for accurate syntactic control. We propose a regularization method to guide the learning of attention using the alignment of nodes and words in a parse tree. For example, as shown in Fig. 2(4), there is an alignment relationship between the word “but” and the node “CC”, therefore the gold attention weight is $(1, 0, 0, 0, 0)$ at the first timestep. So the decoder should focus on the node “CC” when generating the word “but”. Specifically, we propose a regularization loss as follows:

$$\mathcal{L}_{rl} = \sum_{t=0}^T \text{MSE}(\mathbf{a}_t^z, \hat{\mathbf{a}}_t^z) \quad (5)$$

where the \mathbf{a}_t^z is computed by Eq. (3), the $\hat{\mathbf{a}}_t^z$ is the gold attention weight obtained from the alignment of nodes and words in a parse tree. MSE denotes the Mean Square Error objective function. By doing so, we can make learned \mathbf{a}_t^z closed to gold attention distribution. We also use label smoothing (0.1) to reduce the errors caused by parsing.

Sentence and Content Attention. We introduce sentence and content attention to make use of the sentence-side hidden states \mathbf{h}^x and the content-side hidden states \mathbf{h}^c , respectively.

Having obtained the enhanced decoder hidden state $\tilde{\mathbf{h}}_t$, the calculation of sentence attention as follows:

$$\begin{aligned}\mathbf{a}_t^x &= \text{softmax}(\tilde{\mathbf{h}}_t^\top \mathbf{W}_x \mathbf{h}^x) \\ \mathbf{c}_t^x &= \sum_i \mathbf{a}_t^x(i) \mathbf{h}_i^x \\ \tilde{\mathbf{h}}_t^x &= \tanh(\mathbf{W}_{cx}[\mathbf{c}_t^x; \tilde{\mathbf{h}}_t])\end{aligned}\quad (6)$$

where the \mathbf{W}_x and \mathbf{W}_{cx} are trainable parameters. \mathbf{a}_t^x and \mathbf{c}_t^x denote the attention weight and the weighted sum of the sentence-side hidden states, respectively. We define $\tilde{\mathbf{h}}_t^x$ as the sentence-based attentional hidden state.

Likewise, we can obtain the content-based attentional hidden state $\tilde{\mathbf{h}}_t^c$ in the same way, with trainable parameters \mathbf{W}_c and \mathbf{W}_{cc} . Then, We fuse these two hidden states by addition:

$$\mathbf{h}_t^f = \text{relu}(\mathbf{W}_f(\tilde{\mathbf{h}}_t^x + \tilde{\mathbf{h}}_t^c)) \quad (7)$$

where \mathbf{W}_f is the trainable parameter, the fusion representation \mathbf{h}_t^f and the $e(y_{t-1})$ will be used for predicting the word distribution as follows:

$$\mathbf{p} = \text{softmax}(\mathbf{W}_o([\mathbf{h}_t^f; e(y_{t-1})] + \mathbf{b}_o)) \quad (8)$$

where the \mathbf{W}_o and \mathbf{b}_o are trainable parameters. We also use the copy mechanism [18] to augment the model, which can produce OOV words.

3.6 The Overall Objective Function

The overall objective is defined as follow:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{pl} + \lambda_2 \mathcal{L}_{rl} \quad (9)$$

where $\mathcal{L}_{pl} = -\sum_{t=0}^T \log \mathbf{p}(y_t)$ is the cross-entropy loss for ground true y , λ_* are balancing hyper-parameters.

4 Experiments

In this section, we will answer the following questions:

- How does SAGP compare against prior models?
- How do coarse-grained syntactic structures affect semantic preservation and syntactic variations?

4.1 Dataset

We train and evaluate our model on the ParaNMT-small dataset [5, 19]. It contains 500k paraphrase pairs for training, 500 and 800 manually labeled tuples (sentence-exemplar-reference) for development set and test set.

4.2 Experiment Setup

In this experiment, we evaluate four different granularities of syntactic information for controllable paraphrase generation:

- F : Uses full parse tree of the exemplar for controllable paraphrase generation.
- $F(rmpos)$: Uses full parse tree that removes part-of-speech (pos) nodes as syntactic control.

- E : We extract top ht level parse tree for paraphrase generation. By setting $ht = \{3, 4, 5, 6, 7\}$, the model can generate 5 candidate sentences, and then we choose the sentence that has the highest ROUGE-1 score with the input sentence as the final paraphrase.
- $E(rmpos)$: We first remove the part-of-speech nodes of the parse tree, the next processes are consistent with **-E**.

4.3 Automated Evaluation

Semantic Metrics. We compute the BLEU [15], ROUGE [11], and METEOR [2] scores between the generated and the reference paraphrases in the test set. We also used the embedding-based evaluation method Sentence-BERT² [17] to evaluate the semantic similarity between the generated sentence and the original sentence.

Syntactic Metrics. Following previous work [9], we compute the tree edit distance (TED) against the parse trees of the reference and exemplar, denoted as TED-r and TED-e, respectively. But the TED-r and TED-e only evaluate the difference in the full parse tree between two sentences, don't measure the controllability of the model to overall syntactic structure that is also important for generating syntactic paraphrases. We propose the exact syntactic template match (ESTM) automatic evaluation method: a paraphrase g is deemed as an exact syntactic template match to exemplar e only if the top two levels of its parse tree p_g exactly matches those of p_e . We evaluated how often generated paraphrases completely conform to the syntactic templates of exemplars by computing the rate of exact syntactic template match.

4.4 Human Evaluation

We conduct the human evaluation on 100 randomly selected data points from the test set in a blind fashion. Three annotators evaluate the generated paraphrases in terms of semantic preservation (the generation and input sentence) and syntactic controllability (the generation and exemplar); each aspect was scored from 1 to 5.

4.5 Results

As shown in Table 1, according to the used syntactic type, we divided the previous works into two groups and compared with them. Simple outputting the input sentence shows high scores across semantic metrics, but shows worse performance across syntactic metrics. The opposite is true for simple outputting the exemplar. These results help to show the performance of model. We also note that TED-r between the syntactic exemplar and the reference sentence is 5.9, which indicates there are still problems of incompatibility in the manually labeled syntactic exemplars.

² We used the paraphrase-distilroberta-base-v1, which is available at: <https://public.ukp.informatik.tu-darmstadt.de/reimers/sentence-transformers/v0.2/>.

Table 1. Evaluation results on the ParaNMT-small dataset. All our scores are reported as the mean over three runs. R-1, R-2, and R-L denotes the ROUGE-1, ROUGE-2, and ROUGE-L, respectively. S-BERT and ESTM indicate the Sentence-BERT and exact syntactic template match, respectively.

Model	Semantic Metrics						Syntactic Metrics		
	BLEU↑	METEOR↑	R-1↑	R-2↑	R-L↑	S-BERT↑	TED-r↓	TED-e↓	ESTM↑
Return-input baselines									
Semantic input	18.5	28.8	50.6	23.1	47.7	1.0	12.0	13.0	36.9
Syntactic input	3.3	12.1	24.4	7.5	29.1	0.218	5.9	0.0	100
Fine-grained syntactic structure									
CGEN [5]	13.6	24.8	44.8	21.0	48.3	0.531	6.7	3.3	83.3
LIU [12]	14.3	26.2	47.3	22.9	49.3	0.604	6.6	5.5	77.2
SCPN-F [8]	15.35	26.6	47.2	23.1	50.5	0.563	6.6	2.9	86.8
SGCP-F [9]	15.3	25.9	46.6	21.8	49.7	0.560	6.1	1.4	88.9
SAGP-F	17.5	27.0	48.5	24.3	51.7	0.578	5.9	0.9	93.1
SAGP-F(rm pos)	17.5	28.7	50.3	25.7	51.9	0.669	7.0	5.1	94.5
Coarse-grained syntactic structure									
SGCP-R [9]	16.4	27.2	49.6	22.9	50.5	0.664	8.7	7.0	75.5
SAGP-E	21.0	31.6	54.8	28.8	55.8	0.713	7.1	5.7	88.3
SAGP-E(rm pos)	20.4	32.1	55.2	29.3	55.3	0.761	8.1	7.9	93.4

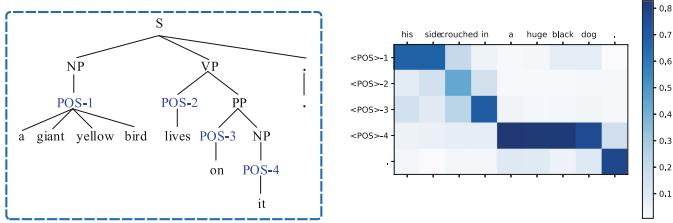
Table 2. Human evaluation scores.

Model	Semantic	Syntactic
CGEN	2.62	4.16
LIU	3.11	3.68
SCPN-F	2.81	4.25
SGCP-F	2.76	4.31
SAGP-F	3.0	4.62
SAGP-F(rm pos)	3.4	4.73
SGCP-R	3.36	3.77
SAGP-E	3.61	4.3
SAGP-E(rm pos)	3.81	4.65

We can observe in Table 1 that SAGP-F obtains the best results across semantic and syntactic metrics under the same setting. These results demonstrate the effectiveness of SAGP in semantic preservation and syntactic controllability.

SGCP-R uses the lower 5 levels of the parse tree as syntactic control, instead, we use the relatively coarse-grained structures. We can see that SAGP-E still achieves substantial improvements across semantic and syntactic metrics. These results further verify the advantages of the proposed model.

It can also be seen that removing the part-of-speech nodes of the full parse tree leads to further gain across the semantic metrics under these two settings, especially



(a) The parse tree without part-of-speech nodes.
(b) Using parse tree without part-of-speech nodes.

Fig. 4. Visualization of syntax attention. (a) shows the parse tree without part-of-speech nodes of exemplar. (b) is visualization result of syntax attention when use processed full parse (remove word) as syntactic control.

Table 3. Effect of different level syntactic structure. POS denotes part-of-speech nodes.

Height	No Remove POS				Remove POS			
	BLEU↑	S-BERT↑	TED-e↓	ESTM↑	BLEU↑	S-BERT↑	TED-e↓	ESTM↑
3	18.8	0.676	7.0	87.8	17.5	0.713	8.6	94.0
4	19.4	0.651	5.3	90	18.7	0.70	7.4	94.8
5	19.2	0.625	3.8	90.4	18.0	0.683	6.5	94.6
6	18.4	0.605	2.8	91.8	17.9	0.680	6.1	94.8
7	17.4	0.593	2	92.3	17.0	0.672	5.6	94.8

S-BERT. These results demonstrate that part-of-speech nodes can bring improvements to the TED metric, but it will harm semantic preservation.

Human Evaluation. Table 2 shows the results of human evaluation which are somewhat consistent with the automated metrics. Our model obtains the highest scores, thereby highlighting the efficacy of our models.

4.6 Model Analysis

Visualization of Syntax Attention. In Fig. 4, we visualize the syntax attention when using the full parse tree without part-of-speech nodes as syntactic control. This is an instance from the test set: {Source: by his side crouched a huge black wolfish dog. Exemplar: a giant yellow bird lives on it. Reference: a huge black wolfish dog squatted down beside him.}

The generated sentence is: his side crouched in a huge black dog. We can see that the words of the generated sentence correctly align with syntactic nodes POS-1-4. The results show that the regularized syntax attention makes the decoder can correctly select syntactic constituents to control the generation of words.

Effect of Different Level Syntactic Structure. We also analyzed the influence of different level structures on the performance of our model. We can see in Table 3 that

Table 4. Example of generated sentences.

Source	I promise to put your dreams before mine
Exemplar	I promise no one can lay their hands on her
Reference	I promise I'll throw out my dreams for yours
Previous works	
SCPN (2018)	I promise you that I'll have my dreams before you
CHEN (2019)	I promise the dream I promised to put on yours
LIU (2020)	I promise the dream that you will dream of mine
SGCP-F (2020)	I promise the you can dream my dreams before me
SGCP-R (2020)	I promise you'd get your dream in front
Ours	
SAGP-F	I promise all you'll have your dreams before me
SAGP-E	I promise you've got a dream before mine
SAGP-E(rm pos)	I swear I'll put your dream before I'm mine

semantic and syntactic metrics are more or less contradictory to each other. High-level structure leading to better content preservation but worse syntactic control, especially TED-e metric. Removing the part-of-speech nodes of the parse tree obtains substantial improvement on S-BERT and ESTM metrics. This is because the part-of-speech information is relatively specific, which will limit the type and number of generated words, thus harm semantic preservation.

Example Generations. Table 4 shows some results generated by the competitive models. We can see that the proposed model can effectively generate semantics-preserved paraphrases that conform to the syntax of exemplar at the same time. The fine-grained syntactic structure strictly limits the output, leads to failing to preserve the semantics of the input sentence. Compare with SGCP, our method can produce better output that conforms to the syntax of the exemplar and the semantics of the input sentence.

5 Conclusion

In this paper, we explore coarse-grained syntactic structures to trade-off semantic preservation and syntactic variations. To improve semantic preservation and syntactic controllability, we further propose a syntax attention-guided paraphrase model that can correctly select a syntactic constituent to control the generation of paraphrases. Experiments show that our model achieves strong improvements over previous methods. Furthermore, we verify that the proposed method is able to generate diverse paraphrases that conform to the syntax of exemplars and the semantics of input sentences. The proposed method can alleviate the issue of incompatibility between the syntactic exemplar and the input sentence.

Acknowledgments. This work is supported by the National Science Foundation of China (Contract 61876198, 61976105, 61976016).

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: ICLR 2015 (2015)
2. Banerjee, S., Lavie, A.: METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In: ACL Workshop, Ann Arbor, Michigan (2005)
3. Bao, Y., et al.: Generating sentences from disentangled syntactic and semantic spaces. In: ACL (2019)
4. Bowman, S.R., Vilnis, L., Vinyals, O., Dai, A., Jozefowicz, R., Bengio, S.: Generating sentences from a continuous space. In: SIGNLL, Berlin, Germany (2016)
5. Chen, M., Tang, Q., Wiseman, S., Gimpel, K.: Controllable paraphrase generation with a syntactic exemplar. In: ACL, Florence, Italy (2019)
6. Dong, L., Mallinson, J., Reddy, S., Lapata, M.: Learning to paraphrase for question answering. In: EMNLP, Copenhagen, Denmark (2017)
7. Gupta, A., Agarwal, A., Singh, P., Rai, P.: A deep generative framework for paraphrase generation. In: AAAI (2018). <https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16353>
8. Iyyer, M., Wieting, J., Gimpel, K., Zettlemoyer, L.: Adversarial example generation with syntactically controlled paraphrase networks. In: NAACL (2018)
9. Kumar, A., Ahuja, K., Vadapalli, R., Talukdar, P.: Syntax-guided controlled generation of paraphrases. TACL **8**, 329–345 (2020). <https://www.aclweb.org/anthology/2020.tacl-1.22>
10. Li, Z., Jiang, X., Shang, L., Li, H.: Paraphrase generation with deep reinforcement learning. In: EMNLP, Brussels, Belgium (2018)
11. Lin, C.Y.: ROUGE: a package for automatic evaluation of summaries. In: Text Summarization Branches Out, Barcelona, Spain (2004)
12. Liu, M., et al.: Exploring bilingual parallel corpora for syntactically controllable paraphrase generation. In: IJCAI-20 (2020)
13. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: EMNLP, Lisbon, Portugal (2015)
14. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: ACL, Baltimore, Maryland (2014)
15. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation (2002)
16. Prakash, A., et al.: Neural paraphrase generation with stacked residual LSTM networks. In: COLING, pp. 2923–2934. The COLING 2016 Organizing Committee, Osaka (2016). <https://www.aclweb.org/anthology/C16-1275>
17. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. In: EMNLP, Hong Kong, China (2019)
18. See, A., Liu, P.J., Manning, C.D.: Get to the point: summarization with pointer-generator networks. In: ACL, Vancouver, Canada (2017)
19. Wieting, J., Gimpel, K.: ParaNMT-50M: pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In: ACL, Melbourne, Australia (2018)
20. Zhang, X., Yang, Y., Yuan, S., Shen, D., Carin, L.: Syntax-infused variational autoencoder for text generation. In: ACL, Florence, Italy (2019)
21. Zhao, S., Meng, R., He, D., Saptono, A., Parmanto, B.: Integrating transformer and paraphrase rules for sentence simplification. In: EMNLP, Brussels, Belgium (2018)
22. Zhou, Z., Sperber, M., Waibel, A.: Paraphrases as foreign languages in multilingual neural machine translation. In: ACL: Student Research Workshop (2019)



Chinese Poetry Generation with Metrical Constraints

Yingfeng Luo¹, Changliang Li³, Canan Huang¹, Chen Xu¹, Xin Zeng¹, Binghao Wei¹,
Tong Xiao^{1,2(✉)}, and Jingbo Zhu^{1,2}

¹ NLP Lab, School of Computer Science and Engineering,
Northeastern University, Shenyang, China

1971646@stu.neu.edu.cn, {xiaotong, zhujingbo}@mail.neu.edu.cn

² NiuTrans Research, Shenyang, China

³ Kingsoft AI Lab, Beijing, China
lichangliang@kingsoft.com

Abstract. Poetry is a kind of literary art, which conveys emotion with aesthetic expressions. Poetry automatic generation is challenging because it is required to confirm the semantic representation (content) and metrical constraints (form). Most previous work lacks the effective use of metrical information, resulting in the generated poems may break these constraints, which are essential for poetry. In this paper, we formulate the poetry generation task as a constrained text generation problem. A Transformer-based dual-encoder model is then proposed to force the poetry generation conditioned on both the writing intention and the metrical patterns. We conduct experiments on three popular genres of Chinese classical poetry: quatrains (绝句), regulated verse (律诗) and Song iambic (宋词). Both automatic and human evaluation results confirm that our method (poetry generation with metrical constraints, **MCPG**) significantly improves metrical complimentance of generated poems while maintaining coherence and fluency.

Keywords: Poetry automatic generation · Natural language generation · Metrical constraints

1 Introduction

Poetry is the spiritual and cultural heritage of humanity. Chinese classical poetry has formed a series of rich genres in the development process of more than 1300 years. These genres have their specific metrical patterns, mainly including the following three aspects:

1. Structure pattern. It strictly defines the number of lines and the number of characters in each line. For example, quatrains contain four lines, with each consisting of five or seven characters, called “五绝” and “七绝” in Chinese respectively.
2. Rhyme pattern. The basic pronunciation units of Chinese characters are initials (声母) and finals (韵母). We call those characters that use the same or similar finals rhymes. The rhyme patterns specify that the last character in certain lines should rhyme. For example, in quatrains, the last character of the first (optional), second and fourth lines must rhyme.

3. Tone pattern. Ancient Chinese characters used five tones, which were further divided into two categories: level tone (平) and downward tone (仄). The tone pattern defines the tone of each position in a poem, which means that each position needs to be filled with characters that match the tone determined by the tone pattern.

The above metrical constraints are sorted by priority. Figure 1 is a famous classical poem of five-characters quatrains (五绝). Ignore the content pattern in the figure for the moment, which is introduced in section Methodology.

Poetry automatic generation task is: given a writing intention and a poetry genre, then output a sequence that meets these requirements. As defined, poetry generation is a constrained text generation with multiple constraints: semantic ones (the content of the poem should be semantically consistent with the writing intention) and form ones (the poem should meet the metrical patterns defined by a given genre). The constraint not only tells what to say (content) but also how to say (form). Most previous work treats poetry generation as free text generation and assumes that the model is able to learn all it needs from the corpus automatically [12, 14, 24, 26, 29, 33, 35]. However, poetry data is not sufficient in most cases, especially given the fact that capturing the variety of such literature is non-trivial. This issue is arguably further amplified when we face Song iambic with complex pattern constraints but few training data [22, 34]. Consequently, the generated poems may break these metrical constraints, which are essential for poetry. In addition, the lack of metrical modelling results in poor portability and generalization ability of the model. Most of the work is designed for the generation of quatrains [12, 24, 26, 29, 33, 35], but these models are not convenient to be used for the generation of other genres such as Song iambic.

On the other hand, incorporating discrete structural information has shown promising results in machine translation [2, 23] and controlled text generation [10, 17]. And very powerful models such as Transformer [20] and variants [21] have emerged. Inspired by these, we use metrical information to improve poetry generation. Note that some work [5, 13, 34] have made attempts along the line of this research, but they just used limited metrical information and restrict themselves to a specific poetry genre. In this work, we investigate more general methods and apply them to more poetry genres.

For this goal, we represent the problem of poetry generation as a constraint satisfaction problem, where we can modify constraints to restrict the genre of poetry we want. This enables us to train all genres of poetry in a unified framework, make full use of different types of data, and further improve the model’s performance. Specifically, we format the different levels of metrical constraints into discrete character sequences. A Transformer-based dual-encoder model is then proposed to force the poetry generation conditioned on both the writing intention and the metrical patterns. Experiments on a classical Chinese poetry corpus demonstrate that our approach significantly improves metrical compliance of generated poems while maintaining coherence and fluency.

Genre&Title	五言绝句·登鹳雀楼
Body	白 日 依 山 尽 , 黄 河 入 海 流 。 欲 穷 千 里 目 , 更 上 一 层 楼 。
Structure pattern	5 , 5 。 5 , 5 。
Tone Pattern	pt pt zt zt pt co zt zt pt pt zt pe pt zt zt pt pt co pt pt pt zt zt pe
Rhyme Pattern	* * * * * co * * * * 10 pe * * * * * co * * * * 10 pe
Content Pattern	* * 依 * * co * 河 * * * pe * * * * co * * * * 层 * pe

Fig. 1. An example of a five characters quatrains. The rhyming characters are red. ‘pt’ and ‘zt’ indicate the level tone (平) and downward tone (仄), ‘co’ and ‘pe’ indicate two kinds of punctuations. The number indicates the rhyme class. For the content pattern, a few characters are randomly selected from the Body during training, and all the positions except punctuations are replaced by placeholder ‘*’ at test time. (Color figure online)

2 Related Work

The use of computers to create art has always been expected. In addition to the long-term goal of building artificial intelligence, research on this kind of task can assist humans in artistic activities. The automatic generation of poetry is one of the research fields widely concerned, and its related research has gone through three stages.

The first stage of poetry generation is based on templates or rules, which has the characteristics of symbolism and rationalism. The system selects words or characters that meet the metrical constraints according to the rules written by experts for text generation [6]. In the second stage, some methods based on statistical machine learning are gradually applied to this task, such as genetic algorithm [16, 36], text summarization [27] and statistical machine translation [7, 8]. These methods focus on the surface forms of words or characters and lack understanding of the intrinsic meaning of poetry. Therefore, although the poems generated could meet the requirements in form, fluency, coherence, and meaning are far from the requirements.

In recent years, with the great success of deep learning in natural language processing, significant progress has been made in the research of poetry generation. [24] were the first to try to use neural networks to generate classical Chinese poetry. They used a CNN and RNN to capture the sentence-level representation and generated historical sentences respectively. They then used an RNN to receive the output of both and to predict character by character in an autoregressive way. The experimental results showed that their method based on the neural network has obvious advantages over the traditional method. After that, in order to improve semantic consistency and coherence, researchers proposed a series of methods, such as topic planning [24], iterative generation [5, 26], conditional variational autoencoders [12, 29], memory networks [32, 33], and reinforcement learning [15, 31]. In addition, researchers have also carried out a series of explorations of the content diversity [3, 28, 30, 33].

All of the above works are promising, but they more or less ignore the prerequisite of poetry, that is, the metrical compliance. Metres is an essential part of poetry, which is the most significant difference from other literature. As far as we know, only [13, 34] take into account the metrical constraints, but they just used limited metrical information and restrict themselves to a specific poetry genre such as Song iambic. In

addition, in the poetry generation task of other languages, they usually force the model to conform to the constraint by modifying the probability distribution of the decoder [4]. We think this method will damage the fluency of semantics and is not suitable for Chinese classical poetry. In this paper, we study the problem of the metrical constraints of Chinese classical poetry and puts forward an effective and general solution to solve it. We believe that our work can be complementary to other poetry generation systems and useful for various constrained text generation tasks, including music composition, template-based abstractive summarization, text simplification, and data-to-text generation.

3 Methodology

3.1 Problem Formulation

We begin with the notation definition. We use $P = \{co, pe, ca, *\}$ to denote three common punctuation¹ in Chinese poetry and the placeholder, $T = \{pt, zt\}$ to denote level tone (平) and downward tone (仄), $R = \{1, 2, \dots, 16\}$ to denote 16 kinds of rhyme classes² (韵部), and V to denote the vocabulary.

The input of the task includes n topic keywords $K = \{K_1, K_2, \dots, K_n\}$ which express the writing intention, and d kinds of metrical constraints $S = \{S_1, S_2, \dots, S_d\}$, where all S_i are sequences of the same length. Let $s_{i,j}$ be the rule in S_i that character y_j must follow, where $Y = \{y_j\}_{j=1}^{|Y|}$ is the output sequence. In particular, we set d to 3 to define the main three constraints of poetry: tone, rhyme and content. The more fine-grained constraints such as pairing (对仗) are ignored in this paper. We use pre-defined symbols above to represent punctuation, rhyme and tone, and use placeholder ‘*’ to represent characters without relevant information about rule S_i . More specifically, we use $S_1 = \{s_{1,j}\}_{j=1}^{|Y|}$, $S_2 = \{s_{2,j}\}_{j=1}^{|Y|}$ and $S_3 = \{s_{3,j}\}_{j=1}^{|Y|}$ to denote the tone, rhyme and content pattern respectively, where $s_{1,j} \in P \cup T$, $s_{2,j} \in P \cup R$, $s_{3,j} \in P \cup V$. Our goal is to output a sequence Y that is semantically consistent with K and formally satisfies the constraints S . See Fig. 1 for an example poem with constraints.

Note that the structure pattern mentioned in the introduction can be represented in any S , so there is no need to define it. In addition, the content pattern S_3 , which defines the characters that should appear in Y , doesn’t belong to metrical patterns. The content patterns are designed to make the model more flexible by providing a means of character-level constraints that allow users to intervene the content of the target sequence directly. For example, users want to have some characters appear in the target sequence (such as acrostic poem (藏头诗) or other interesting types) or refine the poem draft, which can be achieved through it easily.

3.2 The Dual-Encoder Model

Following the practice of [32], we use TextRank and extract 1 to 4 keywords from each poem to construct a keyword sequence where a separator separates keywords. Besides,

¹ ‘co’, ‘pe’ and ‘ca’ refer to the punctuation ‘，。、’ respectively.

² Tone and Rhyme are defined by TongYun (a pioneering book on Chinese rhythm). Learn more at <https://sou-yun.cn/mqr.aspx>.

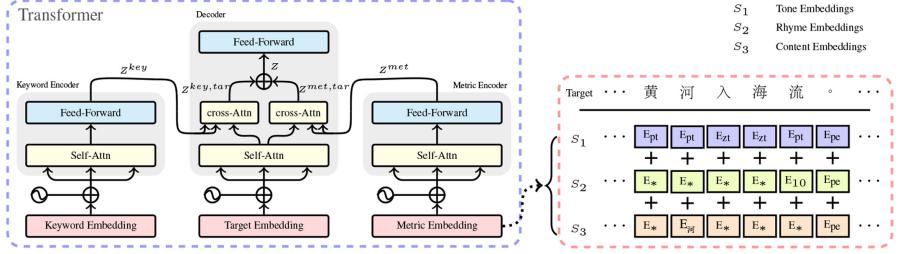


Fig. 2. The architecture of our model.

we noticed some work [12, 14] took the title of the poem as the writing intention input. We also implement this method in our model. We extract the rhyme and tone pattern from each poem and represent them as S_1 and S_2 respectively. For S_3 , we randomly select a few characters from the target sequence, at test time, all the positions except punctuations are replaced by placeholder ‘*’.

The backbone of our approach is Transformer-based [20] encoder-decoder framework. We use two encoders, including a keyword encoder and a metric encoder, to encode the writing intention and metrical information. The keyword encoder acts as the standard encoder in NMT, and can be treated as the generation’s source. The metric encoder provides constraints information to guide decoding. Then the decoder generates the poem by using the output of the two encoders simultaneously. See Fig. 2 for the workflow of our method.

We use h_l^{met} and h_l^{key} to denote the hidden states of metric encoder and keyword encoder respectively, where l is the layer index. For h_0^{met} , we sum all the embeddings of the input symbols as the representation:

$$h_{0,t}^{met} = \mathbf{E}_t^{S_1} + \mathbf{E}_t^{S_2} + \mathbf{E}_t^{S_3} + \mathbf{E}_t^{pos} \quad (1)$$

where t is the position index, $h_{0,t}^{met}$ is the representation of the t -th element in the embedding layer of the metric encoder. $\mathbf{E}_t^{S_1}$, $\mathbf{E}_t^{S_2}$ and $\mathbf{E}_t^{S_3}$ are the embedding of the tone token, rhyme token and content token respectively. \mathbf{E}_t^{pos} is the position embedding. For h_0^{key} , we have:

$$h_{0,t}^{key} = \mathbf{E}_t^K + \mathbf{E}_t^{pos} \quad (2)$$

where \mathbf{E}_t^K is the keyword token embedding, $h_{0,t}^{key}$ is the representation of the t -th element in the embedding layer of the keyword encoder. Once we have the representations Z^{key} and Z^{met} from the keyword encoder and the metric encoder, the decoder computes two hidden states $Z^{key,dec}$ and $Z^{met,dec}$ with respective multi-head cross-attention parameters, then interpolate it in a reasonable way:

$$Z = \alpha Z^{key,dec} + (1 - \alpha) Z^{met,dec} \quad (3)$$

where α is the interpolation coefficient. Then the decoder makes use of the integrated context to generate the poem character by character.

Table 1. Details of the datasets. In the test set of quatrains and regulated verse, 5-characters (五言) and 7-characters (七言) account for 500 poems respectively. Token is the number of characters.

Corpus	Train	Valid	Test	5-characters	7-characters	Token
Quatrains	121436	1000	1000	21368	100,068	3753568
Regulated verse	179487	1000	1000	78,900	100587	10286864
Song iambic	11190	1000	1000	–	–	946183
All	312113	3000	3000	–	–	14986615

4 Experiments

4.1 Setup

We conducted experiments on three popular genres of Chinese classical poetry: quatrains, regulated verse, and Song iambic. We train one model jointly for all genres and beam search is used for decoding in the inference phase. We segment words in character units [25]. The details of our datasets are shown in Table 1. For the model configuration, we set the number of encoder and decoder layers to 4, and the rest was the same as the Transformer-Base configuration [20]. We train our model using the Adam [9] optimization method with: $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 1e - 8$. Additionally, we add dropout [18] with drop ratio $r = 0.2$ to prevent overfitting. We set the interpolation coefficient α to 0.5.

4.2 Comparison Methods

We consider comparing the following models:

MCPG. Our proposed model. We noticed some work [12, 14] took the title of the poem as the writing intention input. We also implement this method (namely MCPG-title) in our model. The model using keywords is called MCPG-key.

LMPG. [14] proposes a simple and effective method to generate classical Chinese poetry with a language model. Their model did not use any metrical information and distinguished different genres by using a genre label.

S2S. A sequence-to-sequence framework [1] based on Transformer. We put the keywords and genre information (such as a genre token) together as the input sequence and the target as the output sequence.

SongNet. [13] used pre-defined templates that contained partial metrical information (length and rhyme) to generate poetry. They experimented on Song iambic and sonnets, and we extended its method to quatrains and regulated verse.³

These methods are based on neural network and have achieved satisfactory performance compared with the traditional methods. Among them, LMPG and S2S do not use any metrical information, and SongNet uses templates containing length and rhyme information to generate poetry.

³ Code:<http://github.com/lipiji/SongNet>.

4.3 Evaluation Metrics

Automatic Evaluation. We used BERT-CCPoem⁴ instead of BLEU to measure the similarity between the generated poems and the ground-truth because BLEU was not designed to capture deep semantic information. Besides, we used three criteria to quantify how well the generated poems meet the given metrical requirement [13, 34]. The details are as follows:

- Format accuracy: The format accuracy is the average percentage of the lines with the correct length in a poem and then further averaged over all samples.
- Tone accuracy: The tone accuracy is the percentage that the tone is predicted correctly over all samples.
- Rhyme accuracy: The rhyme accuracy is the average percentage of rhyme correctness over all rhymes groups⁵ in a poem and then further averaged over all samples.

Human Evaluation. We recruited ten well-educated native annotators to evaluate the generated poems in a blind review manner by four criteria, namely, fluency, meaning, coherence and poeticness [5, 24, 26, 31]. Because the user’s writing intention (keywords) will eventually appear in the generated poem, it is not necessary to evaluate the intention relevance. The details are as follows:

- Fluency: Is the poem grammatically and syntactically formed?
- Meaning: Does the poem convey meaningful information?
- Coherence: Is the poem thematically coherent across lines?
- Poeticness: Does the poem have some poetic and artistic beauties?

Each criterion was rated from 1 to 3, representing bad, normal, good, respectively. For each genre, we randomly selected 20 topic words to generate poems with these models respectively.

5 Results and Discussions

5.1 Automatic Evaluation

As observed in the automatic evaluation of Table 2, our methods (MCPG-key/title) demonstrate outstanding format, rhyme and tone accuracies on three genres. The poems generated by our model almost in perfect compliance with the metrical constraints, indicating the effect of the metrical information. For quatrains and regulated verse, due to their regular structure (four lines or eight lines, each line contains five characters or seven characters), each model can achieve nearly 100% performance on Format accuracy. However, for Song iambic with complex structure pattern and little training data, the model does not perform well without the metrical constraints, as shown by LMPG and S2S. Besides, MCPG has additional metrical information from the original poem,

⁴ A pre-trained model for Chinese classical poetry, developed by Research Center for Natural Language Processing, Computational Humanities and Social Sciences, Tsinghua University. URL: <https://github.com/THUNLP-AIPoet/BERT-CCPoem>.

⁵ A poem may have multiple groups of rhymes, especially Song iambic.

Table 2. The automatic and human evaluation results. Flu., Mea., Coh., and Poe. represent the Fluency, Meaning, Coherence and Poeticness respectively.

Genre	Model	Automatic evaluation ↑				Human evaluation ↑			
		Sim.	Format	Rhyme	Tone	Flu.	Mea.	Coh.	Pot.
Quatrains	LMPG	74.92	99.68	86.47	66.52	2.47	2.33	2.37	2.41
	S2S	81.65	99.95	85.98	62.45	2.36	2.28	2.29	2.28
	SongNet	73.02	99.96	87.78	67.39	2.44	2.26	2.34	2.30
	MCPG-title	75.53	100.00	99.41	99.98	2.59	2.32	2.17	2.39
	MCPG-key	82.41	100.00	98.13	99.57	2.46	2.35	2.56	2.32
Regulated verse	LMPG	81.58	99.9	84.69	69.27	2.34	2.33	2.32	2.26
	S2S	80.75	99.92	82.41	64.27	2.29	2.31	2.23	2.20
	SongNet	78.08	99.89	85.70	68.84	2.31	2.30	2.32	2.21
	MCPG-title	79.93	100.00	98.80	99.94	2.38	2.36	2.25	2.28
	MCPG-key	83.32	100.00	97.54	99.70	2.55	2.45	2.51	2.35
Song iambic	LMPG	81.23	86.12	64.56	65.36	2.03	2.08	2.12	2.06
	S2S	81.42	77.83	60.12	62.78	2.01	2.04	2.06	2.01
	SongNet	81.08	99.70	65.33	65.33	2.06	2.11	2.21	2.14
	MCPG-title	81.45	99.64	97.85	99.92	2.11	2.24	2.25	2.16
	MCPG-key	83.84	99.96	97.19	99.80	2.35	2.18	2.29	2.17

which leads to higher performance of Similarity than other models. However, it is worth noting that for the artistic creation task, the similarity between the generated results and the references can not confidently explain the quality of the work, which is also why human evaluation is needed.

5.2 Human Evaluation

As we all know, the restriction of characters and the fluency of semantics are always contradictory to each other to some extent. The human evaluation results show that our model obtains a comparable score with other models on four criteria, which indicates that our approach can achieve a good balance between semantics and constraints. We think it benefits from softly introducing constraints rather than forcing the model to satisfy constraints, such as modifying the decoding process. Figure 3 is a visualization of the cross-attention with the metric encoder of the example generated in Case Study (See Fig. 4), where the y-axis is the input of the metric encoder and the x-axis is the generated result. As shown in the figure, the diagonal has a significant weight, which indicates that the decoder can automatically learn to use metrical information to guide and constrain the generation process.

Another observation is that using keywords performs better semantically than using titles. This may be because keywords can provide more direct and abundant writing materials for the model. The titles vary in quality, with some titles having little meaning and a weak relationship to the content, making the model more difficult to learn.

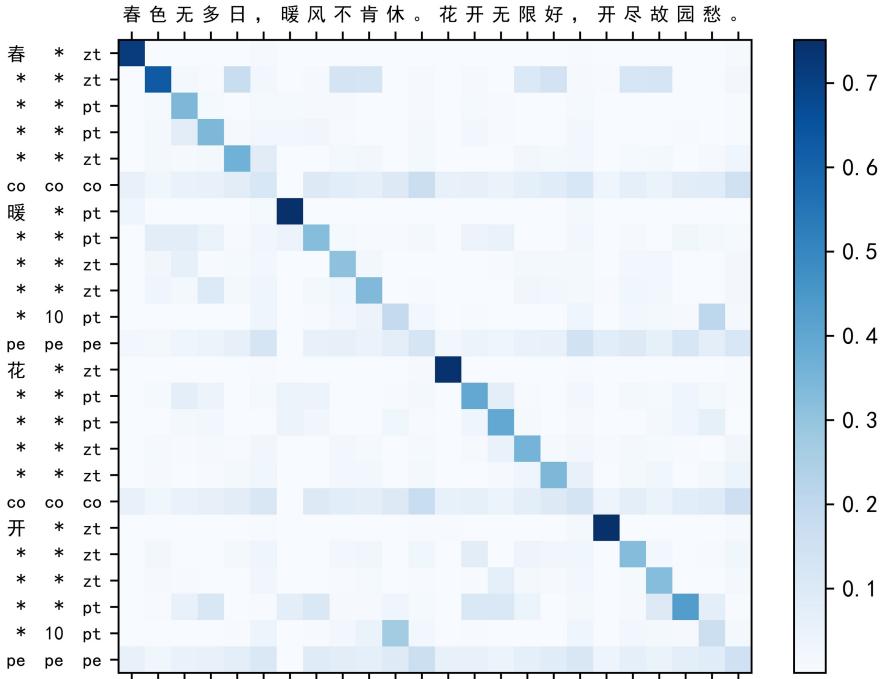


Fig. 3. The visualization of the cross-attention with the metric encoder of the example generated in Case Study.

5.3 Case Study

In most previous methods, users can only influence the generated result macroscopically through their writing intention. The choice of wording, phrasing and metrical patterns depends entirely on the model. This may make the model tend to use a common pattern to generate the poem. That means the model will pay much more attention to high-frequency patterns whereas ignoring low-frequency ones, which limits its practicability. Actually this problem exists not only in poem generation but also widely in various text generation tasks [11, 19, 33].

In our model, users have a strong ability to intervene in the model by modifying the metrical patterns to influence the generated poem. We can assign any tone and rhyme pattern and the character expected to appear in the target sequence. This metrical information provides a regularization function, which adjusts the model's behaviour to generate the poem conformed to a specific metrical pattern we want. This mode is flexible, controllable, and very helpful for poetry education, literary studies and auxiliary creation. For example, we use “故园” as the keyword, use the tone pattern and rhyme pattern in Fig. 1, and use “春暖花开” to construct the content pattern to generate an acrostic poem (藏头诗). As shown in Fig. 4, the generated poem has coherent sentiment and smooth grammar and agrees with the metrical patterns broadly. Similarly, we can

Keywords	故园
Tone Pattern	zt zt pt pt zt co pt pt zt zt pt pe zt pt pt zt zt co zt zt zt pt pt pe
Rhyme Pattern	* * * * * co * * * * * 10 pe * * * * * co * * * * * 10 pe
Content Pattern	春 * * * * co 暖 * * * * pe 花 * * * * co 开 * * * * pe
Result	春色无多日，暖风不肯休。花开无限好，开尽故园愁。

Fig. 4. A generated acrostic poem with “春暖花开”. The rhyming characters are red. (Color figure online)

Tone Pattern	pt zt pt pt zt co pt pt zt zt pt pe zt pt pt zt zt co pt zt zt pt pt pe
Rhyme Pattern	* * * * * co * * * * * 4 pe * * * * * co * * * * * 4 pe
Result	独立秋风外，孤舟落日西。不知何处去，犹有故人啼。
Tone Pattern	zt zt pt pt zt co pt pt zt zt pt pe pt pt pt zt zt co zt zt zt pt pt pe
Rhyme Pattern	* * * * * co * * * * * 12 pe * * * * * co * * * * * 12 pe
Result	落日孤城外，秋风万里心。谁知今夜月，照见故人吟。

Fig. 5. Examples of using different metrical patterns to generate the same keywords “落日”. The rhyming characters are red. (Color figure online)

compose poetry with arbitrary form by manipulating the metrical patterns, even new patterns that we create and never appeared in training data.

Another interesting finding is that in our model, given the same keywords, using different metrical patterns can make the model generate vastly different poems, which increases the diversity of poetry to some extent. Figure 5 shows an example. Both poems achieve good performance in terms of metrical compliance and semantics. This means that in addition to using common decoding strategies such as Top-K, we can adjust metrical patterns to increase the diversity, and the latter is far more effective than the former.

5.4 Limitations

Although MCPG can produce semantically fluent and emotionally rich poems, these poems are deficient in the story occasionally. In other words, logical rationality and relevance are insufficient in the emotional transition process. Another phenomenon is that MCPG seems to be inclined to generate sad emotional poetry. We think this may be due to the bias in data.

There are many elements in poetry creation, such as voice, diction, imagery, figures of speech, symbolism, allegory, syntax, sound, and metres. In this paper, we mainly focus on solving the problem of metres. We believe that the introduction of more elements in the model will solve the above problems. This is also our future work.

6 Conclusions

In this paper, we have proposed MCPG, a simple and effective model for poetry automatic generation. We define the incorporation of metrical constraints into the poetry

generation system as an encoding problem. A Transformer-based dual-encoder model is then proposed to force the poetry generation conditioned on both the writing intention and the metrical patterns. We empirically demonstrate that our model can generate excellent poems satisfied with the designated metrical patterns while maintaining coherence and fluency.

We believe that our work can be complementary to other poetry generation systems and useful for various constrained text generation tasks, including music composition, template-based abstractive summarization, text simplification, and data-to-text generation. In future work, we will explore the introduction of other elements in our model to improve the quality of poetry further and verify the effectiveness of our proposed framework in different tasks.

References

- Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015, Conference Track Proceedings (2015)
- Bugliarello, E., Okazaki, N.: Enhancing machine translation with dependency-aware self-attention, pp. 1618–1627. Association for Computational Linguistics (2020)
- Chen, H., Yi, X., Sun, M., Li, W., Yang, C., Guo, Z.: Sentiment-controllable Chinese poetry generation, pp. 4925–4931. ijcai.org (2019)
- de Cruys, T.V.: Automatic poetry generation from prosaic text, pp. 2471–2480. Association for Computational Linguistics (2020)
- Deng, L., et al.: An iterative polishing framework based on quality aware masked language model for Chinese poetry generation, pp. 7643–7650. AAAI Press (2020)
- Gervás, P.: An expert system for the composition of formal Spanish poetry. Knowl. Based Syst. **14**(3–4), 181–188 (2001)
- He, J., Zhou, M., Jiang, L.: Generating Chinese classical poems with statistical machine translation models. In: AAAI (2012)
- Jiang, L., Zhou, M.: Generating Chinese couplets using a statistical MT approach. In: COLING, pp. 377–384 (2008)
- Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015, Conference Track Proceedings (2015)
- Kumar, A., Ahuja, K., Vadapalli, R., Talukdar, P.: Syntax-guided controlled generation of paraphrases. Trans. Assoc. Comput. Linguist. **8**, 330–345 (2020)
- Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B.: A diversity-promoting objective function for neural conversation models, pp. 110–119. The Association for Computational Linguistics (2016)
- Li, J., et al.: Generating classical Chinese poems via conditional variational autoencoder and adversarial training, pp. 3890–3900. Association for Computational Linguistics (2018)
- Li, P., Zhang, H., Liu, X., Shi, S.: Rigid formats controlled text generation, pp. 742–751. Association for Computational Linguistics (2020)
- Liao, Y., Wang, Y., Liu, Q., Jiang, X.: GPT-based generation for classical Chinese poetry. CoRR [arXiv:1907.00151](https://arxiv.org/abs/1907.00151) (2019)
- Liu, D., Lv, J., Li, Y.: Generating style-specific Chinese tang poetry with a simple actor-critic model. IEEE Trans. Emerging Top. Comput. Intell. **3**(4), 313–321 (2018)
- Manurung, H.: An evolutionary algorithm approach to poetry generation (2004)

17. Peng, H., Parikh, A.P., Faruqui, M., Dhingra, B., Das, D.: Text generation with exemplar-based adaptive decoding, pp. 2555–2565. Association for Computational Linguistics (2019)
18. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014)
19. Su, H., et al.: Diversifying dialogue generation with non-conversational text, pp. 7087–7097. Association for Computational Linguistics (2020)
20. Vaswani, A., et al.: Attention is all you need, pp. 5998–6008 (2017)
21. Wang, Q., et al.: Learning deep transformer models for machine translation. In: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, 28 July–2 August 2019, Volume 1: Long Papers, pp. 1810–1822 (2019)
22. Wang, Q., Luo, T., Wang, D., Xing, C.: Chinese song iambics generation with neural attention-based model, pp. 2943–2949. IJCAI/AAAI Press (2016)
23. Wang, X., Pham, H., Yin, P., Neubig, G.: A tree-based decoder for neural machine translation, pp. 4772–4777. Association for Computational Linguistics (2018)
24. Wang, Z., et al.: Chinese poetry generation with planning based neural network. In: COLING, pp. 1051–1060 (2016)
25. Xiao, T., Zhu, J., Zhang, H., Li, Q.: NiuTrans: an open source toolkit for phrase-based and syntax-based machine translation. In: Proceedings of the ACL 2012 System Demonstrations, pp. 19–24 (2012)
26. Yan, R.: I, poet: automatic poetry composition through recurrent neural networks with iterative polishing schema. In: IJCAI, pp. 2238–2244 (2016)
27. Yan, R., Jiang, H., Lapata, M., Lin, S., Lv, X., Li, X.: I, poet: automatic Chinese poetry composition through a generative summarization framework under constrained optimization. In: IJCAI, pp. 2197–2203 (2013)
28. Yang, C., Sun, M., Yi, X., Li, W.: Stylistic Chinese poetry generation via unsupervised style disentanglement, pp. 3960–3969. Association for Computational Linguistics (2018)
29. Yang, X., Lin, X., Suo, S., Li, M.: Generating thematic Chinese poetry using conditional variational autoencoders with hybrid decoders, pp. 4539–4545. ijcai.org (2018)
30. Yi, X., Li, R., Yang, C., Li, W., Sun, M.: MixPoet: diverse poetry generation via learning controllable mixed latent space, pp. 9450–9457. AAAI Press (2020)
31. Yi, X., Sun, M., Li, R., Li, W.: Automatic poetry generation with mutual reinforcement learning, pp. 3143–3153. Association for Computational Linguistics (2018)
32. Yi, X., Sun, M., Li, R., Yang, Z.: Chinese poetry generation with a working memory model, pp. 4553–4559. ijcai.org (2018)
33. Zhang, J., et al.: Flexible and creative Chinese poetry generation using neural memory, pp. 1364–1373. Association for Computational Linguistics (2017)
34. Zhang, R., Liu, X., Chen, X., Hu, Z., Xu, Z., Mao, Y.: Generating Chinese CI with designated metrical structure, pp. 7459–7467. AAAI Press (2019)
35. Zhang, X., Lapata, M.: Chinese poetry generation with recurrent neural networks, pp. 670–680. ACL (2014)
36. Zhou, C.L., You, W., Ding, X.: Genetic algorithm and its implementation of automatic generation of Chinese SONGCI. *J. Softw.* **21**(3), 427–437 (2010)



CNewSum: A Large-Scale Summarization Dataset with Human-Annotated Adequacy and Deducibility Level

Danqing Wang¹, Jiaze Chen¹, Xianze Wu¹, Hao Zhou¹, and Lei Li^{2(✉)}

¹ ByteDance AI Lab, Shanghai, China

wangdanqing.122, chenjiaze, zhouchao.nlp}@bytedance.com

² Computer Science Department, University of California, Santa Barbara, Santa Barbara, USA
lilei@cs.ucsb.edu

Abstract. Automatic text summarization aims to produce a brief but crucial summary for the input documents. Both extractive and abstractive methods have witnessed great success in English datasets in recent years. However, there has been a minimal exploration of text summarization in other languages, limited by the lack of large-scale datasets. In this paper, we present a large-scale Chinese news summarization dataset CNewSum, which consists of 304,307 documents and human-written summaries for the news feed. It has long documents with high-abstractive summaries, which encourages document-level understanding and generation for current summarization models. An additional distinguishing feature of CNewSum is that its test set includes adequacy and deducibility annotations for the summaries. The adequacy level measures the degree of summary information covered by the document, and the deducibility indicates the reasoning ability the model needs to generate the summary. These annotations help researchers target their model performance bottleneck. We examine recent methods on CNewSum and will release our dataset after the anonymous period to provide a solid testbed for automatic Chinese summarization research.

Keywords: Automatic text summarization · Chinese summarization dataset · Adequacy and Deducibility

1 Introduction

Text summarization is an important task in natural language processing, which requires the system to understand the long document and generate a short text to summarize its main idea. There are two primary methods to generate summaries: *extractive* and *abstractive*. Extractive methods select semantic units from the source document and reorganize them into a consistent summary, while abstractive models generate summaries using words and phrases freely. Benefiting from pre-trained language models [2, 10, 14], much process has been made on English summarization datasets, such as Newsroom [5], CNN/DailyMail [6], and NYT [19].

However, the lack of the high-quality datasets in other languages, such as Chinese, limits further researches on summarization under different language habits and cultural customs. It hinders the application of current summarization models to more languages. Currently, most Chinese summarization datasets are collected from Chinese

Table 1. An example of our CNewSum dataset.

Article [0]图为在广元市朝天区发现的白耳夜鹭。[1]广林局供。[2]中新网广元3月15日电。[3]记者15日从四川省广元市野生动物救治中心获悉：近日，该市朝天区东溪河乡的群众发现一只受伤的“怪鸟”引起多方关注，随后，上报到广元市林业部门。[4]后经当地野生动物保护专家鉴定“怪鸟”为世界最濒危鸟类白耳夜鹭。……[7]该鸟是我国特有的珍稀鸟类、国家二级保护动物白耳夜鹭，被列为世界最濒危的30种鸟类之一，目前全世界仅存1000余只……[14]此后一直再没有关于该鸟踪迹的报道。

[0]*The picture shows the Gorsachius magnificus in Chaotian District of Guangyuan City.* [1]*Supplied by Guangyuan Forestry Department.* [2]*Xinhua News Agency, Guangyuan, March 15.* [3]*Reporters learned from the Wildlife Treatment Center of Guangyuan City, Sichuan Province on the 15th that recently, the discovery of an injured "strange bird" by the local people in Dongxihe village, Chaotian District of the city attracted much attention and was subsequently reported to the forestry department of Guangyuan City.* [4]*After that, local wildlife protection experts identified the "strange bird" as the world's most endangered bird, the Gorsachius magnificus.....*[7]*It is a rare bird unique to our country and a national second-class protected animal, the Gorsachius magnificus. It has been listed as one of the world's most endangered 30 species of birds. At present, there are only about 1,000 birds in the world.....*[14]*There have been no reports of the bird's trace since then.*

Summary 今日获悉，广元一市民发现受“怪鸟”，经鉴定系世界濒危鸟类白耳夜鹭，全球仅存1000只。

(*It was reported today that a citizen of Guangyuan found an injured "strange bird", which was identified as a world-endangered bird, the white-eared night heron, of which only 1,000 exist worldwide.*)

Sentence-Label:[0,4] **Adequacy:** 1 **Deducibility:** 1

social media Weibo, subject to a 140-word length limit [4,7]. There are also some datasets scraped from news websites, such as Toutiao [8] and ThePaper [12]. However, those datasets are either small-scale or not of high quality.

In this paper, we present a large-scale Chinese news summarization dataset, CNewSum, to make up for the lack of Chinese document-level summarization, which can become an important supplement to current Chinese understanding and generation tasks. Different from previous summarization datasets crawled from news websites, we called for news articles from over hundreds of thousands press publishers and hired a team of expert editors to provide human-written summaries for the daily news feed. During the summarization process, the editors may perform simple reasoning or add external knowledge to make the summary more reader-friendly. Thus, we further investigate our test set and explore how much knowledge the models need to generate a human-like summary. Specifically, we ask annotators to determine two questions: 1) **Adequacy:** *Is the information of summaries self-contained in the source document?* 2) **Deducibility:** *Can the information be deduced from the source document directly, or needs external knowledge?* We provide these two scores for each example in the test set. Table 1 is an example of our dataset.

Our main contribution are as follows:

- (1) We propose a large-scale Chinese news summarization dataset collected from over hundreds of thousands news publishers. We hire a team of expert editors to write summaries for news feed.

- (2) In order to figure out how much knowledge the model need to generate a human-like summary, we manually annotate the adequacy and deducibility level for our test set.
- (3) We also provide several strong extractive and abstractive baselines, which makes the dataset easy to use as the benchmark for Chinese summarization tasks.

2 Related Work

News Summarization Dataset. Most news summarization datasets focus on English language, and here we give a brief introduction to some popular ones and list the detailed information in the first part of Table 2. NYT is a news summarization dataset constructed from New York Times Annotated Corpus [19]. We tokenize and convert all text to lower-case, follow the split of Paulus et al. [18]. The CNN/DailyMail question answering dataset [6] modified by Nallapati et al. [16] and See et al. [20] is the most commonly-used dataset for single-document summarization. It consists of online news articles with several highlights. Those highlights are concatenated as the summary. Newsroom [5] is a large-scale news dataset scraped from 38 major news publications, ranging from business to sports. These summaries are often provided by editors and journalists for social distribution and search results.

Chinese Summarization Dataset. There are also several Chinese summarization datasets in other domains [3, 9, 22], but here we only discuss news summarization datasets. The detailed statistics are listed in the second part of Table 2. The LCSTS [7] is a large-scale Chinese social media summarization dataset. It is split into three parts, and the part II and part III are usually used as development and test set after filtering out low-quality examples. RASG [4] collects the document-summary-comments pair data for their reader-aware abstractive summary generation task. It utilizes users' comments to benefit the generation of the abstractive summary of main content. The document is relatively short and has about 9 comments as a complement. TTNews [8] is provided for NLPCC Single Document Summarization competition,¹ including 50,000 training examples with summaries and 50,000 without summaries. CLTS [12] is a Chinese summarization dataset extracted from the news website ThePaper. It contains more than 180,000 long articles and corresponding summaries written by professional editors and authors.

3 The CNewSum Dataset

3.1 Data Collection

We receive news submissions from over hundreds of thousands press publishers.² We hire a team of expert editors to provide human-written summaries for the daily news

¹ <http://tcci.ccf.org.cn/conference/2018/taskdata.php>.

² The press publishers include thepaper.cn, wallstreetcn.com, cankaoxiaoxi.com, yicai.com, and so on. They submit their articles in web format to our company. These publishers retain any copyright they may have in their content and grant us a royalty-free, perpetual licence to use, copy, edit and publish their content.

feed. Each example will be double-checked by different experts to ensure its quality. We construct CNewSum by extracting news article from 2015 to 2020³ and filtering summaries with less than 5 words. We further limit the length of documents to 50–5000. To solve the problem of missing and inaccurate punctuation in web format, we train an extra punctuation tagging model via Bi-LSTM on Chinese articles to correct punctuation.⁴

Finally we obtain a Chinese news corpus with 304,307 document-summary pairs. It is split into training/validation/test by 0.9/0.05/0.05. Besides, we compare document sentences with human-written summaries and use the greedy algorithm following [16] to get the ORACLE sentences with label 1 as the signal for extractive summarization.

Table 2. The summarization datasets. The top part contains the commonly-used English news summarization and the bottom contains the Chinese summarization datasets. ‘-’ means the original dataset does not provide the standard split for train/dev/test set. For TTNews, we only take training examples with summaries into consideration. ‘*’ includes 2,000 evaluation examples for NLPCC2017 and 2,000 for NLPCC2018.

Dataset	Train	Dev	Test	Total	Article	Summary	Source
NYT [19]	589,282	32,737	32,739	654,758	552.14	42.77	New York Times
CNNDM [6]	287,227	13,368	11,490	312,085	791.67	55.17	CNN & Daily Mail
Newsroom [5]	995,041	108,837	108,862	1,212,740	765.59	30.22	38 news sites
LCSTS [7]	2,400,591	8,685	725	2,410,001	103.7	17.90	Weibo
RASG [4]	863,826	-	-	863,826	67.08	16.61	Weibo
TTNews [8]	50,000	-	4,000*	54,000	747.20	36.92	Toutiao
CLTS [12]	148,317	20,393	16,687	185,397	1363.69	58.12	ThePaper
CNewSum	275,596	14,356	14,355	304,307	790.55	37.58	News publishers

3.2 Adequacy and Deducibility Annotation

Analyzing our dataset, we find that the expert editors often perform some reasoning or add external knowledge to make the summary more friendly for the readers. For example, the precise figure (2,250) may be summarized as an approximate number (more than two thousand). In another case, a specific date will be converted to a relative time based on the time of publication, e.g. tomorrow. This information is not directly available in the original document. Thus, we wonder how much knowledge the model needs to generate the human-written summary. Inspired by [1], we ask annotators to answer the following two questions for each document-summary pair in our test set:

- 1) **Adequacy.** *Does necessary information of the summary has been included in the document?* For example, all words in the summary can be directly found in the document, or they have synonyms or detailed descriptions in the original text. Under these circumstances, the summary is labeled as 1.

³ These data have been checked for legality and can be released for research use.

⁴ The accuracy rate is 96.20%.

- 2) **Deducibility.** *Can the information of the summary be easily inferred from the document?* Unit conversion, number calculation, and name abbreviations that can be inferred are labeled as 1. In contrast, complex conclusions with no direct mentions in the original document are labeled as 0.

For each question, the annotators should choose 0 or 1. We hired a team of 12 employees to annotate the test set.⁵ We first trained these employees on basic annotation rules, and they were required to annotate 100 examples and then be checked and corrected by us. Two voluntary expert annotators were employed to control quality. They were asked to sample 10% from each annotator and recheck the annotation. If one's consistent rate is less than 95%, all annotations of this annotator will be returned and re-annotated. It is consistent only if the two experts and the annotator agree on their answers, otherwise the example will be further discussed.

Table 3. The statistics of news summarization datasets. Cov., Den. and Comp. correspond to the *Coverage*, *Density* and *Compression* introduced by [5]. The Bi., Tri. and 4-gram are the n-gram novelty (%). The novelties of NYT/CNNDM/Newsroom are from [17]. For Chinese data, it is calculated by words.

Dataset	Cov.↓	Den.↓	Comp.↑	Bi.↑	Tri.↑	4-gram↑
NYT	0.83	3.50	24.19	55.59	71.93	80.16
CNNDM	0.85	3.70	13.76	49.70	70.20	79.99
Newsroom	0.82	9.50	36.03	46.80	58.06	62.72
LCSTS	0.54	1.23	6.61	80.29	90.92	94.53
RASG	0.61	2.52	7.27	67.89	76.94	80.15
TTNews	0.76	3.21	22.24	61.09	76.30	83.64
CLTS	0.99	28.73	24.81	5.14	8.08	10.36
CNewSum	0.76	2.77	20.83	63.29	78.54	85.64

3.3 Dataset Analysis

As shown in Table 2, our CNewSum dataset has a similar scale with the most popular English summarization dataset CNNDM, which is suitable for training and evaluating different summarization models. For the Chinese dataset, the average length of the document and the summary are significantly longer than datasets collected from Weibo and similar with TTNews.

Following Grusky et al. [5], we also use *Coverage*, *Density* and *Compression* to characterize our summarization dataset. *Coverage* measures the overlap degree of the extractive fragment between the article and summary, and *Density* measures the average length of the extractive fragment. *Compression* is the ratio of the article length to the summary length. In addition, we calculate the n-gram novelty of the summary, which

⁵ We paid 1 RMB (0.15 dollar) for each example, and the average hourly wage is 60 RMB (the minimum hourly wage is 24 RMB).

Table 4. The adequacy (A) and deducibility (D) level in our test set.

A = 1 & D = 1	A = 0 & D = 1	A = 0 & D = 0
91.08%	4.11%	4.81%

is the percentage of n-grams that do not appear in the document, as described in [17]. The results are shown in Table 3. We can find that the datasets collected from Weibo usually have lower coverage and density ratio, with high compression and novelty. This indicates that the summaries for these short documents are more abstractive. For news article summarization, CLTS almost copy most words of the summary from the document directly, which is indicated by the highest coverage, density and the lowest novelty. Our CNewSum provides a large-scale document-level summarization dataset with comparable abstractiveness with short social media datasets.

Since all adequacy summaries can be inferred from the document, the $A = 1 \& D = 0$ is meaningless. For the summarization models, the examples with $A = 1 \& D = 1$ is relatively easy to generate, and $A = 0 \& D = 1$ ask for some inference abilities. The $A = 0 \& D = 0$ cannot be solved with the original document and may need the help of external knowledge. From Table 4, we find that more than 80% examples are adequate and deducible, but 20% lack essential information. With $D = 1$, the information can be inferred from the document. For example, “2005–2015” will be summarized as “ten years” which requires the model to do simple calculation. The rest summaries are factual but need external knowledge. News articles from the websites are time-sensitive and are filled with pictures. The editors often write the summary based on the time of the event and the image, which will cause the relative time, such as ‘yesterday’, and the picture description to appear in the summary. In addition, famous people will be mapped to their position in the summary, such as Obama and the American president of that time. It is difficult for the model to deduce such information from the news text without additional information. We keep these in our dataset to simulate real-world data distribution and let researchers evaluate the model performance from different aspects.

4 Experiment

We train several summarization models on our CNewSum. These systems include both abstractive and extractive methods, and the performance can serve as the baseline for future work.

4.1 Models

Baseline. We calculate three popular summarization baseline for our dataset. LEAD is a common lower bound for news summarization dataset [5, 16, 20]. For ORACLE, we concatenate the sentences with label 1 in the original order. TextRank [15] is simple unsupervised graph-based extractive methods.

Table 5. Results on the test set of CNewSum. The first part contains the Lead and Oracle baseline. The second and third part are extractive and abstractive summarization models.

Models	ROUGE-1	ROUGE-2	ROUGE-L
LEAD	30.43	17.26	25.33
ORACLE	46.84	30.54	40.08
TextRank [15]	24.04	13.70	20.08
NeuSum [24]	30.61	17.36	25.66
TFExt [13]	32.87	18.85	27.59
BERTExt	34.78	20.33	29.34
PG [20]	25.70	11.05	19.62
TFAbs [13]	37.36	18.62	30.62
BERTAbs	44.18	27.37	38.32

Neural Models. NeuSum [24] jointly score and select sentences for extractive summarization. PG [20] is the pointer-generator network which is a commonly-used encoder-decoder abstractive summarization model with the copy and coverage mechanism. Transformer [21] is a well-known sequence-to-sequence model based on the self-attention mechanism. Following the settings in [13], we employ two Transformer baselines: TFExt and TFAbs. The pre-trained language models such as BERT [2] have improved both abstractive and extractive summarization by a large margin, so we also apply the BERTSum mode [13] to our dataset. We train a Chinese BERT language model with Chinese news articles,⁶ which is noted as BERTExt and BERTAbs.

For extractive summarization, we choose the top-2 sentences as the summary due to the average sentence number (1.49) of the ground truth summary. The automatic metric ROUGE [11] is used for evaluation. Since the original ROUGE is made only for English, we follow the method of [7] and map the Chinese words to numbers. Specifically, the Chinese text is split by characters and the English words and numbers will be split by space. For example, “Surface Phone 将装载 Windows 10 (*The Surface Phone will be loaded with Windows 10*)” will be transformed to “surface/phone/将/装/载/windows/10” and then mapped to numeral IDs.

4.2 Results

As shown in Table 5, the abstractive models have better results on CNewSum test set, which is consistent with our analysis in Sect. 3.3. The abstractive methods has performed better than extractive models, which means that extractive methods have many performance limitations in CNewSum.

⁶ Since the bert-base-chinese model of Google does not perform well in our dataset.

Table 6. The results of models on different adequacy and deducibility level.

Model	Category	ROUGE-1	ROUGE-2	ROUGE-L
TFExt	A = 1 & D = 1	33.16	19.19	27.88
	A = 0 & D = 1	30.89	15.60	25.38
	A = 0 & D = 0	28.92	14.88	23.74
TFAbs	A = 1 & D = 1	37.54	18.85	30.83
	A = 0 & D = 1	36.36	16.70	29.63
	A = 0 & D = 0	34.73	15.95	27.52
BERTExt	A = 1 & D = 1	35.05	20.67	29.62
	A = 0 & D = 1	32.81	16.90	27.05
	A = 0 & D = 0	31.07	16.57	25.72
BERTAbs	A = 1 & D = 1	44.51	27.76	38.70
	A = 0 & D = 1	41.75	23.64	35.34
	A = 0 & D = 0	40.18	23.34	33.60

We further evaluate models based on adequacy and deducibility level. The results shown in Table 6 indicate that this model performs well on A = 1 where all necessary information can be easily found in the source document. However, when it asks for simple deducing or external knowledge, the performance degrade significantly.

4.3 Case Study

We illustrate the differences between abstractive models with a typical example in the appendix. As stated in previous work [20, 23], PG tends to copy directly from the original document instead of generating from vocabulary, which makes the output less abstractive. Besides, although it has used the coverage mechanism to avoid repetition, it still suffers the most from the meaningless duplication. For Transformer-based models, the random initialized model TFAbs introduces fake information, while the BERTAbs and TTBERTAbs perform much better in both capturing important information and generating fluent summaries.

Table 7. An example for abstractive summarization models. The text with underline is directly copied from the original article, and the bolded text contains fake information.

Article	<p>英雄联盟神秘预告再现。官方最新发布了一个短片视频，其短片的名称是“他已归来”。而最近更新的巨神峰新故事中就有描述星灵的，难道新英雄是星灵来自银河？今日，国外的LOL官方社交媒体上，放出了一一个预告短片，名称为“他已归来”。短片内容为，潘森正在凝视夜空中被星云所围绕的亮光。有人猜测，视频中的场景为潘森故事《巨神之枪》中的末尾内容，也是巨神峰新故事中所描述的《星灵》。歪果仁点评：Gigathor：天啊，下一个新英雄是银河系的！MrBananaHump：跟你们开玩笑呐，这只不过是巴德。SoSaysCory：应该是潘森的兄弟，潘林将会加入峡谷，技能与潘森一样，他们将会成为有史以来最强力的下路组合。Sharjo：将会有全新的巨神峰英雄了！潘森新的背景故事已提到了这个，在《巨神之枪》故事的结尾，指出了新的星灵到来。来自另一个次元的潘森老朋友将会和我们见面了！太酷了！DracCusS：感觉是：a)新英雄。b)潘森模型更新。c)宝石重做？</p> <p><i>League of Legends released a mysterious trailer and the official latest posted a short video. The name of the short film is "He Has Returned". In the recent new story of Mount Titan, there is a description of the Protoss. Will the new hero be the Protos from the Milky Way? Today, a short trailer was released on the official social media of LOL abroad, titled "He Has Returned." The content of the video is, Pan Sen stares at the bright light surrounded by nebula in the night sky. Some people guess, the scene in the video is the content of Pan Sen's story "Gun of the Titans", and the new story of Mount Titan is also described in the "Protoss". Foreigns Review: Giant: Oh my god, the next new hero is from the galaxy! BananaHump: I'm joking with you, this happens to be Bud. SoSaysCory: It should be Pan Sen's brother. Pan Lin will join the canyon. The skills are the same as Pan Sen. They will become the most concentrated bot combination ever. Sharjo: There will be a brand new Giant Mountain soon! Pan Sen's new background hero story has already mentioned it. At the end of the "Gun of the Gun" story, I saw the new Protoss. A friend of Pan Sen from another dimension will meet with us soon! so cool! DracCusS: It may be: a) New hero. b) Pan Sen model update. c) Gems reform?</i></p>
Gold	<p>英雄联盟官方发布星空主题预告短片，或将推出全新巨神峰英雄</p> <p><i>League of Legends officially released a starry sky theme trailer, or will launch a new Titan Heroes</i></p>
PG	<p>LOL官方发布了一个短片视频短片内容为，其短片内容为潘森正在凝视夜空中被星云所围绕亮光（组图）</p> <p><i>LOL officially released a short video , its short video content is that Pan Sen is gazing at the night sky surrounded by nebula light (Pictures)</i></p>
TFAbs	<p>英雄联盟官方发布新短片，称将于明年1月发行；英雄联盟新短片显示，新将于9月下旬发布。（组图）</p> <p><i>The League of Legends officially released a new short film, said it will be released in January next year. The new League of Legends video shows that new will be released in late September. (Pictures)</i></p>
BERTAbs	<p>英雄联盟公布新英雄预告：巨神峰新英雄是星灵来自银河？潘森新英雄将加入峡谷，宝石重做巨神之枪（组图）</p> <p><i>League of Legends announced the new hero trailer: Is the new hero of Titan Peak from the Milky Way? Pan Sen's new hero will join the canyon, and the gem will be remade the Titan's Spear (Pictures)</i></p>

5 Conclusion

We present CNewSum, a high-quality summarization dataset composed of human-written summaries to fill up for the lack of news summarization dataset in Chinese. We annotate all test set with adequacy and deducibility levels to help abstractive models solve the unfaithful problem. Finally, we give several popular extractive and abstractive baselines on the dataset for future research.

References

- Chen, D., Bolton, J., Manning, C.D.: A thorough examination of the CNN/Daily Mail reading comprehension task. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, pp. 2358–2367. Association for Computational Linguistics, August 2016. <https://doi.org/10.18653/v1/P16-1223>, <https://www.aclweb.org/anthology/P16-1223>
- Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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, pp. 4171–4186. Association for Computational Linguistics, June 2019. <https://doi.org/10.18653/v1/N19-1423>, <https://www.aclweb.org/anthology/N19-1423>
- Gao, S., Chen, X., Li, P., Chan, Z., Zhao, D., Yan, R.: How to write summaries with patterns? Learning towards abstractive summarization through prototype editing. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, pp. 3741–3751. Association for Computational Linguistics, November 2019. <https://doi.org/10.18653/v1/D19-1388>, <https://www.aclweb.org/anthology/D19-1388>
- Gao, S., et al.: Abstractive text summarization by incorporating reader comments. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, 27 January–1 February 2019, pp. 6399–6406. AAAI Press (2019). <https://doi.org/10.1609/aaai.v33i01.33016399>, <https://doi.org/10.1609/aaai.v33i01.33016399>
- Grusky, M., Naaman, M., Artzi, Y.: NEWSROOM: a dataset of 1.3 million summaries with diverse extractive strategies. 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, pp. 708–719. Association for Computational Linguistics, June 2018. <https://doi.org/10.18653/v1/N18-1065>, <https://www.aclweb.org/anthology/N18-1065>
- Hermann, K.M., et al.: Teaching machines to read and comprehend. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, Montreal, Quebec, Canada, 7–12 December 2015, pp. 1693–1701 (2015). <https://proceedings.neurips.cc/paper/2015/hash/afdec7005cc9f14302cd0474fd0f3c96-Abstract.html>
- Hu, B., Chen, Q., Zhu, F.: LCSTS: a large scale Chinese short text summarization dataset. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, pp. 1967–1972. Association for Computational Linguistics, September 2015. <https://doi.org/10.18653/v1/D15-1229>, <https://www.aclweb.org/anthology/D15-1229>

8. Hua, L., Wan, X., Li, L.: Overview of the NLPCC 2017 shared task: single document summarization. In: Huang, X., Jiang, J., Zhao, D., Feng, Y., Hong, Yu. (eds.) NLPCC 2017. LNCS (LNAI), vol. 10619, pp. 942–947. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73618-1_84
9. Huang, K.H., Li, C., Chang, K.W.: Generating sports news from live commentary: a Chinese dataset for sports game summarization. In: Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, Suzhou, China, pp. 609–615. Association for Computational Linguistics, December 2020. <https://www.aclweb.org/anthology/2020.aacl-main.61>
10. Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7871–7880 (2020)
11. Lin, C.Y.: ROUGE: a package for automatic evaluation of summaries. In: Text Summarization Branches Out (2004)
12. Liu, X., Zhang, C., Chen, X., Cao, Y., Li, J.: CLTS: a new Chinese long text summarization dataset. In: Zhu, X., Zhang, M., Hong, Yu., He, R. (eds.) NLPCC 2020. LNCS (LNAI), vol. 12430, pp. 531–542. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60450-9_42
13. Liu, Y., Lapata, M.: Text summarization with pretrained encoders. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, pp. 3730–3740. Association for Computational Linguistics, November 2019. <https://doi.org/10.18653/v1/D19-1387>, <https://www.aclweb.org/anthology/D19-1387>
14. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach (2019)
15. Mihalcea, R., Tarau, P.: TextRank: bringing order into text. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, pp. 404–411. Association for Computational Linguistics, July 2004, <https://www.aclweb.org/anthology/W04-3252>
16. Nallapati, R., Zhai, F., Zhou, B.: SummaRuNNer: a recurrent neural network based sequence model for extractive summarization of documents. In: Singh, S.P., Markovitch, S. (eds.) Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, California, USA, 4–9 February 2017, pp. 3075–3081. AAAI Press (2017). <http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14636>
17. Narayan, S., Cohen, S.B., Lapata, M.: What is this article about? Extreme summarization with topic-aware convolutional neural networks. *J. Artif. Intell. Res.* **66**, 243–278 (2019)
18. Paulus, R., Xiong, C., Socher, R.: A deep reinforced model for abstractive summarization. In: 6th International Conference on Learning Representations, ICLR 2018, Conference Track Proceedings, Vancouver, BC, Canada, 30 April–3 May 2018. OpenReview.net (2018). <https://openreview.net/forum?id=HkAClQgA->
19. Sandhaus, E.: The New York times annotated corpus. In: Linguistic Data Consortium, Philadelphia, vol. 6, no. 12, p. e26752 (2008)
20. See, A., Liu, P.J., Manning, C.D.: 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), Vancouver, Canada, pp. 1073–1083. Association for Computational Linguistics, July 2017. <https://doi.org/10.18653/v1/P17-1099>, <https://www.aclweb.org/anthology/P17-1099>
21. Vaswani, A., et al.: Attention is all you need. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017, pp. 5998–6008 (2017). <https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fb0d053c1c4a845aa-Abstract.html>

22. Xi, X., Pi, Z., Zhou, G.: Global encoding for long Chinese text summarization. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **19**(6), 1–17 (2020). <https://doi.org/10.1145/3407911>
23. Zhang, F., Yao, J.G., Yan, R.: On the abstractiveness of neural document summarization. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, pp. 785–790. Association for Computational Linguistics, October–November 2018. <https://doi.org/10.18653/v1/D18-1089>, <https://www.aclweb.org/anthology/D18-1089>
24. Zhou, Q., Yang, N., Wei, F., Huang, S., Zhou, M., Zhao, T.: Neural document summarization by jointly learning to score and select sentences. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, pp. 654–663. Association for Computational Linguistics, July 2018. <https://doi.org/10.18653/v1/P18-1061>, <https://www.aclweb.org/anthology/P18-1061>



Question Generation from Code Snippets and Programming Error Messages

Bolun Yao¹, Wei Chen², Yeyun Gong³, Bartuer Zhou^{3(✉)}, Jin Xie¹, Zhongyu Wei², Biao Cheng³, and Nan Duan³

¹ Nanjing University of Science and Technology, Nanjing, Jiangsu, China
{yaob1001,csjxie}@njust.edu.cn

² Fudan University, Shanghai, China
{chenwei18,zywei}@fudan.edu.cn

³ Microsoft Research Asia, Beijing, China
{yegong,bazhou,bicheng,nanduan}@microsoft.com

Abstract. For some inexperienced developers, extracting key information from code snippets and programming error messages and turning it into a highly readable question can help them better understand, locate and search for the cause of errors. This paper proposes a **copy mechanism guided transformer** with **pre-trained programming and natural languages representations** (CMPPN) to automatically generate questions with high human readability from code snippets and programming error messages. Our CMPPN is pre-trained on a large scale code corpus with code summarization task based on transformer, and incorporated with copying mechanism in the fine-tuning phase. To evaluate our proposed model, we create a new dataset based on Stack Overflow posts, which contains code snippets, programming error messages and corresponding question headlines in 3 programming languages (Java, C# and Python). Extensive experimental results on this dataset verify the effectiveness of our CMPPN compared to baseline methods. Both dataset and model are available on <https://github.com/YuiTH/CEMS-SO>.

Keywords: Deep learning · Dataset · Question generation · Software development

1 Introduction

Debugging is a significant component in modern software development life cycle (SDLC) [6, 18]. In debugging process, developers find and resolve bugs by analyzing codes and error messages. Although the compilers or interpreters of modern programming languages usually return the call stack that lead the error when encountering exceptions, these prompts that contain a lot of redundant information are dazzling for novice developers. Even if they go to online programming forums such as *Stack Overflow*¹ and *Github Issues*² for help, due to lack of

B. Yao—Worked during the internship at Microsoft Research Asia.

¹ <https://stackoverflow.com/>.

² <https://github.com/>.

understanding of some common senses and terminologies, in many cases they cannot accurately describe the problems they have, which leads to inefficient development and communication.

In this paper, we introduce a novel task program, which is to generate a question described in natural language from code snippets and programming error messages to help developers better understand, locate and search for the cause of errors. Our task, namely **programming error related question generation** (PERQG) is similar to **text summarization** [7, 14, 25], both of which need to find key information from a large amount of potentially redundant content and refine it into a short summary. The difference is that the input of our task is a mixture of programming language and formatted error messages, and the output is a natural language description which is likely to contain some professional terms, as showed in Fig. 1. Note that since error messages can point to the location of ill codes, both the error messages and code snippets that cause the error can be easily accessed, thence our task has certain potential as a programming aid, such as an IDE plugin that can automatically provide users with debugging analysis.

Program Error Related Question Generation	Text Summarization
<p>Code:</p> <pre>con = httplib.HTTPSConnection("www.google.com") con.request("POST", "/tbproxy/spell?lang=he", data) response = con.getresponse().read()</pre> <p>Error Message:</p> <pre>Traceback (most recent call last): File "C:\Scripts\Quality\test.py", line 47, in <module> print spellFix(u"\u05d0") File "C:\Scripts\Quality\test.py", line 26, in spellFix con.request("POST", "/tbproxy/spell?lang=%s" % lang, data) ... File "C:\Python27\lib\ssl.py", line 189, in send v = self._sslobj.write(data) UnicodeEncodeError: 'ascii' codec can't encode characters</pre> <p>Title:</p> <p>How do I post unicode characters using httpplib?</p>	<p>Article:</p> <p>At least two people have died and up to 40 people are feared trapped after a roof collapsed at a construction site in eastern South Africa, emergency services say. The collapse occurred in the township of Tongaat, near Durban. Thirty people have been transported to the hospital with injuries ranging from moderate to critical, Crisis Medical operations director Neil Powell told CNN from the scene. "A local crane has arrived to start clearing the rubble to look for unaccounted people," he said. "We're waiting at this stage to get to the rubble that is still crushing an estimated 30 to 40 people." The South African Press Association reported that a 100-meter area of 18-inch-thick concrete slabs had collapsed late Tuesday afternoon local time.</p> <p>Summary:</p> <p>Two people have died after a building collapsed in South Africa, emergency services say. Crisis Medical operations director says 30 people had been taken to the hospital.</p>

Fig. 1. Comparison between program error related question generation and text summarization.

Inspired by recent works on code representation learning [3, 5, 9], we propose a transformer-based sequence to sequence (Seq2Seq) model, which maps code snippets and formatted error messages into high-dimensional vector space and generates natural language summaries word by word. We use transformer to pre-train on CodeSearchNet³ corpus with code summarization task. The task

³ <https://github.com/github/CodeSearchNet>.

takes function source code as input and corresponding human-written function comments as output, i.e., describing what the code does using natural language. In fine-tuning stage, we incorporate copying mechanism in pre-trained transformer, which allows to copy some key fragments directly from the code and error messages during decoding.

To evaluate our method, we collect a new dataset contains 203K human-written questions⁴ about Java, C# and Python each of question has at least one code snippet and one error message. To compare with our proposed method, we build sufficient baseline models, including a strong rule-based method and several non-pretrained methods. Extensive experimental results show that our method has a significant improvement in BLEU [15], ROUGE [12] and METEOR [1] compared with baseline methods.

The main **contributions** of our paper are as follows: 1) We introduce a new task to generate questions from code snippets and error messages; 2) We construct a new dataset based on Stack Overflow posts, as a benchmark of program error related question generation; 3) We propose CMPPN, a model shows superior performance on PERQG. Both dataset and model will be release for further research.

2 Dataset and Experiment Setup

We get the original data from Stack Overflow dump.⁵ Stack Overflow is a professional technical Q&A community, in which the questioner is asked to edit a complete question, including the title and specific contents, most of the questions will be accompanied by problematic codes or error messages.

To collect code snippets, error messages and corresponding question titles, we limit that each question must have at least one code snippet and one error message. Both code snippets and error messages are extracted from *code* tag from dump file in XML format, since most questioners indiscriminately format the codes and error messages with *code* tag. To distinguish between codes and error messages, we design corresponding regular patterns⁶ for error messages based on language characteristics. Those blocks with *code* tag match the pattern are considered as error messages, and others as codes. We also use another *quote* tag to identify error message. We make some trivial rules to clean up the noise, such as >>> which means the command prompt and some indentation at the beginning of the code section.

We chose three widely used programming languages to construct the dataset: Java, C# and Python. Programming languages can be divided into interpretative

⁴ 82% of the questions have at least one answer, indicating that these questions are of high quality.

⁵ <https://archive.org/details/stackexchange>.

⁶ For Java, we use “at:” plus Java top level domain package name including java org io net etc., as pattern for error message; for C#, pattern are “CS” with four digits for compile error and “Exception:” for runtime error; for Python, pattern is message start with “Traceback (most recent call last)”.

and compilable languages. In compiled languages, the code is translated into machine code by the compiler before running, both compiler and runtime could produce errors. In interpreted languages, the source code is executed line by line directly, so only the runtime may produce errors. C# is the representative of a compiled language and Python is the representative of interpreted programming languages. Java is an intermediate state of the two languages: its code needs to be compiled before it can be executed, but the format of error message during compile is same as runtime. The error message format of the three languages is shown in Fig. 2. These three programming languages are very representative for evaluating our task and models.

Java: <pre>Exception in thread "main" org.bouncycastle.cms.CMSEException: key inappropriate for algorithm. at org.bouncycastle.cms.CMSEnvelopedDataStreamGenerator.open(Unknown Source) Caused by: java.security.InvalidKeyException: Illegal key size or default parameters at javax.crypto.Cipher.a(DashoA13*..) at javax.crypto.Cipher.init(DashoA13*..) at org.bouncycastle.cms.CMSEnvelopedGenerator\$RecipientInfo.toRecipientInfo(Unknown Source) ... 4 more</pre>
C# runtime error: <pre>System.Configuration.ConfigurationErrorsException was unhandled Message="Configuration system failed to initialize" Source="System.Configuration" StackTrace: at System.Configuration.ClientConfigurationSystem.EnsureInit(String configKey)</pre>
C# compile error: <pre>System.InvalidOperationException: Unable to generate a temporary class (result=1). error CS0012: The type 'System.Object' is defined in an assembly that is not referenced. You must add a reference to assembly System.Runtime, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a.</pre>
Python: <pre>Traceback (most recent call last): File "/local/lib/python2.7/site-packages/django/core/management/_init_.py", line 443, in execute_from_command_line utility.execute() ... File "/local/lib/python2.7/site-packages/psycopg2-2.4.6-py2.7-linux-x86_64.egg/psycopg2/_init_.py" , line 178, in connect return _connect(dsn, connection_factory=connection_factory, async=async) psycopg2.OperationalError: FATAL: Peer authentication failed for user "django"</pre>

Fig. 2. Sample error messages of Java, C# and Python. Due to layout limitations, some examples have been truncated.

Popular language C/C++ was not selected because it is difficult to obtain runtime error information. Obtaining C/C++ runtime error information requires analyzing the core dump file, which is too difficult for beginners. And it is hard for obtaining development tools to obtain such information.

Relying only on error messages to generate question or summarization is sometimes insufficient. In the Fig. 1’s case, the information given by error messages is relatively more general. To generate informative and reasonable questions, error messages must be analyzed in conjunction with codes. The ground question in Fig. 1 involves a certain library not mentioned in the error message but is used in the code snippet.

We finally collected 203K questions from Stack Overflow dump with at least one code block and one traceback block. We categorize these problems according to programming language, and further divide them into training set, validation set and test set in 7:2:1 ratio. We call it **Code and Error Message Summarization of Stack Overflow posts (CEMS-SO)**. Data statistics are shown in Table 1.

Table 1. Data statistic of CEMS-SO dataset.

Language	Examples		
	Train	Valid	Test
Python	62,252	17,787	8,894
Java	61,162	17,476	8,738
C#	18,808	5,375	2,687
Overall	142,222	40,638	20,319

3 Method

3.1 Models

We adopt a sequence-to-sequence framework based on transformer [23] both in the pre-training and fine-tuning stage with a fully supervised manner. The highly parallelized transformer allows the pre-training on large-scale corpus for better representations and faster convergence of downstream tasks, and the self-attention mechanism that breaks the distance limit can better capture the long-term dependence of the input code and error messages.

3.2 Pre-training

We use code summarization, one of the downstream tasks in CodeBERT [3], as our pre-training task. Compared with the fine-tuning stage, we only use programming language as input and natural language as output. More specifically, the goal is to generate a natural language description for the codes. Our pre-training is expected to learn a good representations of programming language and natural language simultaneously. Based on the structure of encoder-decoder, the bi-modal embeddings can be learned to be aligned in high-dimensional vector space.

We pre-train CMPPN with programming language (PL)-nature language (NL) pairs from CodeSearchNet [8]. We train four programming languages together, including Python, Java, C#, Go, etc. We add Go since the language syntax of Go is similar to Java and C#. Following CodeXGLUE [13], we filter some low-quality samples, including: 1) the code cannot be parsed into an abstract syntax tree; 2) document not in English; 3) code too long or too short; 4) special pattern like “<https://>” shows in code. We finally collect 642K samples for pre-training, which is about 5 times the size of CEMS-SO dataset. We train a joint WordPiece [24] to tokenize for both PL (input) and NL (output). The tokenized PL sequence $\{[CLS], c_1, c_2, \dots, c_n\}$ is passed through the transformer encoder to obtain the hidden state, and the transformer decoder calculates the probability distribution of the output token at each step via *teacher forcing* during training. We optimize the cross entropy of the predicted probability distribution and the tokenized gold NL sequence $\{s_1, s_2, \dots, s_m, [EOS]\}$.

3.3 Fine-Tuning

In the fine-tuning stage, we use the concatenated sequence of error messages and codes as input. We use [SEP] token to separate two segments as $\{[CLS], e_1, e_2, \dots, e_n, [SEP], c_1, c_2, \dots, c_n\}$, where e_i is the i -th token of error messages and c_i is the i -th token of codes. Due to limitations of BERT position embeddings and GPU memory, we truncate the input to within 512 tokens. Our truncation rule is as follows: 1) truncate from the right to the left; 2) truncate codes first to the given lower limit (150); 3) then truncate error messages to the given lower limit (350). One exception is that when training the Python language, the error messages is truncated from left to right.⁷ We train in exactly the same way as pre-training.

Copying Mechanism. Error messages usually contains important fragments that the questioner may copy directly when editing the question, such as an unknown exception. We incorporate the copy mechanism [4] in the fine-tuning to guide the generation. In addition to directly generating words from vocabulary, the copy mechanism can directly copy a word from the input sequence as output when it takes effect.

The copy mechanism intervenes with a probability p_{copy} for each token in the decoding process, and the intervening probability is calculated from the hidden state s_t of the decoder at time step t .

$$p_{copy} = \sigma(w_{copy}^T s_t) \quad (1)$$

where w_{copy} is parameter and $p_{copy} \in [0, 1]$. When implementing the copy mechanism for the transformer, we use the encoder-decoder attention $P_{input}(y_t)$ as the probability of copying the word from input following previous work [19].

⁷ When a Python program reports an error, the information pointing to user codes usually appears at the end.

Incorporation with decoder output probability $P_{vocab}(y_t)$, final distribution of vocabulary $P(y_t)$ is:

$$P(y_t) = (1 - p_{copy})P_{vocab}(y_t) + p_{copy}P_{input}(y_t) \quad (2)$$

We incorporate copying mechanism on both LSTM and transformer to verify its effectiveness, as well as our CMPPN.

3.4 Model Settings

Our model consists of 12 layers of transformer encoder and 6 layers of transformer decoder. More specifically, each layer of encoder has 12 self-attention heads, and the hidden dimension is 768. Decoder has exact same number of self-attention heads and hidden sizes with encoder, but only 6 layers following CodeBERT [3].

We use Adam optimizer [10] with learning rate 5e-5 and train 18 epochs on CodeSearchNet. Note that our pre-training does not completely start from scratch, where the encoder uses CodeBERT-base⁸ to initialize the parameters, and the decoder initializes randomly. 8 NVIDIA Tesla V100 32 GB are used to pre-training, which takes 13 h. The effective batch size is 4096.

In the fine-tuning phase, we use the same Seq2Seq architecture with pre-training phase. We fine-tune three languages with 10 epochs on the CEMS-SO respectively, and also used Adam optimizer with 5e-5 learning rate. The best checkpoint is picked based on the BLEU-4 [15] metric of validation set. The effective batch size of fine-tuning is 512. The fine-tuning of each language requires 4 h of training on 8 NVIDIA Tesla V100 32 GB.

3.5 Automatic Metric

We employed BLEU [15] as our automatic metrics. BLEU compares the degree of overlap of n-grams in the candidate question and golden. The higher the degree of overlap, the higher the generation quality. Following CodeXGLUE [13] we report smoothed BLEU-4 score. We also report ROUGE-L [12] and METEOR [1], which is often used in summarizing task [19].

4 Experiments

4.1 Baselines

Rule Based Method. We directly use the most important key sentence in the error messages as the question. Generally, the error information reported by the interpreter and compiler contains all the call stacks of the error code, but in many cases, most of the error information comes from system or library code rather than user code. More specifically, we use the first sentence of Java and C#'s error message and fill it into a common question template. Last sentence of error message is used in Python's case. We named this method Rule-copy, which is a powerful baseline since it conforms to the user's questioning habits.

⁸ <https://huggingface.co/microsoft/codebert-base>.

Non-pretrained Based Method. We report transformer [23] and LSTM [22] as non-pretrained based baselines. We use transformer with the same structure as CMPPN randomly initialized to test the effect of the pre-training and add copying mechanism to both the two models to verify the effectiveness of the copying mechanism.

4.2 Main Result

Table 2. Automatic metrics for baseline and CMPPN.

Methods	Java			C#			Python		
	BLEU-4	ROGUE-L	METEOR	BLEU-4	ROGUE-L	METEOR	BLEU-4	ROGUE-L	METEOR
Rule-Copy	9.01	14.90	11.78	9.09	15.67	14.39	14.68	21.00	18.59
Transformer	7.23	9.31	2.44	5.97	10.07	4.40	6.48	9.19	5.35
Transformer+copy	11.36	17.26	8.51	9.72	15.60	9.59	12.22	19.07	11.74
LSTM	8.62	13.00	4.97	5.13	7.32	3.70	10.58	15.76	9.77
LSTM+copy	11.66	17.80	9.05	8.19	12.79	7.31	15.32	21.69	14.07
CMPPN wocopy	14.27	20.85	11.79	12.05	18.86	12.31	16.79	24.11	16.16
CMPPN	14.34	21.39	12.37	13.64	21.26	14.27	17.02	24.86	16.80
CMPPN-union	14.42	21.54	12.95	13.97	21.33	14.97	17.28	24.64	16.81

Rule-Based vs. Model-Based. Rule-copy performs better than LSTM and Transformer, and is close to the result of LSTM-copy and Transformer-copy, but not as good as our CMPPN. Questioners tend to copy part of the error messages directly in the title, especially in Python, therefore some outputs of Rule-copy are very similar to ground truth. In addition, the output of Rule-copy often contains keywords such as the exception type. These keywords have a high probability of appearing in the ground truth, which makes the Rule-copy a powerful baseline. However, in some complex situations, rule-based method is a little laborious. For example, the location of the key information in the error messages is not always determined, or the error messages needs to be combined with the code to infer more accurate problems like Fig. 2. Sometimes questions will condense error messages and briefly describe the scenario where the problem occurs. This kind of refinement cannot be achieved by simple cherry-picking methods. Rule-copy gets the highest METEOR in the python language. We guess it may be because our title is likely to contain some professional terms, and it is difficult for METEOR which relies on synonym mining to find these synonyms. But for complete comparison, we still report this metric.

LSTM vs. Transformer. As shown in Table 2, LSTM [22] performs better than non-pretrained transformer [23] in Java and Python. This trend remains the same even after joining the copying mechanism. But after pre-training, CMPPN outperforms LSTM by a large margin in all three languages. LSTM has great potential in this task, but limited by the autoregressive training, large-scale pre-training cannot be carried out even with less parameters. This limits its further

improvement. From these results, we can see that the representation of position information in the code is extremely important. Further research on position embedding may further improve performance.

Copying Mechanism Benefit. Copying mechanism [4] shows obvious effects in both LSTM and transformer. The result of the baseline model without the copying mechanism is even worth than rule-base method. In the Python experiment, due to the clearer error message format, the copying mechanism plays a more obvious role. After the model pre-training, the effect of the copy mechanism decreases, but it can still bring certain improvement to the model.

Pre-training Improvement. From the results of Table 2, pre-trained CMPPN far exceeds transformer and LSTM on all metrics. CMPPN nearly double the BLEU-4 score from transformer baseline. The pre-training of the summary task also significantly improves the readability of the generated results.

Joint Training vs. Independent Training. We also combine the corpus of all languages for training during fine-tuning, denoted as CMPPN-union in Table 2. Result shows joint training slightly improve the performance. Even the format of the error message and the code is different, sample from other languages can still improve the effect of model generation to generate a better question. We suspect that our union model is easy to transfer to other languages, which further illustrates the potential of the model.

4.3 Influence of Bi-modal Inputs

In order to compare the role of code and error information in problem generation, we use code only and error information only as input to train CMPPN. Same settings and truncation strategy as CMPPN are used. The results shown in Table 3 decrease when code or error messages absence. Not surprisingly, the decrease is more pronounced when only using code.

Table 3. CMPPN result with single-modal input in Java. The number in parentheses indicates the performance degradation between it and bi-modal (Both). The result of Both comes from CMPPN-union.

	BLEU-4	ROGUE-L	METEOR
Both	14.53	21.54	12.95
Error messages only	13.48 (-1.05)	21.41 (-0.13)	12.41 (-0.54)
Codes only	10.10 (-4.43)	19.15 (-2.39)	10.35 (-2.60)

5 Related Works

Deep Learning in Code Representation. In automating software engineering area, using deep learning to understand and generate code or code-related text is wildly used. New software engineering datasets and tasks are constantly being proposed. The summative work CodeXGLUE [13] integrates many code-related tasks and has become a general benchmark in this field. Recent work CodeBERT [3] GraphCodeBERT [5] pre-train BERT with CodeSearchNet [8] corpus. CodeBERT shows an impressive performance on many downstream tasks, including code refinement, cloze test, defect detection etc. For the generation task, GPT style model like GPT-C [21] shows a much better performance than BERT in code completion task thanks to the consistency of its pre-training and fine-tuning tasks. Sequence-to-sequence model like ProphetNet-X [16] use MASS [20] style denoising task on CodeSearchNet corpus. We did not use such denoising task as a pre-training task in CMPPN. Because we want to train the model to generate a summary sentence similar to the final PERQG task.

Text Summarization. Abstraction summarization is a generation task where is not constrained to reusing the phrases or sentences from input text. Recent work BART [11], UNILM [2] and ProphetNet [17] applied pre-training model on abstraction summarization. The unsupervised pre-trained model using unlabeled corpus produced great performance after fine-tuning. Inspired by the above work, we hope to help developers understand the problem by condensing lengthy error messages and generating brief questions.

6 Conclusion

This paper introduces a novel task program error related question generation, with a new dataset CEMS-SO in three languages. The goal of the task is to summarize a readable headline from the long error message with a lot of terminology and code segment. In order to obtain high-quality data, we designed a set of rules to filter Stack Overflow questions, and finally got 203K samples. We also introduce CMPPN, a sequence-to-sequence transformer model that use a copy mechanism and pre-training, which reach state-of-the-art performance in our task. In the future, we will further study this task and explore the application of CMPPN in similar application.

References

1. Banerjee, S., Lavie, A.: METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Ann Arbor, Michigan, pp. 65–72. Association for Computational Linguistics, June 2005

2. Dong, L., et al.: Unified language model pre-training for natural language understanding and generation. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems, pp. 13063–13075 (2019)
3. Feng, Z., et al.: CodeBERT: a pre-trained model for programming and natural languages. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pp. 1536–1547 (2020)
4. Gu, J., Lu, Z., Li, H., Li, V.O.: Incorporating copying mechanism in sequence-to-sequence learning. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1631–1640 (2016)
5. Guo, D., et al.: GRAPHCODEBERT: pre-training code representations with data flow. In: International Conference on Learning Representations (2020)
6. Hailpern, B., Santhanam, P.: Software debugging, testing, and verification. IBM Syst. J. **41**(1), 4–12 (2002)
7. Hermann, K.M., et al.: Teaching machines to read and comprehend. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 28. Curran Associates, Inc. (2015)
8. Husain, H., Wu, H.H., Gazit, T., Allamanis, M., Brockschmidt, M.: CodeSearchNet challenge: evaluating the state of semantic code search. arXiv preprint [arXiv:1909.09436](https://arxiv.org/abs/1909.09436) (2019)
9. Kanade, A., Maniatis, P., Balakrishnan, G., Shi, K.: Pre-trained contextual embedding of source code. arXiv preprint [arXiv:2001.00059](https://arxiv.org/abs/2001.00059) (2019)
10. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
11. Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint [arXiv:1910.13461](https://arxiv.org/abs/1910.13461) (2019)
12. Lin, C.Y.: ROUGE: a package for automatic evaluation of summaries. In: Text Summarization Branches Out, Barcelona, Spain, pp. 74–81. Association for Computational Linguistics, July 2004
13. Lu, S., et al.: CodeXGLUE: a machine learning benchmark dataset for code understanding and generation. arXiv preprint [arXiv:2102.04664](https://arxiv.org/abs/2102.04664) (2021)
14. Narayan, S., Cohen, S.B., Lapata, M.: Don’t give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium (2018)
15. 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 of the Association for Computational Linguistics, pp. 311–318 (2002)
16. Qi, W., et al.: ProphetNet-x: large-scale pre-training models for English, Chinese, multi-lingual, dialog, and code generation. arXiv preprint [arXiv:2104.08006](https://arxiv.org/abs/2104.08006) (2021)
17. Qi, W., et al.: ProphetNet: predicting future n-gram for sequence-to-sequence pre-training. arXiv preprint [arXiv:2001.04063](https://arxiv.org/abs/2001.04063) (2020)
18. Ruparelia, N.B.: Software development lifecycle models. ACM SIGSOFT Softw. Eng. Notes **35**(3), 8–13 (2010)
19. See, A., Liu, P.J., Manning, C.D.: Get to the point: summarization with pointer-generator networks. arXiv preprint [arXiv:1704.04368](https://arxiv.org/abs/1704.04368) (2017)
20. Song, K., Tan, X., Qin, T., Lu, J., Liu, T.Y.: MASS: masked sequence to sequence pre-training for language generation. In: International Conference on Machine Learning, pp. 5926–5936. PMLR (2019)

21. Svyatkovskiy, A., Deng, S.K., Fu, S., Sundaresan, N.: IntelliCode compose: code generation using transformer. In: Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp. 1433–1443 (2020)
22. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint [arXiv:1503.00075](https://arxiv.org/abs/1503.00075) (2015)
23. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
24. Wu, Y., et al.: Google’s neural machine translation system: bridging the gap between human and machine translation. arXiv preprint [arXiv:1609.08144](https://arxiv.org/abs/1609.08144) (2016)
25. Zhang, J., Zhao, Y., Saleh, M., Liu, P.: PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In: International Conference on Machine Learning, pp. 11328–11339. PMLR (2020)



Extractive Summarization of Chinese Judgment Documents via Sentence Embedding and Memory Network

Yan Gao¹ , Zhengtao Liu¹ (✉) , Juan Li² , and Jin Tang¹

¹ School of Automation, Central South University, Changsha, China
{gaoyan,tjin}@csu.edu.cn

² School of Law, Central South University, Changsha, China

Abstract. A rapidly rising number of open judgment documents has increased the requirement for automatic summarization. Since Chinese judgment documents are characterized by a lengthy and logical structure, extractive summarization is an effective method for them. However, existing extractive models generally cannot capture information between sentences. In order to enable the model to obtain long-term information in the judgment documents, this paper proposes an extractive model using sentence embeddings and a two-layers memory network. A pre-trained language model is used to encode sentences in judgment documents. Then the whitening operation is applied to get isotropic sentence embeddings, which makes the subsequent classification more accurate. These embeddings are fed into a unidirectional memory network to fuse previous sentence embeddings. A bidirectional memory network is followed to introduce position information of sentences. The experimental results show that our proposed model outperforms the baseline methods on the SFZY dataset from CAIL2020.

Keywords: Extractive summarization · Memory network · Sentence embedding · Judgment documents

1 Introduction

Chinese judgment documents record the process and result of legal cases, making the trial activities open to the public and publishing the reasons, bases and results of their decisions. It is the evidence for the courts to determine and allocate the substantive rights and obligations of the parties. A judgment document contains a wealth of information, including the case category, the cause of action, claims, the facts and reasons of disputes, evidences, court decision, and the cited legislation. None the less, the judgment documents use domain knowledge in a lengthy way and their structures tend to be complex, which pose barriers to an ordinary reader to read and understand. In the meantime, China has witnessed a surge in judgment documents published online ever since the Supreme People's Court (SPC) released Provisions on the Online Issuance of Judgment Documents

by People's Courts in November 2013. As of 22:00 on May 28, 2021, the number of documents on Chinese Judgment Documents Website¹ has exceeded 119 million, with a total of over 62 billion visits. This calls for automatic summarization of open judgment documents in an effective way.

Summarization of judgment documents is a compression of court decisions, which mainly consists of the adjudication process, facts, reasons and the cited legislation in the process of a trial. At present, the automatic text summarization methods broadly include extractive and abstractive. Extractive summarization directly selects sentences from the original text as summarization. Researches on summarizing legal texts are generally based on this method [1, 2]. Abstractive summarization creates new sentences through the fusion of the original information, but it has two shortcomings: out-of-vocabulary (OOV) and repetition. Pointer-Generator [3], a typical framework for abstractive summarization, proposed copy and coverage mechanism to ease the above problems. However, this method is unsatisfactory when dealing with a large number of Chinese OOV in lengthy legal documents. In addition, the cost of calculating is unaffordable to the general readers.

山西孟县农村商业银行股份有限公司与赵贵青、梁翠平借款合同纠纷一审民事判决书山西省孟县人民法院民事判决书(2017)晋0322民初1203号原告：山西孟县农村商业银行股份有限公司。法定代表人：耿立涛，公司董事长。委托诉讼代理人：赵学耀，男，1975年12月15日生，汉族，山西省孟县。被告：赵贵青，男，1976年4月20日生，汉族，山西省孟县。被告：梁翠平，女，1976年3月28日生，汉族，山西省孟县。原告山西孟县农村商业银行股份有限公司与被告赵贵青、梁翠平借款合同纠纷一案，本院于2017年10月19日立案后，依法适用普通程序，公开开庭进行了审理。原告山西孟县农村商业银行股份有限公司的委托诉讼代理人赵学耀到庭参加诉讼，被告赵贵青、梁翠平，经本院合法传唤，无正当理由拒不到庭。本案现已审理终结。山西孟县农村商业银行股份有限公司向本院提出诉讼请求：1、依法判令被告赵贵青归还原告借款本金12万元及利息，梁翠平承担连带清偿责任；2、诉讼费用由被告承担。事实与理由：被告赵贵青由梁翠平担保于2015年12月25日向山西孟县农商银行营业部借款12万元，约定还款日期为2016年12月24日。借款到期后，经公司多次催要未果。为保护原告的合法权益，特诉至法院。被告赵贵青、梁翠平未到庭，亦未提交书面答辩意见及证据材料。经审理查明，被告赵贵青由梁翠平担保于2015年12月25日向山西孟县农商银行营业部借款12万元，到期日期为2016年12月24日，贷款用途为购房，借款年利率为4.35%。逾期利率为原借款利率的1.5倍。赵贵青从2015年12月25日开始欠息，截止2016年12月24日欠息5292.5元，2016年12月24日到2017年11月6日逾期利息为6894.75元，共计欠息12187.25元，本息合计132187.25元。上述事实，有《贷款合同》、《保证合同》、借款借据、欠息证明、赵贵青、梁翠平身份证件信息及庭审笔录在案佐证。本院认为，原告山西孟县农村商业银行股份有限公司与被告赵贵青签订的《贷款合同》以及与被告梁翠平签订《保证合同》系各方当事人真实意思表示，未违反法律法规的强制性规定，应认定为合法有效的民事合同。合同各方均应按照约定自觉履行合同义务。现原告已履行了放贷义务，被告在借款到期后，未按合同约定偿还借款本息，其行为属违约行为应履行偿还义务。被告梁翠平作为保证人，在借款人赵贵青未按期偿还借款的情况下，应当承担相应的连带保证责任。其承担连带保证责任后有权向赵贵青追偿。原告诉讼请求合理、合法，本院予以支持。综上所述，依照《中华人民共和国合同法》第一百零七条、第二百零五条、第二百零六条、第二百零七条、《中华人民共和国担保法》第十八条、第二十一条、第三十一条、《中华人民共和国民事诉讼法》第一百四十四条之规定，判决如下：一、被告赵贵青于本判决生效之日起十五日内返还原告山西孟县农村商业银行股份有限公司借款本金12万元，并支付截止2017年11月6日的利息12187.25元。二、被告梁翠平对被告赵贵青的上述债务承担连带清偿责任，被告梁翠平承担连带清偿责任后有权向被告赵贵青追偿。如果未按本判决指定的期间履行给付金钱义务，应当按照《中华人民共和国民事诉讼法》第二百五十三条规定，加倍支付迟延履行期间的债务利息。案件受理费2700元，由被告赵贵青、梁翠平负担。如不服本判决，可在判决书送达之日起十五日内，向本院递交上诉状，并按对方当事人的人数提出副本，上诉于山西省阳泉市中级人民法院。审判长郭树祥；人民陪审员霍俊弟；人民陪审员王京二〇一七年十二月二十八日书记员张剑锋

Fig. 1. A manually marked judgment document from Chinese Judgment Documents Website. Blue words (bold) represent the key sentences selected as extractive summarization, red words (italic) represent keywords in these sentences. (Color figure online)

¹ <https://wenshu.court.gov.cn>.

Unlike ordinary texts, judgment documents are highly logical. A manually marked judgment document is shown in Fig. 1. Although extractive method has been widely used in summarization task [4–15], how to apply it in judgment documents reminds a challenge, since when sentences are selected as summarization, attention should be paid to not only the word relationship therein, but also the semantic relationship in context. To address this problem, we propose an extractive summarization model based on sentence embedding and memory network. Pre-trained language model is used as an encoder to obtain embeddings of sentences in a judgment document, which are fused with word feature vectors. The whitening operation is applied on these sentence embeddings, which are then fed into a two-layers memory network that makes sentence embeddings integrate semantic relationship in context.

2 Related Work

How to employ computers to process legal texts is always a core problem in the AI&Law domain [16–18]. With the development of summarization technology in the general domain, research on extractive summarization of legal texts attracts growing attention. Polsley et al. [1] designed CaseSummarizer, a legal text summarization system based on word frequency and domain-specific knowledge. Liu et al. [2] used language information, statistical information, legal information and word embedding as features to construct a text classifier for summarization.

Extractive summarization selects important sentences directly from the original text, then sorts and reorganizes them to form summarization. Extractive methods can be divided into unsupervised and supervised.

TextRank [4] is a typical unsupervised method for extraction by computing similarity between sentences. Liu et al. [5] applied sparse coding techniques into the extractive summarization task and regarded the task as an optimization problem. Li et al. [6] incorporated more detailed grammatical units (nouns and phrasal verbs) based on sparse coding and rewrote named entities to improve the quality of summarization. Fevry et al. [7] added noise to extend sentences and trained a denoising auto-encoder to recover the original, constructing an end-to-end training regime without the need for any examples of compressed sentences. Zheng et al. [8] revisited popular graph-based ranking algorithms and modified how node (aka sentence) centrality is computed: employed BERT to capture sentential meaning and built graphs with directed edges arguing that the contribution of any two nodes to their respective centrality is influenced by their relative position in a document.

In supervised methods, extractive summarization is regarded as binary classification problems. Cao et al. [9] developed a Ranking framework upon Recursive Neural Networks (R2N2). They formulated the sentence ranking task as a hierarchical regression process, which simultaneously measures the salience of a sentence and its constituents in the parsing tree. Cheng et al. [10] proposed a data-driven approach based on Long Short-Term Memory (LSTM) [19] and continuous sentence features. Nallapati et al. [11] presented SummaRuNNer, a

Recurrent Neural Network (RNN) based sequence model for extractive summarization, in which a hierarchical neural network is used to extract the features among words, sentences and documents. With pre-trained language model raised up, BERT [20] was firstly applied to extractive summarization in [12]. In this work, [CLS] is added before each sentence in a document to get their embeddings, which are fed into a summarization layer to get the final summary. The summarization layer can be built in three ways: set up a classifier directly by matrix operation, add Transformer [21] before the classifier, add RNN before the classifier.

3 Proposed Model

In this section, we will describe the extractive summarization model in detail. First, we briefly introduce a BERT-based encoder with the self-attention mechanism. Then we adopt the whitening operation in the sentence embedding layer. Finally, these embeddings are fed into a two-layers transformed memory network to fuse other embeddings in context. The framework of our proposed model is shown in Fig. 2.

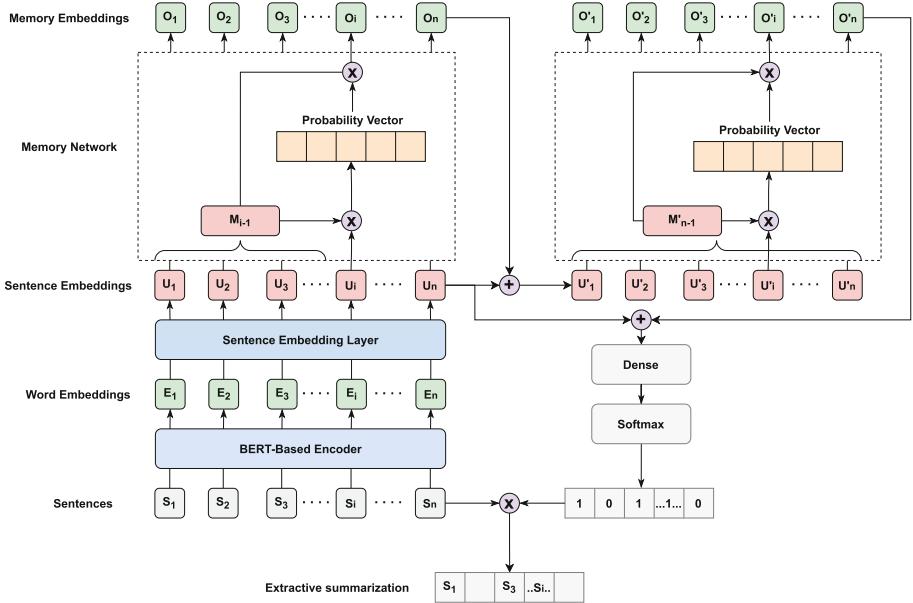


Fig. 2. Overview of proposed model based on sentence embeddings and memory network.

3.1 BERT-Based Encoder

For a BERT-Based encoder, the input representation of a given token is constructed by summing the corresponding token embeddings, segment embeddings, and position embeddings [20]. To collect global semantic information, the encoder is based on a self-attention mechanism, which is calculated as:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

where Q denotes a target word, K denotes each word in context of the target word and V denotes the value of the target word and its context. Softmax is used to normalize the similarity, d_k is a scaling matrix.

A judgment document is segmented into a sentence set $\{S_i\}_{i=1}^n$. Words in sentences are converted into input embeddings and transformed into word embeddings $\{E_i\}_{i=1}^n$ through the BERT-Based encoder. The shape of E_i is $(m, 768)$, m denotes the length of character string '[CLS] S_i [SEP]'. It is worth mentioning that parameters of the Bert-Based Encoder are frozen when training overall model.

3.2 Sentence Embedding Layer

In the sentence embedding layer, word embeddings in a sentence are used to compute an embedding representing the whole sentence. We firstly introduce two approaches to obtain initial sentence embeddings. Then these embeddings are transformed into final sentence embeddings through whitening operation.

First Token. [CLS] is the first token of the input sequence. It fuses the semantic information of the other tokens. The word embedding corresponding [CLS] could be used as an initial sentence embedding.

Average Pooling. It is found in [22] that an average pooling over word embeddings in the last one or two layers of BERT outperforms [CLS]. The average of word embeddings could be computed as an initial sentence embedding.

On semantic textual similarity tasks, [23] transformed the anisotropic sentence embedding distribution to a smooth and isotropic Gaussian distribution. Whitening operation in traditional machine learning can also achieve a similar effect [24]. To improve the effect of sentence classification, we adopt the whitening operation to transform the initial sentence embeddings.

First, the mean value μ and covariance matrix C of initial sentence embedding $\mathbf{X} = \{X_i\}_{i=1}^n$ are calculated as follows:

$$\mu = \frac{1}{n} \sum_{i=1}^n X_i \quad (2)$$

$$C = \frac{1}{n} \sum_{i=1}^n (X_i - \mu)^T (X_i - \mu) \quad (3)$$

The purpose of whitening operation is to transform μ into zero and C into an identity matrix. This operation can be denoted as Eq. 4, in which U_i denotes the final sentence embeddings, W denotes a transformation matrix.

$$U_i = (X_i - \mu)W \quad (4)$$

We denote the covariance matrix of final sentence embeddings $\{U_i\}_{i=1}^n$ as C' :

$$C' = W^T C W = I \quad (5)$$

$$C = (W^T)^{-1} W^{-1} \quad (6)$$

The covariance matrix C is a positive definite symmetric matrix. According to [24], SVD decomposition can be applied on C as Eq. 7, and the transformation matrix W is solved by Eq. 8, in which α is an orthogonal matrix, β is a diagonal matrix.

$$C = \alpha \beta \alpha^T \quad (7)$$

$$W = \alpha \beta^{-\frac{1}{2}} \quad (8)$$

After the whitening operation as Eq. 4, the final sentence embeddings that are more isotropic than the initial, which is beneficial to make the subsequent classification more accurate.

3.3 Memory Network

Memory network is first proposed in [25] for question answering (QA) task. In this paper, the memory network is transformed to make sentence embeddings integrate semantic relationships in context. There are two layers in the transformed memory network, the first layer is a unidirectional network, the second is a bidirectional network.

Unidirectional Memory Network. There is a strong logic between the sentences in a judgment document. To make the classification more accurate, we combine a target sentence embedding with previous sentence embeddings via the unidirectional memory network.

Sentence embeddings $\{U_i\}_{i=1}^n$ are fed into the network. The purpose is to output a set of memory embeddings $\{O_i\}_{i=1}^n$. To compute O_i , we firstly convert $\{U_j\}_{j=1}^{i-1}$ into a vector M_{i-1} of size $(i-1) \times 768$. In the embedding space, the match between U_i and its previous sentence embeddings M_{i-1} is calculated by taking their inner product:

$$Match = U_i M_{i-1}^T \quad (9)$$

Then, a softmax function is used to normalize the match vector and obtain a probability vector P :

$$P = softmax(Match) \quad (10)$$

The memory embedding O_i is a sum over M_{i-1} , weighted by the probability vector:

$$O_i = PM_{i-1} \quad (11)$$

Bidirectional Memory Network. In the unidirectional memory network, input embeddings are fused indiscriminately. To enhance the order of sentence embeddings, we introduce position information by feeding $\{U_i\}_{i=1}^n$ and $\{O_i\}_{i=1}^n$ into the bidirectional memory network. The output is a set of enhanced memory embeddings $\{O'_i\}_{i=1}^n$.

First, sentence embeddings and memory embeddings are concatenated as input $\{U'_i\}_{i=1}^n$. Then $\{U'_j\}_{j \neq i}^n$ are converted into a vector M'_{n-1} of size $(n-1) \times 768$. Then the match between U'_i and M'_{n-1} is computed by taking the inner product followed by a softmax:

$$P' = \text{softmax}(U'_i M'^T_{n-1}) \quad (12)$$

The enhanced memory embedding O'_i is a sum over M'_{n-1} , weighted by the probability vector P' :

$$O'_i = P' M'_{n-1} \quad (13)$$

3.4 Classification and Extraction

Enhanced memory embeddings O' are concatenated after sentence embeddings U as extra features:

$$UO' = \text{concat}(U, O') \quad (14)$$

where $\text{concat}(\cdot)$ is the splicing function. These embeddings are fed into a Dense network followed by a softmax function, for dimensionality reduction and classification:

$$Y = \text{softmax}(\text{Dense}(UO')) \quad (15)$$

where Y is a zero-one vector. Finally, extractive summarization is obtained by (S denotes the sentence set):

$$S \times Y \quad (16)$$

Since the parameters of the encoder are frozen and the memory networks do not introduce additional parameters, all the trainable parameters of the proposed model are in the Dense layer.

4 Experiments

In this section, we first introduce the experimental settings. Then we report the results of our experiments by comparing the performance of our model with some classic methods and discussing the impact of different components in the model. The codes are publicly available at github.²

² <https://github.com/csu-lzt/judgment-pytorch>.

4.1 Settings

Dataset. To verify the effectiveness of the proposed model, the SFZY dataset available from CAIL2020³ is used in the experiment. It contains 13.5K Chinese judgment documents, with a label of each sentence (“0” or “1”) and manual summarization. All the sentences labeled “1” in a judgment document make up the extractive summarization. The length quantiles of documents and summarizations are shown in Table 1.

Table 1. The length quantile of document and summarization (words)

Text type	Maximum	99 quantile	98 quantile	95 quantile	90 quantile
Judgment documents	13064	6701	5814	4767	3982
Extractive summarization	3790	1762	1583	1357	1186

Evaluation Metric. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [26] is adopted as evaluation metric in this paper. We use ROUGE-1, ROUGE-2 and ROUGE-L, which respectively represent the overlap of N-gram and the longest common sequence between the reference summarization and the generated summarization.

Training Details. In the BERT-Based encoder, RoBERTa-wwm⁴ is used. We randomly split the dataset into training, verification and test set, which respectively contains 10K, 1.7K and 1.7K judgment documents. The training set is used to train the model. The verification set is used for the best model selection and hyperparameter optimization. The test set is used for evaluation. The maximum length of a sentence is set to 512. Sparse categorical crossentropy is used as the loss function. Adam with a $2e^{-5}$ initial learning rate is used as the optimizer. A NVIDIA GPU V100 with 32 GB is used in our experiment.

4.2 Comparison of Results

To evaluate the performance of our model, we compare it with the following methods: **LEAD-3**, the baseline of CAIL2020, which selects consecutive sentences with particular keywords or locations in a judgment document as summarization. **TextRank** [4], which calculates the score of each sentence by the similarity between sentences and keywords. Then N sentences with the highest score are selected to compose the summary. In this experiment, higher scores are given to the legal information and N is set to 8. **FastText** [27], in which word features are averaged to form sentence representations and are used to extract sentences as summarization. **TextCNN** [28], convolution neural network (CNN) is applied to the sentences classification task. The kernel of several

³ <http://caill.cipsc.org.cn/>.

⁴ <https://github.com/ymcui/Chinese-BERT-wwm>.

different sizes is used to extract the key information and capture the local correlation in sentences. **TextRNN** [29], recursive neural network (RNN) is applied to the sentences classification task, which captures long-distance dependencies in a document. **RoBERTa**, sentence embeddings obtained by RoBERTa-wwm are directly used in classification.

Table 2 shows the experimental results on the SFZY test set, in which **P** denotes Precision, **R** denotes Recall, **F1** denotes the F1 value of P and R. It can be seen that our proposed model performs better than other comparative experiments on most metrics. The results confirm that the method of combining sentence embeddings with the memory network is feasible.

Table 2. Extractive summarization result on SFZY test set

Model	ROUGE-1(%)			ROUGE-2(%)			ROUGE-L(%)			Accuracy(%)
	P	R	F1	P	R	F1	P	R	F1	
LEAD	40.02	20.62	26.01	20.69	10.15	13.02	31.54	15.83	20.16	–
TextRank	38.93	63.34	47.28	25.56	41.62	31.05	33.17	53.94	40.28	–
FastText	36.32	74.16	47.16	25.07	50.92	32.49	31.49	64.29	40.90	89.95
TextCNN	35.47	79.33	47.55	25.39	56.32	33.96	31.49	70.41	42.23	90.28
TextRNN	35.79	71.72	45.90	24.23	48.37	31.03	30.44	61.07	39.06	88.57
RoBERTa	32.43	80.25	46.56	25.32	59.02	33.87	30.82	71.14	39.81	89.98
Our Model	33.65	83.14	48.56	24.70	60.47	34.08	30.49	75.13	42.26	91.86

To explore the contribution of each part of the model, we decompose the model into the following five types: **RoBERTa-wwm**(baseline), initial sentence embeddings are directly used for classification and extraction. **Whitening**, which applied the whitening operation on initial sentence embeddings. **Whitening+UMN**, which adds the unidirectional memory network to the whitening operation. **Whitening+BMN**, which adds the bidirectional memory network to the whitening operation. **Whitening+UMN+BMN**, the whole proposed model, which adds the two-layers memory network to the whitening operation.

Table 3 shows the experimental results with different component models. It can be seen that by adding the whitening operation, the accuracy of the model improves by 0.86%, and all the ROUGE metrics have been improved slightly. The unidirectional memory network can also improve the model. However, directly adding the bidirectional memory network, accuracy and ROUGE decreased, because we design it just for introducing the position information. After the whitening operation and the two-layers memory network, the model has been significantly improved compared to the baseline. The accuracy improves by 1.88%, ROUGE-1, ROUGE-2 and ROUGE-L respectively increase by 2.00%, 0.21%, 2.35%. This shows that it is effective to use the sentence embeddings, whitening operation and memory network to get extractive summarization of Chinese judgment document.

Table 3. Experimental results with different component models

Model	ROUGE-1(%)	ROUGE-2(%)	ROUGE-L(%)	Accuracy(%)
RoBERTa-wwm	46.56	33.87	39.81	89.98
+Whitening	47.33	33.93	40.90	90.84
+Whitening+UMN	48.02	34.01	41.96	91.02
+Whitening+BMN	47.28	33.90	40.88	90.78
+Whitening+UMN+BMN	48.56	34.08	42.26	91.86

5 Conclusion

In this paper, we propose an extractive summarization model for Chinese judgment documents based on sentence embedding and memory network. The model has the following innovations: First, the whitening operation is used on the sentence embeddings to make the classification more accurate. Second, we transform the structure of memory network and apply it to sentence embeddings. Last, a unidirectional memory network and a bidirectional memory network are combined to make sentence embeddings fuse semantic relationships in context and introduce position information. Experimental results show the effectiveness of this work. In the future, we plan to build a knowledge graph based on public judgment documents and integrate more external legal knowledge into our model.

Acknowledgements. This work is supported by The National Social Science Foundation Project of China (No. 20BFX077).

References

- Polsley, S., Jhunjhunwala, P., Huang, R.: CaseSummarizer: a system for automated summarization of legal texts. In: Proceedings of COLING 2016, the 26th international conference on Computational Linguistics: System Demonstrations, pp. 258–262 (2016)
- Liu, C.L., Chen, K.C.: Extracting the gist of Chinese judgments of the supreme court. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law, pp. 73–82 (2019)
- See, A., Liu, P.J., Manning, C.D.: 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), pp. 1073–1083 (2017)
- Mihalcea, R., Tarau, P.: TextRank: bringing order into text. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pp. 404–411 (2004)
- Liu, H., Yu, H., Deng, Z.H.: Multi-document summarization based on two-level sparse representation model. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29 (2015)
- Li, P., Bing, L., Lam, W., Li, H., Liao, Y.: Reader-aware multi-document summarization via sparse coding. In: Proceedings of the 24th International Conference on Artificial Intelligence, pp. 1270–1276 (2015)

7. Fevry, T., Phang, J.: Unsupervised sentence compression using denoising auto-encoders. In: Proceedings of the 22nd Conference on Computational Natural Language Learning, pp. 413–422 (2018)
8. Zheng, H., Lapata, M.: Sentence centrality revisited for unsupervised summarization. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 6236–6247 (2019)
9. Cao, Z., Wei, F., Dong, L., Li, S., Zhou, M.: Ranking with recursive neural networks and its application to multi-document summarization. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29 (2015)
10. Cheng, J., Lapata, M.: Neural summarization by extracting sentences and words. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 484–494 (2016)
11. Nallapati, R., Zhai, F., Zhou, B.: SummaRuNNer: a recurrent neural network based sequence model for extractive summarization of documents. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017)
12. Liu, Y.: Fine-tune BERT for extractive summarization. arXiv preprint [arXiv:1903.10318](https://arxiv.org/abs/1903.10318) (2019)
13. Bouscarrat, L., Bonneau, A., Peel, T., Pereira, C.: STRASS: a light and effective method for extractive summarization based on sentence embeddings. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pp. 243–252 (2019)
14. Yuan, R., Wang, Z., Li, W.: Fact-level extractive summarization with hierarchical graph mask on BERT. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 5629–5639 (2020)
15. Zhou, Q., Wei, F., Zhou, M.: At which level should we extract? An empirical analysis on extractive document summarization. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 5617–5628 (2020)
16. Rissland, E.L., Ashley, K.D., Loui, R.P.: AI and law: a fruitful synergy. Artif. Intell. **150**(1–2), 1–15 (2003)
17. Bench-Capon, T., et al.: A history of AI and law in 50 papers: 25 years of the international conference on AI and law. Artif. Intell. Law **20**(3), 215–319 (2012)
18. Surden, H.: Artificial intelligence and law: an overview. Ga. St. UL Rev. **35**, 1305 (2018)
19. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
20. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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), pp. 4171–4186 (2019)
21. Vaswani, A., et al.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 6000–6010 (2017)
22. Reimers, N., et al.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (2019)

23. Li, B., Zhou, H., He, J., Wang, M., Yang, Y., Li, L.: On the sentence embeddings from pre-trained language models. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 9119–9130. Association for Computational Linguistics, November 2020. <https://doi.org/10.18653/v1/2020.emnlp-main.733>, <https://www.aclweb.org/anthology/2020.emnlp-main.733>. Online
24. Su, J., Cao, J., Liu, W., Ou, Y.: Whitening sentence representations for better semantics and faster retrieval. CoRR [arXiv:2103.15316](https://arxiv.org/abs/2103.15316) (2021)
25. Weston, J., Chopra, S., Bordes, A.: Memory networks. arXiv preprint [arXiv:1410.3916](https://arxiv.org/abs/1410.3916) (2014)
26. Lin, C.Y.: ROUGE: a package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp. 74–81 (2004)
27. Joulin, A., Grave, É., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pp. 427–431 (2017)
28. Kim, Y.: Convolutional neural networks for sentence classification (2014)
29. Liu, P., Qiu, X., Huang, X.: Recurrent neural network for text classification with multi-task learning. arXiv preprint [arXiv:1605.05101](https://arxiv.org/abs/1605.05101) (2016)

Question Answering



ThinkTwice: A Two-Stage Method for Long-Text Machine Reading Comprehension

Mengxing Dong¹, Bowei Zou², Jin Qian¹, Rongtao Huang¹, and Yu Hong¹(✉)

¹ Computer Science and Technology, Soochow University, Suzhou, China

² Institute for Infocomm Research, Singapore, Singapore

zou_bowei@i2r.a-star.edu.sg

Abstract. Long-text machine reading comprehension (LT-MRC) requires machine to answer questions based on a lengthy text. Despite transformer-based models achieve promising results, most of them are incapable of dealing with long sequences for their time-consuming. In general, a proper solution by sliding window splits the passage into equally spaced fragments, then predicts the answer based on each fragment separately without considering other contextual fragments. However, this approach suffers from lack of long-distance dependency, which severely damages the performance. To address this issue, we propose a two-stage method ThinkTwice for LT-MRC. ThinkTwice casts the process of LT-MRC into two main steps: 1) it firstly retrieves several fragments that the final answer is most likely to lie in; 2) then extracts the answer span from these fragments instead of from the lengthy document. We do experiments on NewsQA. The experimental results demonstrate that ThinkTwice can capture the most informative fragments from a long text. Meanwhile, ThinkTwice achieves considerable improvements compared to all existing baselines. All codes have been released at Github (<https://github.com/Walle1493/ThinkTwice>).

Keywords: Machine reading comprehension · Question answering · Long text

1 Introduction

Machine reading comprehension (MRC) [6], which aims to teach machines to answer questions over given passages, has always been one of the cutting-edge research in the field of natural language processing (NLP). Many pre-trained language models (PLMs) [19] have achieved promising results due to their multi-layer architectures and self-attention mechanisms [19].

Despite the success in short text situation, existing MRC systems (even other NLP systems) cannot effectively deal with long sequences, due to the length limit of the PLMs¹. Meanwhile, if simply increasing the input length, the complexity

¹ Such as the max position embedding length of BERT is 512.

of the model ($O(n^2)$) will also quadratically scale and lead to a dimensional disaster.

The intuitive solutions are truncation [15, 24] and sliding window [9]. The former is to truncate the long text to the model-acceptable length, while the latter divides the passage into fixed-length blocks and predicts the answer for each block. Both two methods suffer from sacrificing part of texts or discarding the contextual information. Since the radical problem lies in the high complexity of time and space, another line of research by simplifying transformers [2, 5, 26] has been proposed. However, few of them have been applied to the real world due to their various problems.

Inspired by human reading behavior, we propose a ThinkTwice method (as illustrated in Fig. 1.) to address the challenges of long-text machine reading comprehension (LT-MRC), which is a two-stage strategy. ThinkTwice compresses the long text into a short one instead of merely simplifying self-attention based models. Specially, when a human reads a long passage with doubts in his mind, he firstly selects several fragments unconsciously that are related to the given question and sorts them in a sensory register, then these fragments are put into human’s working memory [1] to infer an answer. Following this human-like behavior, the ThinkTwice utilizes a Retriever and a Reader module to implement the above information compression/filtering and question answering functions, respectively. In addition, a Segmentor and a Fusion module are employed before and after the Retriever to complete the long text segmentation and the integration of selected key fragments.

We evaluate and verify the proposed ThinkTwice method on the NewsQA dataset [18], within which the text length of the news article is generally long. The experimental results demonstrate that our method achieves significant and substantial improvement as compared to all baselines [3, 7, 17]. In particular, our method reaches a considerable performance by selecting a few informative paragraphs in the first retrieval stage, which greatly speeds up the inference process in the second stage.

Our main contributions are summarized as follows:

1. We propose a novel approach ThinkTwice on LT-MRC, which is capable of compressing long texts into short ones instead of directly dealing with lengthy articles.
2. Experimental results show that our approach ThinkTwice achieves considerable improvements on four main pre-trained language models [3, 7, 12, 13] on the long text dataset NewsQA [18].

2 Related Work

Machine reading comprehension (MRC) [6] is one of the most important tasks in NLP. In recent years, with the development of pre-trained language models (PLMs) [19] and an effective architecture of transformers, the research has been invested in long-text machine reading comprehension (LT-MRC), which has attracted great interest with the release of a diversity of benchmark datasets such as NewsQA [18] and TriviaQA [8].

To handle the overlong text, the simplest solution is to truncate the document, commonly used for text classification tasks [24]. This method suffers from losing considerable context once the text lengths are exceedingly large. Sliding window [9, 22], processing continuous 512 tokens by BERT or other PLMs, is the most straightforward solution. This approach sacrifices the long-distance dependency within contexts, which performs badly in some complicated tasks such as multi-hop QA [25]. Since the sequence length that BERT can accommodate is restricted to 512 due to the high consumption of time and space in the self-attention mechanism, another way of research attempts to simplify the structure of transformers, such as Longformer [2], Big Bird [26], and Ernie Doc [5], etc. However, various issues still exist in these lightweight transformers. For example, Longformer is yet in demand for enormous memory if 4,096 tokens are urgent to be dealt with simultaneously, which is not friendly to the memory of GPU.

Our work is mainly inspired by the way humans think [1]: Incoming information first enters the sensory register, where it resides for a brief time, then decays and stays the most important part; the short-term storage is the subject’s working memory which receives selected inputs from the sensory register and deals with them. Similar lines of research, where the authors put under scrutiny the function of the sensory register have been undertaken in text summarization [10], which aims at compressing the long document into a shorter form that conveys the main idea of the original document.

3 Method

Figure 1 illustrates the architecture of ThinkTwice, which is composed of four basic components: 1) *segmentor* to split the given passage into shorter text fragments (Subsect. 3.1); 2) *retriever* to select the segmented fragments that are most relevant to the question (Subsect. 3.2); 3) *fusion* integrates the selected fragments according to the original order (Subsect. 3.3); 4) *reader* reads given question and fused text fragments to predict an answer (Subsect. 3.4).

3.1 Segmentor

The main challenge of LT-MRC is to exactly locate the most important information from a large number of knowledge fragments in the given passage, whose length usually exceeds the maximum length (e.g. 512 tokens) that the existing model can handle. To address this issue, we set a separator at the end of each paragraph and segment the passage, by which the lengths of most fragments are restricted to 60–80 tokens. Formally, the input passage P is segmented into shorter text fragments P_1, P_2, \dots, P_n .

3.2 Retriever

When looking for answers from large amounts of texts, a human tends to keep several slices that are most related to the current question in mind and filter

other trivial information. Inspired by such a human-like behavior, we come up with a Retriever module to select the most significant fragments which are most likely to answer the question.

We first pack the question Q with fragments $\{P_i\}_{i=1}^n$ into sequences $\{x_i\}_{i=1}^n$, where $x_i = [CLS]Q[SEP]P_i[SEP]$ ². Then, we utilize the pre-trained language model BERT [3] as Retriever's encoder ($BERT_{cls}$) to encode the input x_i into a sequence of contextual embeddings H_i :

$$H_i = BERT_{cls}(x_i). \quad (1)$$

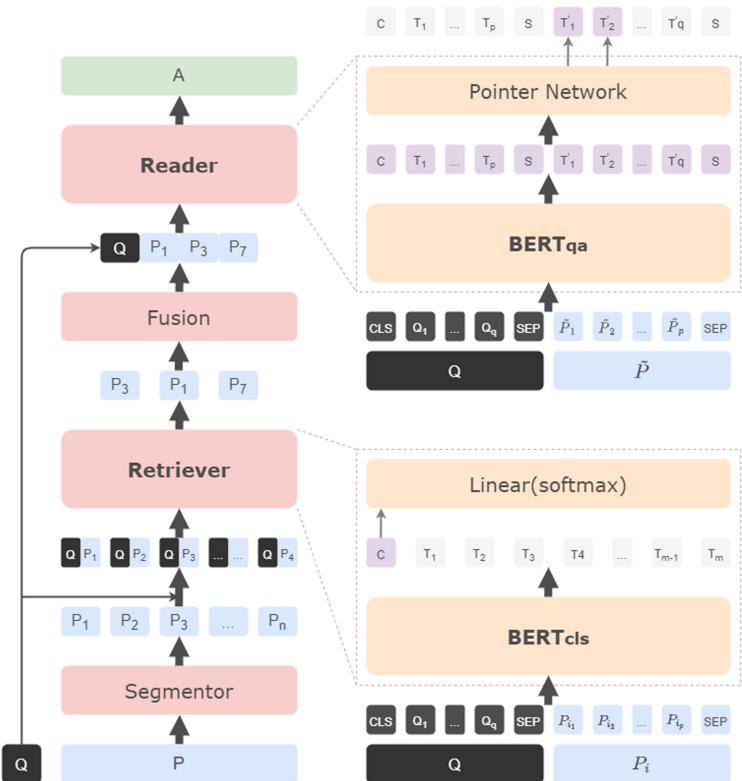


Fig. 1. Architecture of ThinkTwice.

The hidden vectors H_i^{cls} of $[CLS]$ represent the overall representation of the whole sequence. Further, a Linear Network and Softmax layer are employed to

² $[CLS]$ and $[SEP]$ are special tokens. The former can theoretically represent the overall information of the whole input sequence after being encoded, and the latter is used for input segmentation.

get the classification probability \hat{y}_c :

$$\hat{y}_c = \text{Softmax}(\text{Linear}(H_i^{cls})), \quad (2)$$

where \hat{y}_c indicates whether the current fragment P_i contains valid information that can answer the given question [27]. The “has/no answer” label $y_{c(i)}$ is set to 0 when P_i contains the annotated answer and the contrary to 1 at the training step.

The loss between ground truth y_c and predicted probability \hat{y}_c is calculated by the cross-entropy loss function:

$$\mathcal{L}_{retriever} = -\frac{1}{n} \sum_{i=1}^n [y_{c(i)} \log \hat{y}_{c(i)} + (1 - y_{c(i)}) \log(1 - \hat{y}_{c(i)})]. \quad (3)$$

3.3 Fusion

We obtain the output logits $\{\hat{y}_{c(i)}\}_{i=1}^n$ by the Retriever, which represent the probabilities of fragments $\{P_i\}_{i=1}^n$ to answer the given question Q . Then, we select the top k fragments that are most relevant to Q through \hat{y}_c and discard the others. Moreover, the selected text fragments are merged into a single sequence according to the original order to ensure semantic coherence and contextual continuity. For example, if k is set to 3, and the top three fragments with higher scores are P_3 , P_1 , and P_7 . Then we concatenate them by their relative orders in the original passage and simultaneously truncate few of sequences which are massively long. Thus, the output of the Fusion module is $\tilde{P} = < P_1, P_3, P_7 >$, as shown in Fig. 1.

3.4 Reader

By the above modules, we get a shorter snippet \tilde{P} instead of the original passage P , to extract the answer span. In particular, we first pack the question Q and \tilde{P} into a single sequence $z = [\text{CLS}]Q[\text{SEP}]\tilde{P}[\text{SEP}]$. Then we utilize another BERT [3] as the Reader’s encoder ($BERT_{qa}$) to map the input z into a sequence of contextual hidden vectors. Next, a Pointer Network (PN) [20] is employed to decode the start/end position of the answer span from the question-aware passage representation:

$$\hat{y}_s, \hat{y}_e = PN(BERT_{qa}(z)), \quad (4)$$

where \hat{y}_s and \hat{y}_e denote the probabilities that the start and end position of the labelled answer decoded by the Reader, respectively.

During the training step, we use the cross-entropy loss function to calculate the loss at the start and end positions:

$$\mathcal{L}_{reader} = \frac{1}{2} \text{CrossEntropy}(\hat{y}_s, y_s) + \frac{1}{2} \text{CrossEntropy}(\hat{y}_e, y_e), \quad (5)$$

where y_s and y_e denote the labels of the start and end position of the answer, respectively. If the question cannot be answered by the current passage, both y_s and y_e are set to 0, which points to the [CLS] tokens.

During the prediction step, we first calculate the has-answer score $score_{has}$ and the no-answer score $score_{null}$ through \hat{y}_s and \hat{y}_e :

$$score_{has} = \max_{1 \leq i \leq j < L} (\hat{y}_s^{(i)} + \hat{y}_e^{(j)}), \quad (6)$$

$$score_{null} = \hat{y}_s^{(0)} + \hat{y}_e^{(0)}, \quad (7)$$

where i and j denote the answer position within the whole sequence length L and i is restricted to be less than j since the start position must be before the end position. Furthermore, $\hat{y}_s^{(0)}$ and $\hat{y}_e^{(0)}$ denote the no-answer logits since the first [CLS] doesn't represent any of tokens in \tilde{P} .

We obtain a distance score $score_{dist}$ by calculating the distance between $score_{has}$ and $score_{null}$ as the criteria of "has/no answer":

$$score_{dist} = score_{null} - score_{has}, \quad (8)$$

$$s, e = \operatorname{argmax}_{1 \leq i \leq j < L} (\hat{y}_s^{(i)} + \hat{y}_e^{(j)}), \quad (9)$$

Then we employ a threshold δ , where if $score_{dist}$ is less than δ , the Reader predicts s and e as the start and end positions of the answer, otherwise predicts it as an unanswerable question.

4 Experiments

4.1 Experimental Settings

We do experiments on NewsQA [18], which is a challenging long-text machine reading comprehension (LT-MRC) dataset including 13k CNN news articles with 120k human-generated question-answer pairs. After excluding 20k bad questions which are considered to make no sense by crowdsourcers, we make an analysis of the remaining dataset with 90k/5k/5k question-answer pairs for training/dev/test set. In detail, we count the concrete numbers of tokens per passage (TPP) and paragraphs per passage (PPP). The median values of TPP are 774/734/707 for train/dev/test set, while the maximums are 3.1k/2.3k/2.3k. Besides, the median values of PPP are 18/18/17 for train/dev/test set and the maximums are 87/63/54.

We compare the following existing LT-MRC models.

- Match-LSTM [21]. The model uses two uni-LSTMs to encode questions and passages.
- BiDAF [16]. The core layer of BiDAF is the attention flow layer, which calculates context-to-query attention and query-to-context attention, respectively.

- AMANDA [11]. It proposes an end-to-end question-focused multi-factor attention network for answer extraction. Multi-factor attentive encoding aggregates meaningful facts located in multiple sentences.
- DecaProp [17]. The model integrates elaborate self-attention into RNN [14].
- Longformer [2]. The pre-trained language model using sparse attention matrix addresses the limitation of sequence length.
- CogLTX [4]. The proposed framework identifies key sentences by judging the relevance among various sentences.

We measure LT-MRC performance with two official metrics: Exact Match (EM), which measures the percentage of predictions that match exactly the ground truths; and F1 measures the average overlap between the prediction and ground truth at the token level.

To verify the effectiveness of our two-stage reading strategy, we employ the following available pre-trained language models and process the LT-MRC task by dividing the article into paragraphs with a sliding window mechanism, to build baselines: BERT [3], RoBERTa [13], ALBERT [12], and SpanBERT [7]. The implementations of these models are based on the public Pytorch implementation from Transformers [19]. In the training step, we set the initial learning rate to be 2e-5 in base models and 2e-6 in large models with a warm-up proportion of 0.1, and L2 weight decay of 0.01. The batch size is selected with 8 in base models and 1 in large models. The number of the epoch is set to be 1 in base models and 2 in large models in Retriever and 3 in Reader for all experiments. Texts are tokenized using wordpieces [23], with a maximum length of 256 in the first stage and 512 in the second stage. We perform several experiments in the first stage to select the k (a hyperparameter in the presentation of top k best paragraphs), and set $k = 5$.

4.2 Results

Comparison with the existing models. We compare our model with the existing models as illustrated in Table 1. It shows that the proposed model outperforms the existing LT-MRC models on NewsQA with 1.4% and 5.8% improvement in F1 and EM, respectively. Moreover, the results also demonstrate that implementing the ThinkTwice strategy improves the performances of all pre-trained MRC models, with significant improvements (F1) on BERT-base (+3.2%) and RoBERTa-base (+4.5%), while the improvement on ALBERT-base is less remarkable (+0.6%). The reasons might be that 1) the sentence-order prediction (SOP) pre-trained task has solved the inter-sentence coherence, and 2) the cross-layer parameter sharing mechanism leads to little distinction for parameters even implemented new strategy. The considerable improvements over other models demonstrate that the ThinkTwice strategy accurately extracts the most significant paragraphs by Retriever, so that long texts are compressed into short ones with appropriate length, and most important information can be reserved.

In addition, to observe how different means of fusion modules contribute to the performance, we try to merge the selected text fragments in two alternative

ways on BERT-base. One way is to merge the fragments extracted in Retriever according to their relevance to the given question by the descending order, while the other way is according to the original order as described in Subsect. 3.3. As shown in the 8th and 9th rows in Table 1, we can see that our original fusion strategy performs significantly better than the descending order (+2.8%), despite the model with descending order also surpasses the baseline (+0.4%), which indicates that disturbing sequential order may lead to the loss of contextual information.

Table 1. The performances of LT-MRC models on NewsQA as well as our models' performances compared to their corresponding pre-trained language models.

Model	Dev		Test	
	F1	EM	F1	EM
Match-LSTM [21]	49.6	34.4	50.0	34.9
BiDAF [16]	-	-	52.3	37.1
AMANDA [11]	63.3	48.8	63.7	48.4
DecaProp [17]	65.7	52.5	66.3	53.1
Longformer-base [2]	68.1	58.3	68.1	58.1
CogLTX [4]	-	-	70.1	55.2
BERT-base [3]	65.6	56.3	65.4	55.2
+ ThinkTwice(descending order)	66.6	57.8	65.8	55.6
+ ThinkTwice(ours)	68.5	58.8	68.6	57.7
RoBERTa-base [13]	63.7	53.5	63.2	53.1
+ ThinkTwice(ours)	67.7	58.6	67.7	58.4
ALBERT-base [12]	68.1	58.2	68.0	58.0
+ ThinkTwice(ours)	68.7	59.1	68.6	58.8
SpanBERT-base [7]	67.7	57.1	67.5	56.2
+ ThinkTwice(ours)	69.9	59.8	69.7	59.4
BERT-large	68.9	59.2	68.8	58.6
+ ThinkTwice(ours)	70.1	59.5	69.8	59.4
SpanBERT-large	71.2	61.8	70.9	59.8
+ ThinkTwice(ours)	72.1	62.2	71.5	61.0

Paragraph Retrieval. Figure 2 shows the relationship between the Retriever and the ThinkTwice model. For the Retriever, the evaluation metric Hits@ k (the top k accuracy) measures that the selected paragraph containing the ground truth answer is included in the top k list returned by the retriever. For the ThinkTwice model, F1 evaluates the final performance of MRC models on NewsQA. According to the red curve, in line with the intuition, we can see that the larger the k , the higher the Hits@ k accuracy. Also, the performance of the Retriever

is over 90% when k is greater than 3. Moreover, we can see the ThinkTwice model (the green curve) achieves the highest performance (68.6) when $k = 5$. It indicates that the Retriever cannot recall enough candidate paragraphs if k is small, while more candidate paragraphs may lead to a larger search space if k is greater than 5. Thus we finally apply the hyperparameter $k = 5$ to all other experiments for ThinkTwice. Also, we compare the ThinkTwice model with the BERT-base MRC model (the blue curve). The results show that the ThinkTwice model performs better than BERT-base when k is from 2 to 9, which verifies the effectiveness of the two-stage strategy for ThinkTwice.

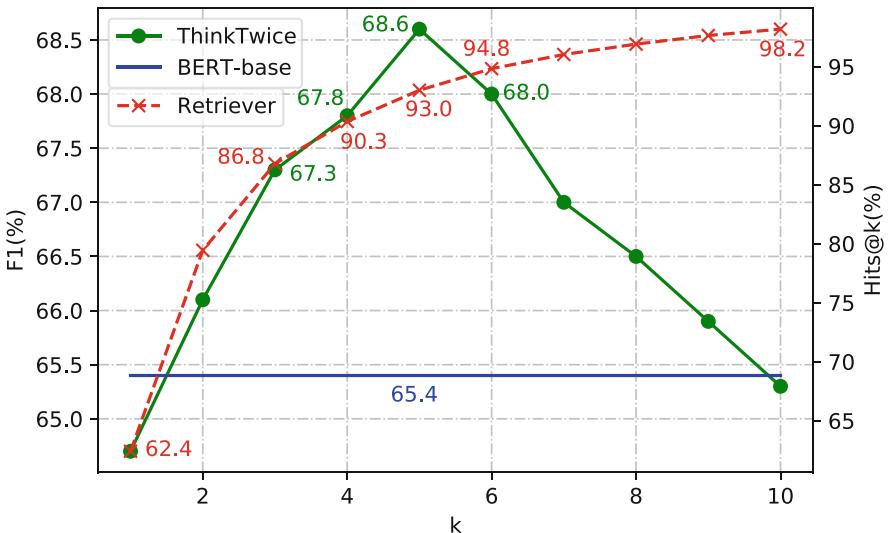


Fig. 2. Performance correlation between the Retriever and the ThinkTwice model (BERT-base) with different k . (Color figure online)

Effect of Text Length. To verify the effectiveness of ThinkTwice on the scenario of longer text, we compare it against BERT-base and Longformer-base MRC models over various text lengths on the test set. Note that the ThinkTwice model applies BERT-base as its reader. Figure 3 shows the result. We can see Longformer achieves the best performance on shorter documents ($[0, 512]$ and $(512, 1024]$). The reason is that Longformer-base inherits the pretrained weights from RoBERTa which performs very well on MRC tasks. However, on longer documents ($(1024, 1536]$ and $(1536, +\infty)$), the proposed ThinkTwice model is significantly better than the others. It proves that ThinkTwice is capable of locating the fragment that contains the answer more accurately. In addition, we can see BERT-base is better than Longformer-base on longer documents, which indicates that the sliding window mechanism (BERT-base) also has advantages compared with long inputs (Longformer-base). Finally, we can see that along

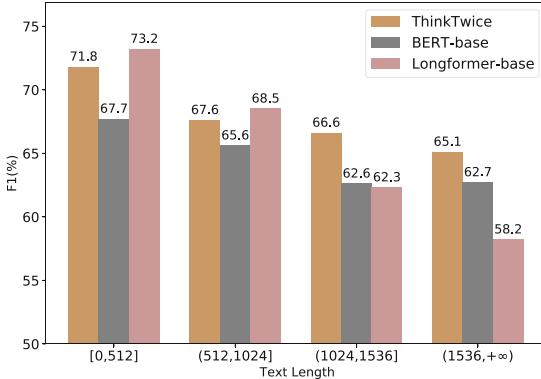


Fig. 3. Performances of ThinkTwice, BERT-base, and Longformer-base over different text lengths. The number of samples in the four length intervals are: 1,611, 2,074, 1,089, and 210.

Table 2. Two examples of three baselines' predictions and our prediction on NewsQA.

Example 1	
Passage	(CNN) – President Barack Obama spoke with Egypt's president moments after Hosni Mubarak addressed his country, telling the Egyptian that <i>he must make good on his promises</i> and avoid a violent response to the thousands of protesters in the streets. (...)
Token nums	2,036
Question	What did Obama say to Mubarak?
Answer	he must make good on his promises
Our pred	he must make good on his promises and avoid a violent response(✓)
BERT's	I just spoke to him after his speech(X)
ALBERT's	It is very important that people have mechanisms in order(X)
Longformer's	he must make good on his promises and avoid a violent response(✓)
Example 2	
Passage	(...) Tucked away in the verdant hills west of St. Andrews, Kingarrock Hickory Golf Course (greens fee, \$40 for nine holes and \$55 for 18) <i>is a nine-hole, 2,022-yard country estate course that is played exclusively with antiquated equipment</i> . (...)
Token nums	2,288
Question	What is Kingarrock Hickory?
Answer	is a nine-hole, 2,022-yard country estate course that is played exclusively with antiquated equipment
Our pred	a nine-hole, 2,022-yard country estate course that is played exclusively with antiquated equipment(✓)
BERT's	the kind of place that can change the way one thinks about golf(X)
ALBERT's	Golf Course(✓)
Longformer's	Top hotel penthouses(X)

with the increase of the document length, the performance of ThinkTwice is the most stable, especially when the length is greater than 512.

4.3 Case Study

We conduct a case study to further compare the predictions of ThinkTwice with other models on NewsQA. We discover that answers predicted by ThinkTwice are closer to ground truths.

To validate the performance over massively long texts, we pick two examples whose passage lengths are over 2,000 as illustrated in Table 2. In Example 1, the model is asked to tell what Obama says, ThinkTwice accurately locates the 1st paragraph containing the final answer and offers the proper content spoken by Obama, while BERT and ALBERT unexpectedly extract a sentence appeared in other paragraphs. In Example 2, our model locates the correct paragraph, while BERT and Longformer performs worse.

5 Conclusion

In this paper, we present a two-stage method on the LT-MRC task. The proposed ThinkTwice model addresses the issues of length limitation of pre-trained models via compressing long texts into shorter forms to precisely locate the position of the answer. The experimental results and analysis verify the effectiveness of our approach on the long texts. A latent drawback of the proposed model is that the short text compressed by Retriever might be incoherent due to the missing of antecedents. In the future, we will address this issue by leveraging coreference resolution or position embeddings.

References

- Atkinson, R.C., Shiffrin, R.M.: Human memory: a proposed system and its control processes. In: *Psychology of Learning and Motivation*, vol. 2, pp. 89–195. Elsevier (1968)
- Beltagy, I., Peters, M.E., Cohan, A.: Longformer: the long-document transformer. arXiv preprint [arXiv:2004.05150](https://arxiv.org/abs/2004.05150) (2020)
- Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) (2018)
- Ding, M., Zhou, C., Yang, H., Tang, J.: CogLTX: applying BERT to long texts. In: *Advances in Neural Information Processing Systems*, vol. 33 (2020)
- Ding, S., et al.: ERNIE-Doc: the retrospective long-document modeling transformer. arXiv preprint [arXiv:2012.15688](https://arxiv.org/abs/2012.15688) (2020)
- Hermann, K.M., et al.: Teaching machines to read and comprehend. arXiv preprint [arXiv:1506.03340](https://arxiv.org/abs/1506.03340) (2015)
- Joshi, M., Chen, D., Liu, Y., Weld, D.S., Zettlemoyer, L., Levy, O.: Spanbert: improving pre-training by representing and predicting spans. *Trans. Assoc. Comput. Linguist.* **8**, 64–77 (2020)
- Joshi, M., Choi, E., Weld, D.S., Zettlemoyer, L.: TriviaQA: a large scale distantly supervised challenge dataset for reading comprehension. arXiv preprint [arXiv:1705.03551](https://arxiv.org/abs/1705.03551) (2017)

9. Joshi, M., Levy, O., Weld, D.S., Zettlemoyer, L.: BERT for coreference resolution: baselines and analysis. arXiv preprint [arXiv:1908.09091](https://arxiv.org/abs/1908.09091) (2019)
10. Kryściński, W., Keskar, N.S., McCann, B., Xiong, C., Socher, R.: Neural text summarization: a critical evaluation. arXiv preprint [arXiv:1908.08960](https://arxiv.org/abs/1908.08960) (2019)
11. Kundu, S., Ng, H.T.: A question-focused multi-factor attention network for question answering. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
12. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: Albert: a lite BERT for self-supervised learning of language representations. arXiv preprint [arXiv:1909.11942](https://arxiv.org/abs/1909.11942) (2019)
13. Liu, Y., et al.: Roberta: a robustly optimized BERT pretraining approach. arXiv preprint [arXiv:1907.11692](https://arxiv.org/abs/1907.11692) (2019)
14. Mikolov, T., Karafiat, M., Burget, L., Černocky, J., Khudanpur, S.: Recurrent neural network based language model. In: Eleventh Annual Conference of the International Speech Communication Association (2010)
15. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100,000+ questions for machine comprehension of text. arXiv preprint [arXiv:1606.05250](https://arxiv.org/abs/1606.05250) (2016)
16. Seo, M., Kembhavi, A., Farhadi, A., Hajishirzi, H.: Bidirectional attention flow for machine comprehension. arXiv preprint [arXiv:1611.01603](https://arxiv.org/abs/1611.01603) (2016)
17. Tay, Y., Tuan, L.A., Hui, S.C., Su, J.: Densely connected attention propagation for reading comprehension. arXiv preprint [arXiv:1811.04210](https://arxiv.org/abs/1811.04210) (2018)
18. Trischler, A., et al.: NewsQA: a machine comprehension dataset. arXiv preprint [arXiv:1611.09830](https://arxiv.org/abs/1611.09830) (2016)
19. Vaswani, A., et al.: Attention is all you need. arXiv preprint [arXiv:1706.03762](https://arxiv.org/abs/1706.03762) (2017)
20. Vinyals, O., Fortunato, M., Jaitly, N.: Pointer networks. arXiv preprint [arXiv:1506.03134](https://arxiv.org/abs/1506.03134) (2015)
21. Wang, S., Jiang, J.: Learning natural language inference with LSTM. arXiv preprint [arXiv:1512.08849](https://arxiv.org/abs/1512.08849) (2015)
22. Wang, Z., Ng, P., Ma, X., Nallapati, R., Xiang, B.: Multi-passage BERT: a globally normalized BERT model for open-domain question answering. arXiv preprint [arXiv:1908.08167](https://arxiv.org/abs/1908.08167) (2019)
23. Wu, Y., et al.: Google’s neural machine translation system: bridging the gap between human and machine translation. arXiv preprint [arXiv:1609.08144](https://arxiv.org/abs/1609.08144) (2016)
24. Xie, Q., Dai, Z., Hovy, E., Luong, M.T., Le, Q.V.: Unsupervised data augmentation for consistency training. arXiv preprint [arXiv:1904.12848](https://arxiv.org/abs/1904.12848) (2019)
25. Yang, Z.: HotpotQA: a dataset for diverse, explainable multi-hop question answering. arXiv preprint [arXiv:1809.09600](https://arxiv.org/abs/1809.09600) (2018)
26. Zaheer, M., et al.: Big bird: transformers for longer sequences. arXiv preprint [arXiv:2007.14062](https://arxiv.org/abs/2007.14062) (2020)
27. Zhang, Z., Yang, J., Zhao, H.: Retrospective reader for machine reading comprehension. arXiv preprint [arXiv:2001.09694](https://arxiv.org/abs/2001.09694) (2020)



EviDR: Evidence-Emphasized Discrete Reasoning for Reasoning Machine Reading Comprehension

Yongwei Zhou¹, Junwei Bao², Haipeng Sun², Jiahui Liang², Youzheng Wu²,
Xiaodong He², Bowen Zhou², and Tiejun Zhao¹(✉)

¹ Harbin Institute of Technology, Harbin, China

ywzhou@hit-mtlab.net, tzhao@hit.edu.cn

² JD AI Research, Beijing, China

{baojunwei,sunhaipeng6,liangjiahui14,wuyouzheng1,xiaodong.he,
bowen.zhou}@jd.com

Abstract. Reasoning machine reading comprehension (R-MRC) aims to answer complex questions that require discrete reasoning based on text. To support discrete reasoning, evidence, typically the concise textual fragments that describe question-related facts, including topic entities and attribute values, are crucial clues from question to answer. However, previous end-to-end methods that achieve state-of-the-art performance rarely solve the problem by paying enough emphasis on the modeling of evidence, missing the opportunity to further improve the model’s reasoning ability for R-MRC. To alleviate the above issue, in this paper, we propose an Evidence-emphasized Discrete Reasoning approach (EviDR), in which sentence and clause level evidence is first detected based on distant supervision, and then used to drive a reasoning module implemented with a relational heterogeneous graph convolutional network to derive answers. Extensive experiments are conducted on DROP (discrete reasoning over paragraphs) dataset, and the results demonstrate the effectiveness of our proposed approach. In addition, qualitative analysis verifies the capability of the proposed evidence-emphasized discrete reasoning for R-MRC (Code is released at <https://github.com/JD-AI-Research-NLP/EviDR>).

Keywords: Evidence · Discrete reasoning · Machine reading comprehension

1 Introduction

Machine Reading Comprehension (MRC) aims to answer questions based on text, which has recently been widely explored and achieved remarkable progress that some methods have approached and even outperformed humans [6, 9, 15, 16]. Most of these studies focus on MRC datasets which have been released around span extraction, e.g., SQuAD [19], conversational state tracking, e.g., CoQA [21]

T. Zhao—Work done during the first author’s internship at JD AI Research.

and QuAC [4], passage retrieval, e.g., HotpotQA [24], multi-choice selection, e.g., RACE [14] and CommonsenseQA [23], and answer generation, e.g., MS-MARCO [18] and DuReader [11]. However, the reasoning capability of MRC models, which is especially crucial for the comprehension of sports news, scientific reports, and financial news where much arithmetic computation is required, has rarely been evaluated on these datasets.

Table 1. An Example of question-answer pairs along with document from the DROP dataset. Sentence level evidences are in blue, and clause level evidences are further in bold. **Ques** means question and **Ans** indicates answer.

Text: As of the census of 2010, there were 31,894 people, 13,324 households, and 8,201 families residing in the city. The population density was 1,851.1 inhabitants per square mile ($714.7/\text{km}^2$). There were 14,057 housing units at an average density of 815.8 per square mile ($315.0/\text{km}^2$). **The racial makeup of the city was 93.9% White (U.S. Census), 0.3% African American (U.S. Census), 1.7% Native American (U.S. Census), 0.8% Asian (U.S. Census), 0.1% Race (U.S. Census), 0.7% from Race (U.S. Census), and 2.4% from two or more races.** Hispanic (U.S. Census) or Latino (U.S. Census) of any race were 2.8% of the population.

Ques: How many more percentage of the population had a racial make-up of White than Asian?

Ans: 93.1

To complement the above shortage of research on reasoning machine reading comprehension (R-MRC), DROP [7] has recently been proposed for tracking complex questions that require discrete reasoning over text. Table 1 shows an example of DROP dataset, where the ground truth answer 93.1 to the question should be derived by arithmetic computation $93.9 - 0.8 = 93.1$, especially referring to the evidences, textual fragments in blue, including topic entities, i.e., White and Asian, and attribute values, i.e., 93.9% and 0.8%. As a preliminary attempt toward the task, NAQANET [7] is proposed as a number-aware framework to deal with questions with multi-predictors corresponding to different answer types, including span, count, and addition/subtraction. Based on NAQANET, NumNet [20] and QDGAT [2] perform reasoning over a heuristic graph including numerical values or additional entities with graph neural network to enhance the reasoning abilities. Although these end-to-end methods achieve state-of-the-art performance, they have rarely placed enough emphasis on explicitly modeling of evidence that is typically crucial clues from question to answer, which miss the opportunity to further improve reasoning ability for R-MRC.

In this paper, we propose to address the R-MRC problem with Evidence-emphasized Discrete Reasoning (EviDR). First, evidence is pinpointed with an evidence detector finetuned on a pre-trained language model via distant supervision. In detail, the evidence detector is trained to judge whether a textual fragment is evidence or not, where the distant supervision signal is obtained under

one-shot heuristic rules without human annotation. We adopt multi-granularity evidence, including sentence-level and clause level, as a trade-off between recall and precision for evidence detection. Then, information about evidence, including evidence representations and evidence pinpointing distribution over text, are used to drive a reasoning module to derive answers. Specifically, the reasoning module is implemented with a relational heterogeneous graph convolutional network (RH-GCN) upon the same encoder to explicitly propagate and emphasize the information of evidence. The heterogeneous graph is constructed based on sentence-level and clause-level nodes linked with different edges and updated with evidence pinpointing distribution as weights. In general, our model is jointly trained with multi-tasks, including evidence detection and reading comprehension.

Experiment results on the DROP dataset show that our approach achieves significant improvements compared with a strong baseline built upon a pre-trained language model without evidence modeling, and similar and even better results compared to the state-of-the-art model, i.e., QDGAT. Besides, the ablation study verifies the effectiveness of the distant supervision of evidence detection and the proposed evidence-emphasized discrete reasoning module with RH-GCN. Moreover, qualitative analysis verifies the reasoning ability of our proposed approach for R-MRC.

The contributions of the paper include the following three aspects. (1) We propose an evidence detector to explicitly pinpoint multi-granularity evidence as clues, which is learned via distant supervision. (2) We propose an evidence-emphasized discrete reasoning network with a relational heterogeneous graph convolutional network, which enhances the reasoning ability of R-MRC models. (3) We conduct extensive experiments and analysis on DROP, proving the effectiveness of the approach, and verifying the capability of the evidence-emphasized discrete reasoning for R-MRC.

2 Related Work

Recently, two lines of approaches have been proposed for the R-MRC task. The first is based on semantic parsing. Dua *et al.* [7] converted the unstructured text into a table and adopted a grammar-constrained semantic parsing model named KDG to answer the question over the table [13]. Chen *et al.* [3] proposed a generative model NeRd, which is composed of a reader and a programmer. They are responsible for encoding questions and passages into vector representation and generate grammatical programs, respectively. Gupta *et al.* [10] learned to parse compositional questions as executable programs where each atomic program is a learnable neural module. However, the model only adapted to questions with predefined templates. The second is based on neural end-to-end methods. As a preliminary attempt toward the task, Dua *et al.* [7] proposed a number-aware framework named NAQANET to produce three different answer types with various predictors, including a span, count, and arithmetic expression. To aggregate relative magnitude relation between two numbers, NumNet [20] was proposed to perform multi-step reasoning over a number comparison graph. Chen *et al.* [2]

proposed QDGAT based on a heterogeneous graph, which aggregates both entity and number nodes information. GenBERT [8] was proposed to inject the discrete reasoning abilities into BERT by generating numerical data. Compared to these existing methods, our proposed method focuses on placing enough emphasis on the evidence modeling to enhance discrete reasoning ability for the R-MRC model.

3 Methodology

MRC aims to predict an answer A with the maximum probability P according to the given question Q and passage text P :

$$A = \arg \max_{A \in \Omega} P(\hat{A}|P, Q). \quad (1)$$

Compared to traditional span extraction MRC, in the R-MRC task, the answer A can not only be spans (single or multiple spans) from the question or passage but also a number obtained by arithmetic computations with some numbers in the context. For questions with a span answer, discrete reasoning is also required in the R-MRC task, such as sort and comparison. In this paper, we explicitly model evidence over text to enhance the reasoning ability of R-MRC systems.

The framework of our proposed model **EviDR** is briefly described in Fig. 1, which is mainly composed of four components, i.e., an encoder, an evidence detector, an evidence-emphasized reasoning module, and a prediction module. The

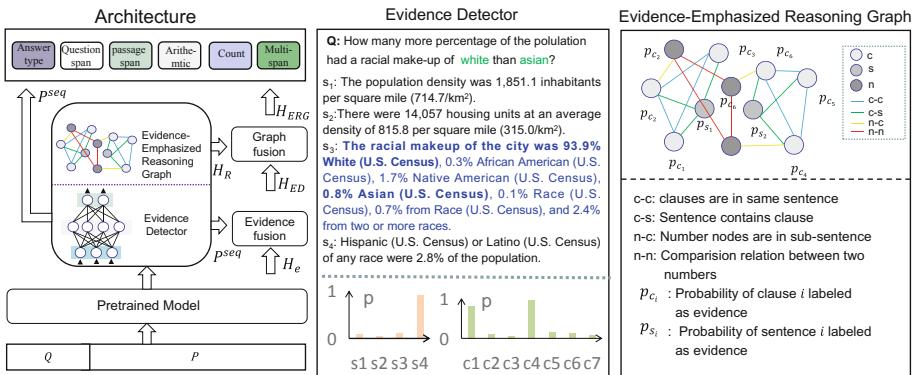


Fig. 1. An illustration of EviDR architecture. It consists of an encoder, an evidence detector, an evidence-emphasized reasoning graph, and a prediction module. Multi-granularity evidence, including sentence-level and clause-level, is pinpointed through the evidence detection. The evidence-emphasized reasoning module performs multi-step reasoning over a heterogeneous graph in which nodes are weighted by evidence, including sentence nodes, clause nodes, and number nodes. The prediction module supports five kinds of answer types, i.e., question span, passage span, arithmetic expression, count, and multi-spans.

encoder is responsible for semantic comprehension for the context of question and passage, commonly implemented by a pre-trained language model at present. Upon the encoder, an evidence detector learns to determine whether the current fragment supports answer prediction. We construct an evidence-emphasized reasoning graph based on the detector result and integrate knowledge from pieces of evidence through a graph convolution network. Finally, we leverage evidence to guide the answer prediction.

3.1 Encoding Module

Without loss of generality, we employ pre-trained language model (PLM) as the backbone architecture to encode context of question and passage, which takes concatenation of [CLS], tokens in question, [SEP], tokens in passage and [SEP] as textual inputs, and the output representation is denoted as:

$$\mathbf{H}_e = [\mathbf{H}_e^Q; \mathbf{H}_e^P] = \text{PLM}(\mathbf{Q}, \mathbf{P}). \quad (2)$$

where $\mathbf{H}_e^Q \in \mathbb{R}^{L_q \times d_h}$ and $\mathbf{H}_e^P \in \mathbb{R}^{L_p \times d_h}$ are the output representations of question and passage, respectively. L_q and L_p are the length of question and passage tokens, respectively. d_h is the hidden size.

3.2 Evidence Detector

Evidence Detector. Exactly as supporting fact prediction task in HotpotQA [24] and documents retrieval procedure in open domain QA task, evidence plays a crucial role in the MRC task. Therefore, we additionally introduce an evidence detector, which is responsible for discriminating whether each fragment can act as evidence to support answer prediction or not.

Specifically, the evidence detector takes the representation of question \mathbf{S}^Q and each fragment in passage \mathbf{S}_k^P as input features ($k = 1, 2, \dots, m$) and output the probability distribution of being identified as evidence to support answer prediction through a feed-forward network $\text{FFN}(\cdot)$ as follows:

$$\mathbf{P}_k = \text{FFN}(\mathbf{S}^Q, \mathbf{S}_k^P), \quad \mathbf{S}^Q = \beta^Q \mathbf{H}_e^Q, \quad \beta^Q = \text{softmax}(\mathbf{H}_e^Q \mathbf{W}), \quad (3)$$

where $\mathbf{W} \in \mathbb{R}^{d_h \times d_h}$ is a learnable parameter matrix. The hidden state of k^{th} fragment in passage \mathbf{S}_k^P can be derived as similar as S^Q . In this paper, we consider sentence-level and clause-level as evidence fragment to support answer prediction.

Evidence Fusion. To leverage the detected evidence, an evidence fusion layer is employed to integrate the evidence information into the hidden state of the input token sequence softly via layer normalization $\text{LN}(\cdot)$ as follows:

$$\begin{aligned} \mathbf{H}_{ED} &= \text{LN}(\mathbf{H}_e + \mathbf{P}^{seq} \odot \mathbf{H}_e), \\ \mathbf{P}_i^{seq} &= \mathbf{P}_k \text{ if token } i \text{ in } k^{th} \text{ evidence fragment,} \end{aligned} \quad (4)$$

where the token-level evidence probability \mathbf{P}^{seq} is denoted according to the (sentence and clause-level) evidence fragment probability distribution \mathbf{P}_k ($k = 1, 2, \dots, m$).

3.3 Evidence-Emphasized Reasoning Graph

Construction of Reasoning Graph We illustrate the details about how to build the evidence-emphasized reasoning graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ in this section. NumNet [20] builds a directed graph with all numbers as nodes, where the direction of the edges reflects the value relationship between numbers. In contrast to NumNet, besides the number nodes \mathbf{V}_N , we additionally introduce sentence evidence nodes \mathbf{V}_S and clause evidence nodes \mathbf{V}_C . i.e. $\mathbf{V} = \mathbf{V}_N \cup \mathbf{V}_S \cup \mathbf{V}_C$. The edges E indicate all the relationships in the heterogeneous graph following the three situations.

- *The edge between two number nodes:* Similar to NumNet [20], we also model the comparison relationship among numbers. An edge $e_{i,j}$ exists between any two number nodes v_i and v_j , the direction of edge $e_{i,j}$ reflects the comparison relation.
- *The edge between clause evidence nodes:* Building an edge $e_{i,j}$ between any two clause evidence nodes e_i and e_j , if they belong to the same sentence.
- *The edge between clause evidence nodes and sentence evidence nodes:* Building an edge $e_{i,j}$ between a clause evidence node e_i and a sentence evidence node e_j , if node e_i belongs to sentence node e_j .
- *The edge between number nodes and clause evidence nodes:* An edge $e_{i,j}$ exists between a number node v_i and a clause evidence node v_j when v_i is in text of v_j .

Evidence-Emphasized Reasoning over Graph. We leverage relational heterogeneous graph convolutional network (RH-GCN) to perform reasoning over the constructed evidence graph \mathcal{G} and illustrate the details of the reasoning process as follows:

Initialization. For each number node $v_i \in \mathbf{V}_N$, its representation is initialized as the corresponding token vector of \mathbf{H}_{ED} , i.e. $\mathbf{v}_i = \mathbf{H}_{ED}[I(v_i)]$ where $I(v_i)$ denotes the token index corresponding to node v_i . For nodes $v_i \in \mathbf{V}_S \cup \mathbf{V}_C$, the initial representation of v_i can be derived by the weighted sum of all the corresponding tokens' representation, that is, $\mathbf{v}_i = \sum_k \alpha_k \mathbf{H}_{ED}[I(v_{ik})]$.

Evidence-Emphasized Information Propagation. To leverage the evidence to guide the reasoning over the graph, each node is assigned a weight with the probability p_j . We leverage relation-specific transform matrices in the message propagation to distinguish different relations among nodes. The message propagation procedure is defined as,

$$\hat{\mathbf{v}}_i = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} p_j \mathbf{W}^{t_{ji}} \mathbf{v}_j, \quad (5)$$

where $\hat{\mathbf{v}}_i$ is denoted as the propagated message representation of node v_i from its all neighbors $v_j \in \mathcal{N}_i$. t_{ji} is the relations between node v_i and v_j . $\mathbf{W}^{t_{ji}}$ is the transform matrices assigned to relation t_{ji} .

Updating of Node Representation. Formally, We update the node representation by fusing the propagated message representation of node v_i obtained in last step with the information of the node as follows:

$$\mathbf{v}_i = \text{ReLU}(\mathbf{W}_v \mathbf{v}_i + \hat{\mathbf{v}}_i), \quad (6)$$

where \mathbf{W}_v is a learnable parameter matrix.

Fusion layer. Following NumNet [20], we integrate the structured knowledge implied in the evidence graph into the representation of the sequence as

$$\begin{aligned} \mathbf{H}_{ERG} &= [\mathbf{H}_{ERG}^Q; \mathbf{H}_{ERG}^P] = \text{ResiGRU}(\text{LN}(\mathbf{H}_{ED} + \mathbf{H}_R)), \\ \mathbf{H}_R[j] &= \mathbf{v}_i \text{ if } j^{\text{th}} \text{ token in node } v_i, \end{aligned} \quad (7)$$

where $\text{ResiGRU}(\cdot)$ means the composite function with residual function and a GRU layer.

3.4 Prediction Module

In this section, we demonstrate the details of the answer prediction module, including five answer predictors corresponding to different answer types and an answer type predictor.

Answer Type. The answer type predictor calculates the probability distribution of different answer type choices as

$$\mathbf{P}^{type} = \text{softmax}(\text{FFN}(\mathbf{M}^{type})), \quad \mathbf{M}^{type} = [\mathbf{h}^Q; \mathbf{h}^P], \quad (8)$$

where \mathbf{h}^Q and \mathbf{h}^P is calculated by weighted sum with token-level representation of question and passage $\mathbf{H}_{ERG}^Q \in \mathbb{R}^{L_q \times d_h}$ and $\mathbf{H}_{ERG}^P \in \mathbb{R}^{L_p \times d_h}$.

Single Span. Following Hu *et al.* [12], we use a question-aware decoding strategy to predict the start and end index. Specifically, the question representation vector, which means the summary of the question sequence information, is first computed as:

$$\alpha^Q = \text{softmax}(\text{FFN}(\mathbf{H}_{ERG}^Q)), \quad \mathbf{g}^Q = \alpha^Q \mathbf{H}_{ERG}^Q. \quad (9)$$

We leverage the pinpointed evidence to direct the prediction of the start index and end index as the following formulas:

$$\begin{aligned} \mathbf{P}_{start}^P, \mathbf{P}_{end}^P &= \text{masked_softmax}(\mathbf{P}^{seq} \odot \text{FFN}(\mathbf{M})), \\ \mathbf{M} &= [\mathbf{H}_{ERG}; \mathbf{H}_{ERG} \odot \mathbf{g}^Q], \end{aligned} \quad (10)$$

where $\text{masked_softmax}(\cdot)$ means $\text{softmax}(\cdot)$ can only be conducted among the element not be masked, which guarantees the answer span either belongs to question or document. \mathbf{P}^{seq} is the token-level evidence distribution derived in Sect. 3.2. When some fragment is identified as evidence by the evidence detector, we enhance the probability of a token in the fragment as the start and end index, and vice versa.

Arithmetic Expression. We yield the number representations $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N) \in \mathbb{R}^{N \times 2d_h}$ by gathering from \mathbf{H}_{ERG} , if N numbers exists. Similar to Dua *et al.* [7], we perform addition and subtraction over all numbers mentioned in the question and document context by assigning a sign (plus, minus or zero) to each number. In this way, arithmetic expression is converted into a sequence role labeling problem. The evidence information is leveraged to direct the prediction of number's sign as follows:

$$\begin{aligned}\mathbf{P}_i^{sign} &= \text{softmax}(\mathbf{P}_i^e \odot \text{FFN}(\mathbf{M}_i^{sign})), \\ \mathbf{M}_i^{sign} &= [\mathbf{u}_i; \mathbf{h}^P; \mathbf{h}^Q], \quad \mathbf{P}_i^e = (\mathbf{P}_i^{seq}, \mathbf{P}_i^{seq}, 1 - \mathbf{P}_i^{seq}),\end{aligned}\tag{11}$$

where $\mathbf{P}_i^e \in \mathbb{R}^3$ is a constructed weight vector for prediction of i^{th} number's sign. P_i^{seq} means the probability of the token as evidence at the position of the i^{th} number. We increase the probability that the sign of a number is discriminative as plus and minus when the segment containing the number is identified as evidence.

Count. We model count question as a 10-class classification problem (0–9). We first compute the input feature vector, integrating all the mentioned numbers, question, and passage information as follows:

$$\begin{aligned}\mathbf{P}^{count} &= \text{softmax}(\text{FFN}(\mathbf{M}^{count})), \quad \mathbf{M}^{count} = [\mathbf{h}^U; \mathbf{h}^P; \mathbf{h}^Q], \\ \mathbf{h}^U &= \alpha^U \mathbf{U}, \quad \alpha^U = \text{softmax}(\mathbf{U} \mathbf{W}^U).\end{aligned}\tag{12}$$

Multi-Spans. For multi-spans extraction, the probabilities are derived with a sequence role labeling method $\text{SRL}(\cdot)$ as the same as Segal *et al.* [22]:

$$\mathbf{P}^{MS} = \text{SRL}(\mathbf{H}_{ERG}),\tag{13}$$

where $\mathbf{P}^{MS} \in \mathbb{R}^{L \times 3}$ are probability distribution of token's BIO tagging.

4 Training with Distant Supervision

For the R-MRC task, we expect the machine can automatically learn to establish the bridge between question and answer. However, it is often expensive to obtain some intermediate signal annotations that explain the reasoning process for question answering according to the given question and passage. In this paper, we propose a few one-shot heuristic rules for evidence detection without human annotations as follows, and we utilize them as distant supervision signals to train the evidence detector.

- For an instance with span answer, we identify all the fragments (sentence-level and clause-level) that contain answer text as evidence. Moreover, the fragments are also marked as evidence if containing the topic entities in question.

- For an instance with arithmetic expression answer, we first find all the expressions that can obtain the answer by conducting addition or subtraction over up to 3 numbers in the context. We identify fragments containing those numbers that can be computed to obtain answers as evidence.

We jointly train the evidence detector and R-MRC model in form of multi-tasks. For evidence retrieval task, we compute the cross-entropy loss with predicted evidence label and the noised ground-truth evidence label as:

$$\mathcal{L}_{evi} = -\frac{1}{m} \sum_{k=0}^m y_k \log p_k + (1 - y_k) \log(1 - p_k), \quad (14)$$

where p_k is the probability of labeling k^{th} fragment as evidence. For R-MRC model part, the probability of the answer $P(A|P; Q; E)$ can be calculated as following:

$$P(A|P; Q; E) = \sum_{z \in \mathcal{T}} P_z(A|P; Q; E)P(z|P; Q; E), \quad (15)$$

where \mathcal{T} denotes all the answer types and E means evidence. To train our model, we adopt the marginal likelihood objective function [5], which sums over the probabilities of all possible annotations. The loss of answer prediction is denoted as \mathcal{L}_{ans} . Therefore the final loss is the weighted sum of two parts of losses with hyper-parameter λ as follows:

$$\mathcal{L} = \mathcal{L}_{ans} + \lambda \mathcal{L}_{evi}. \quad (16)$$

During inference, we first identify the evidence via the evidence detector and then determine the answer type and the corresponding answer with the prediction module.

5 Experiment

5.1 Dataset and Evaluation Metrics

We conduct experiments on an R-MRC benchmark named DROP [7] to evaluate our proposed model. DROP contains 77.4k/9.5k/9.6k instances for training, validation, and testing. DROP is composed of crowd-sourced question-answer pairs based on passages from Wikipedia. Specifically, for each question in DROP, various answer types such as date, number, or spans are involved. We take Exact Match (EM) and F1 as the evaluation metrics that are the same as previous work [2, 7, 20].

5.2 Experiment Settings

Our model is built upon the publicly available pre-trained model ELECTRA_{large} [6]. We use Adam optimizer with a cosine warmup mechanism to train the model. The maximum number of epochs and batch size is set to 12 and 16, respectively. For the parameters of ELECTRA_{large}, the learning rate and L_2 weight decay are 1.5e-5 and 0.01. For the other parts in EviDR, they are set to 5e-5 and 5e-4. The weight λ for loss of evidence detector is 0.2 and 0.4, which corresponds to sentence-level and clause-level evidence detection, respectively. The reasoning step over the heterogeneous graph is set to 3.

5.3 Baselines

We re-implemented NumNet+ [20]¹ and QDGAT [2]² as our baseline systems in this work. NumNet+ integrated relative magnitude between two numbers with a number graph and performed multi-step reasoning with graph convolution network. QDGAT [2] employed a question-directed graph attention network to reasoning over a heterogeneous graph that involved entity nodes and number nodes. In addition, an pre-train model based R-MRC system with multiple answer predictors was selected as another baseline, which was denoted as Baseline (RoBERTa [17]/ELECTRA [6]) in Table 2.

5.4 Main Results

Table 2 displays the performance of our model and other previous competitive models on DROP. Our method achieves 82.55 EM and 85.80 F1 scores on the test set, achieving similar and even better results compared to the state-of-the-art models i.e., NumNet+ and QDGAT. Moreover, EviDR achieves 0.93 EM and 0.92 F1 score improvement over the R-MRC system denoted as Baseline in Table 2. This demonstrates the effectiveness of evidence modeling.

Table 2. Results on the development and test sets of DROP dataset. * denotes our implementation results. Better results are in bold.

Method	Dev		Test	
	EM	F1	EM	F1
w/o Pre-trained Model				
NAQANet [7]	46.20	49.24	44.07	47.01
NumNet [20]	64.92	68.31	64.56	67.97
w/ Pre-trained Model				
GenBERT [8]	68.80	72.30	68.6	72.35
MTMSN [12]	76.68	80.54	75.88	79.99
NeRd [3]	78.55	81.85	78.33	81.71
ALBERT-Calculator [1]	80.22	83.98	79.85	83.56
NumNet+* [20]	80.74	84.09	80.92	84.33
QDGAT* [2]	82.03	85.01	82.31	85.65
Baseline (RoBERTa)	80.24	83.47	80.95	84.37
Baseline (ELECTRA)	81.16	84.22	81.63	84.98
EviDR	82.09	85.14	82.55	85.80
Human			94.90	96.42

¹ https://github.com/llamazing/numnet_plus.

² <https://github.com/emnlp2020qdgat/QDGAT>.

5.5 Ablation Study

Effectiveness of Evidence for Reasoning. To analyze the effectiveness of evidence for R-MRC, ablation studies are conducted on the development set of DROP. As shown in Table 3, we observe removing the evidence graph from our model, which leads to performance declines of 0.34 EM and 0.32 F1, which indicates the effectiveness of the evidence graph. On the other hand, taking off the direction of evidence for answer prediction in Eqs. 10 and 11 and the evidence fusion layer in Eq. 4, which leads to decline in 0.36 EM and 0.46 F1. It demonstrates that integrating the evidence information into the hidden state and answer predictor facilitates the answer prediction. Eventually, our method achieves 0.93 EM and 0.92 F1 points improvements over the baseline system.

Performance in Different Answer Type. Here, we evaluate the performance of our method on different answer types on the development of DROP. As reported in Table 3, when removing the evidence graph (w/o graph) and evidence direction for answer predictor (w/o ED), performance on the number, date, and span answer decline significantly. It further demonstrates the effectiveness of our methods on different answer types.

Table 3. Performance on different answer types on the development set of DROP. w/o ED means without direction of evidence for answer prediction in Eqs. 10 and 11 and the evidence fusion layer in Eq. 4. w/o Graph means removing the evidence-emphasized reasoning graph from EviDR. The Better results are in bold.

Method	Number	Date	Span	Overall
	EM/F1	EM/F1	EM/F1	EM/F1
EviDR	84.11/84.37	64.43/72.24	82.98/88.19	82.09/85.14
w/o ED	83.85/84.06	61.18/69.61	82.43/87.54	81.73/84.68
w/o Graph	83.86/84.10	53.64/61.79	82.65/88.12	81.75/84.82
Baseline	83.27/83.48	58.00/65.48	81.59/87.19	81.16/84.22

Performance of Evidence Detector. To analyze whether the evidence detector in our model can correctly recognize the evidence supporting answer prediction, we evaluate the performance of the evidence detector with Precision (**P**), Recall (**R**) and F1 as metrics. As reported in Table 4, for sentence-level evidence, we eventually achieve 92.57, 93.57, 90.92 on Precision, Recall, and F1 scores, respectively. And for clause-level evidence, they are 89.29, 90.08, and 86.67, respectively, which indicates the evidence detector can accurately recognize the evidence supporting answer prediction. In addition, we evaluate the heuristic rules with the metric, average keep ratio of sentence/clause-level evidence (**AKR**), i.e., the proportion

Table 4. The Evaluation on the evidence detection and heuristic rules on dev dataset.

Granularity	P	R	F1	AKR
Sentence	92.57	93.57	90.92	53.24
Clause	89.29	90.08	86.67	41.31

of labeled as evidence in all the fragments. As Table 4 shown, nearly 47% of sentences and 59% of clauses are filtered out. It significantly reduces the redundancy and noises of evidence while ensuring the answers are available from the evidence fragments.

5.6 Case Study

In Table 5, we give some examples to illustrate the effectiveness of our model compared to baseline systems. Sentence-level and clause-level evidence predicted by EviDR is in blue and bold, respectively. The first example shows the importance of evidence for questions with number answers. NumNet+ and QDGAT fail to capture the clause “*0.15% Native American (U.S. Census)*” is the crucial evidence for answer prediction. In contrast, our model accurately recognizes all the evidence pieces, which further facilitates the prediction of correct answers. The second example highlights the importance of evidence for questions with span answers. We observe that NumNet+ and QDGAT only find the related text “*the later category*” through the semantic matching ability. However, EviDR is capable to correctly capture what is “*the latter category*” by reasoning with the detected evidence “*outer provinces that were adjacent to the inner provinces and tributary states located on the border regions*” and “*destruction of vientiane belonged to the later category*”.

Table 5. The cases are from the development set of the DROP dataset. The predictions from the state of art models NumNet+ and QDGAT are illustrated. The last column indicates our answer prediction. Sentence-level and clause-level evidence predicted is in blue and bold, respectively.

Question-Answer	Passage	Prediction
Q: How many more people, in terms of percentage, made up the biggest racial group compared to the second smallest? A: 97.75	... The population density was 73 people per square mile (28/km ^{00b2}). There were 12,064 housing units at an average density of 36 per square mile (14/km ^{00b2}). The racial makeup of the county was 97.90% White (U.S. Census), 0.56% African American (U.S. Census), 0.15% Native American (U.S. Census), 0.28% Asian (U.S. Census), 0.02% Pacific Islander (U.S. Census), 0.36% from Race (United States Census), and 0.74% from two or more races. Hispanic (U.S. Census) or Latino (U.S. Census) of any race were 0.93% of the population. 21.3% were of English people, 16.5% Germans, 11.4% Irish people, 10.7% United States, 5.3% danish people and 5.3% Italian people ancestry according to Census 2000	NumNet+: +97.90=97.90 QDGAT: +97.90=97.90 EviDR: +97.90-0.15=97.75
Q: Southern Laos belonged to which category of territory? A: tributary states	Before the Monthon reforms initiated by king Chulalongkorn, Siamese territories were divided into three categories: Inner Provinces forming the core of the kingdom, Outer Provinces that were adjacent to the inner provinces and tributary states located on the border regions. The area of southern Laos that came under Siamese control following the Lao rebellion and destruction of Vientiane belonged to the later category, maintaining relative autonomy. Lao nobles who had received the approval of the Siamese king exercised authority on the Lao population as well as the Alak and Laven-speaking tribesmen. Larger tribal groups often raided weaker tribes abducting people and selling them into slavery at the trading hub of Champasak, ...	NumNet+: the later category QDGAT: the later category EviDR: tributary states

6 Conclusion and Future Work

In this work, we propose an evidence-emphasized discrete reasoning framework named EviDR for the R-MRC task. Specifically, we explicitly model evidence and introduce an evidence detector to recognize evidence to support answer prediction. In addition, we leverage an evidence-emphasized reasoning graph module to enhance the reasoning ability of EviDR. Experiments show that EviDR achieves remarkable performance.

Acknowledgments. The work is supported by the National Natural Science Foundation of China (No. U1908216) and the National Key R&D Program of China under Grant No. 2020AAA0108600.

References

1. Andor, D., He, L., Lee, K., Pitler, E.: Giving BERT a calculator: Finding operations and arguments with reading comprehension. In: EMNLP-IJCNLP (2019)
2. Chen, K., et al.: Question directed graph attention network for numerical reasoning over text. In: EMNLP (2020)
3. Chen, X., Liang, C., Yu, A.W., Zhou, D., Song, D., Le, Q.V.: Neural symbolic reader: scalable integration of distributed and symbolic representations for reading comprehension. In: ICLR (2020)
4. Choi, E., et al.: QuAC: question answering in context. In: EMNLP (2018)
5. Clark, C., Gardner, M.: Simple and effective multi-paragraph reading comprehension. In: ACL (2018)
6. Clark, K., Luong, M., Le, Q.V., Manning, C.D.: ELECTRA: pre-training text encoders as discriminators rather than generators. In: ICLR (2020)
7. Dua, D., Wang, Y., Dasigi, P., Stanovsky, G., Singh, S., Gardner, M.: DROP: a reading comprehension benchmark requiring discrete reasoning over paragraphs. In: NAACL (2019)
8. Geva, M., Gupta, A., Berant, J.: Injecting numerical reasoning skills into language models. In: ACL (2020)
9. Glass, M., et al.: Span selection pre-training for question answering. In: ACL (2020)
10. Gupta, N., Lin, K., Roth, D., Singh, S., Gardner, M.: Neural module networks for reasoning over text. In: ICLR (2020)
11. He, W., et al.: DureaderDuReader: a Chinese machine reading comprehension dataset from real-world applications. In: MRQA@ACL (2018)
12. Hu, M., Peng, Y., Huang, Z., Li, D.: A multi-type multi-span network for reading comprehension that requires discrete reasoning. In: EMNLP-IJCNLP (2019)
13. Krishnamurthy, J., Dasigi, P., Gardner, M.: Neural semantic parsing with type constraints for semi-structured tables. In: EMNLP (2017)
14. Lai, G., Xie, Q., Liu, H., Yang, Y., Hovy, E.: RACE: Large-scale ReADING comprehension dataset from examinations. In: EMNLP (2017)
15. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: ALBERT: a lite BERT for self-supervised learning of language representations. In: ICLR (2020)
16. Li, S., et al.: Hopretriever: retrieve hops over wikipedia to answer complex questions. In: AAAI (2021)
17. Liu, Y., et al.: Roberta: a robustly optimized BERT pretraining approach. arXiv preprint [arXiv:1907.11692](https://arxiv.org/abs/1907.11692) (2019)

18. Nguyen, T., et al.: MS MARCO: a human generated machine reading comprehension dataset. In: CoCo@ NIPS (2016)
19. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: SQuAD: 100,000+ questions for machine comprehension of text. In: EMNLP (2016)
20. Ran, Q., Lin, Y., Li, P., Zhou, J., Liu, Z.: NumNet: machine reading comprehension with numerical reasoning. In: EMNLP-IJCNLP (2019)
21. Reddy, S., Chen, D., Manning, C.D.: CoQA: a conversational question answering challenge. In: TACL (2019)
22. Segal, E., Efrat, A., Shoham, M., Globerson, A., Berant, J.: A simple and effective model for answering multi-span questions. In: EMNLP (2020)
23. Talmor, A., Herzog, J., Lourie, N., Berant, J.: CommonsenseQA: a question answering challenge targeting commonsense knowledge. In: NAACL (2019)
24. Yang, Z., et al.: HotpotQA: a dataset for diverse, explainable multi-hop question answering. In: EMNLP (2018)

Dialogue Systems



Knowledge-Grounded Dialogue with Reward-Driven Knowledge Selection

Shilei Liu, Xiaofeng Zhao, Bochao Li, and Feiliang Ren^(✉)

School of Computer Science and Engineering, Northeastern University,
Shenyang 110169, China

{1901750, 1901840, 1901725}@stu.neu.edu.cn, renfeiliang@cse.neu.edu.cn

Abstract. Knowledge-grounded dialogue is a task of generating a fluent and informative response based on both conversation context and a collection of external knowledge, in which knowledge selection plays an important role and attracts more and more research interest. However, most existing models either select only one knowledge or use all knowledge for responses generation. The former may lose valuable information in discarded knowledge, while the latter may bring a lot of noise. At the same time, many approaches need to train the knowledge selector with knowledge labels that indicate ground-truth knowledge, but these labels are difficult to obtain and require a large number of manual annotations. Motivated by these issues, we propose Knoformer, a dialogue response generation model based on reinforcement learning, which can automatically select one or more related knowledge from the knowledge pool and does not need knowledge labels during training. Knoformer is evaluated on two knowledge-guided conversation datasets, and achieves state-of-the-art performance.

Keywords: Dialogue generation · Knowledge-grounded dialogue

1 Introduction

With the advances in sequence to sequence models, neural dialogue systems have attracted more and more research attention. Neural conversation response generation could be formulated as a Seq2Seq task: given a dialogue history, the model is asked to generate a high-quality response. A large number of end-to-end generative neural conversation models have been applied to open-domain conversation and chatbot, have achieved success in generating fluent responses. However, the usual Seq2Seq models tend to produce shorter and simpler responses, which are not informative [7, 8].

In order to generate informative and meaningful responses, a number of methods have been proposed by leveraging external knowledge. Besides the dialogue history, knowledge-based dialog system also combines several external knowledge (in this paper we only discuss unstructured textual knowledge) to construct an input sample. Generally, the collection of candidate knowledge is obtained

through rough text retrieval in the knowledge base with a certain amount of noise. The noisy knowledge will lead the conversation to meaningless themes [9], so it is essential for a model to identify and select the appropriate knowledge.

However, most existing methods [1, 3, 5, 8] can only select one knowledge with the highest confidence from the candidate knowledge to participate in dialogue generation, while abandoning other knowledge with low confidence but possibly containing useful information. In order to make full use of the external knowledge, some approaches like [9] try to use all the knowledge to participate in the response generation. Nevertheless, when the size of candidate knowledge set is too large, this method will bring serious computational overheads and noise knowledge will lead the conversation to irrelevant topics.

Besides, many existing methods [1, 5] require labels that indicate the ground-truth knowledge to train the knowledge-grounded dialog models. However, these knowledge labels are difficult to obtain and need to be constructed through a large number of manual annotations, which is labor-intensive.

Motivated by the above issues, we propose Knoformer, a novel knowledge-grounded dialogue response generation model. Firstly, we present a knowledge-aware dialogue module, which take the concatenation of dialogue history and selected knowledge as inputs and generation responses. We perform supervised learning to train this module. Secondly, we propose a knowledge selection module to select all appropriate knowledge according to hidden states from dialogue module. Due to the lack of knowledge labels, the selection module uses the feedback from the dialogue module as reward and uses the policy gradient algorithm for training. Besides, we also add a weak supervision loss to the selector to further boost the accuracy. The dialogue module and knowledge selector are optimized jointly in a recurrent way.

We conduct experiments on Wizard-of-Wikipedia [3] and Holl-E [11], and results shows that our model achieves the new state-of-the-art performance.

2 Related Work

Knowledge-based conversation have shown promising results in improving response informativeness [3, 15]. PostKS [8] uses prior and posterior distributions over knowledge to train a selector which can choice the most suitable candidate (and discard others) to participate in the response generation; KIC [9] uses recurrent knowledge interaction among response decoding steps incorporate appropriate knowledge and uses pointer-generate networks copy tokens from external knowledge according to knowledge attention distribution; [5] proposes Sequential Knowledge Transformer (SKT) to model knowledge selection history in multi-turn dialogue; PIPM/KDBTS [1] improves on PostKS and SKT by using posterior information prediction and knowledge distillation to bridge the gap between prior and posterior knowledge selection. PostKS, SKT and PIPM/KDBTS only select one knowledge but our Knoformer could select multiple knowledge from knowledge pool; To train SKT and PIPM/KDBTS, ground-truth knowledge labels are needed, but KnoFormer does not need to

specify ground-truth knowledge in advance; KIC uses a special structure to realize knowledge attention which has poor portability, and has no ability to filter out noise knowledge, while our model can directly use Transformer to realize knowledge attention and has the ability to select highly relevant knowledge.

One of the most related models to ours may be KnowledGPT [14], who also focus on the multiple knowledge selection issue in the knowledge-grounded dialogue by reinforcement learning. Our work is novel in that during training, KnowledGPT only calculates one reward for all knowledge selection actions, which is equivalent to only one action in an *episode*, while our model calculates rewards for each action, which can accurately punish or reward each action.

3 Methodology

We use capital letters for sequences (e.g., Y), lowercase letters for tokens in sequences (e.g., u), bold capital letters for matrices (e.g., \mathbf{K}), and bold lowercase letters for vectors (e.g., \mathbf{v}).

Given a conversation history $U = \{u_1, u_2, \dots, u_m\}$ and a set of external knowledge $\mathcal{K} = \{K_j\}_{j=1}^r$, where $K_j = \{k_1^j, k_2^j, \dots, k_n^j\}$ and r is the size of external knowledge set, the task of our proposed approach is to learn a knowledge selection module to select a subset of \mathcal{K} :

$$p(\mathcal{K}_{sub}) = \prod_{i=1}^o p(K_{a_i} | U, K_{a_1}, \dots, K_{a_{i-1}}) \quad \mathcal{K}_{sub} = \{K_{a_i}\}_{i=1}^o \quad 1 \leq a_i \leq r \quad (1)$$

where a_i represents the index of i -th knowledge in \mathcal{K}_{sub} . Specifically, we use a recurrent method to select knowledge one by one. The first knowledge is selected based on the dialogue history U , and then the second knowledge is selected based on U and the first knowledge K_{a_1} , and so on, to reach the maximum number of knowledge choices. \mathcal{K} usually has a lot of noise, so the filtered \mathcal{K} is more closely related to the conversation topic. In addition, we also learn a dialogue generation model generate response using U and \mathcal{K}_{sub} : $p(\hat{Y}) = f(U, \mathcal{K}_{sub})$. Compared with some models [1, 5, 8] that can only select one knowledge, our model uses a recurrent selection mechanism to pick multiple suitable external knowledge.

Our proposed model consists of three parts, a dialogue module which can fuse the features of knowledge and dialogue history to generate a response (Sect. 3.1), a knowledge encoding module which can encode all external knowledge separately (Sect. 3.2), and a knowledge selection module which can be trained without ground-truth knowledge based on reinforcement learning (Sect. 3.3). Section 3.4 introduces the training and inference of our model. The architecture of our proposed model is show in Fig. 1.

3.1 Dialogue Module

Our model operates in an iterative manner. Each time a knowledge is selected, the dialogue module will perform feature fusion on the currently selected

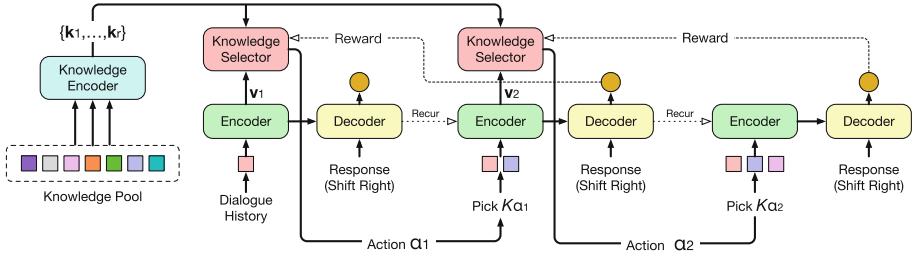


Fig. 1. The architecture of our proposed model (suppose the model chooses two knowledge). Modules of the same color share parameters. The orange circles represent the loss of the dialogue model, and they will be accumulated and averaged to form the final dialogue loss \mathcal{L}'_u .

knowledge. Our dialogue module is based on Transformer [12], which takes the dialogue history and several selected external knowledge as input, and generates a response token by token.

Assuming that $c - 1$ knowledge has been selected at present, then the input of the dialogue model is $X_c = [[\text{SOS}]; U; K_{a_1}; \dots; K_{a_{c-1}}; [\text{EOS}]]$, where **[SOS]** and **[EOS]** are special tokens. We pass X_c to Transformer encoder to obtain the context-aware representation \mathbf{X}_c :

$$\mathbf{X}_c = \text{TFEncoder}(e(X_c), \theta_e) \in \mathbb{R}^{m \times d} \quad (2)$$

where $e(\cdot)$ means to convert a sequence (or token) into word embedding vectors using a lookup table $\mathbf{M} \in \mathbb{R}^{vocab \times d}$ and θ_e represents the learnable parameters. After encoding, the information of dialogue history and external knowledge is fully integrated.

We take \mathbf{X}_c (as memory) and golden response Y (as input) into Transformer decoder to generate response token by token.

$$\mathbf{h}_t = \text{TFDecoder}(\mathbf{X}_c, e(Y_{1:t-1}), \theta_d) \in \mathbb{R}^d \quad p(y_t) = \frac{e(y_t)\mathbf{h}_t^\top}{\sum_{y'} e(y')\mathbf{h}_t^\top} \quad (3)$$

3.2 Knowledge Encoder

For i -th ($1 \leq i \leq r$) knowledge K_i in external knowledge set, we wrap it with two special characters **[SOS]** and **[EOS]**, and sent to a Transformer encoder:

$$\mathbf{K}_i = \text{TFEncoder}(e(K_i), \theta_k) \in \mathbb{R}^{n \times d} \quad (4)$$

We use the representation of **[SOS]** token (denoted as $\mathbf{k}_i \in \mathbb{R}^d$) as the overall representation of i -th knowledge.

3.3 Knowledge Selector

Assuming that we will choose the c -th knowledge in current step, we formulate the problem of learning-to-select under the framework of reinforcement learning. We define the **state** $s = \{K_{a_1}, \dots, K_{a_{c-1}}\}$ (omit some constants) of the model to be the knowledge that the selector has selected before the current step, e.g., $s = \{3, 2, 1, 4\}$. The **action** a is which knowledge to select. We define the **action space** of current step as the knowledge index in \mathcal{K} (i.e., $\{1, 2, \dots, r\}$). During implementation, we use mask technology to avoid choosing repeated action. Since \mathbf{X}_c integrates the features from dialogue history and selected knowledge, in order to propagate the information across different steps, we represent the state s with the representation of [SOS] token (denoted by $\mathbf{v}_c \in \mathbb{R}^d$) in \mathbf{X}_c .

The selection policy gives the probability $\pi(a | s)$ of taking an action a at the current state s , which is modeled by a bilinear matrix:

$$\pi(a | s) = \underset{a \in \mathcal{A}_c}{\text{softmax}}(\mathbf{v}_c \mathbf{W} \mathbf{k}_a^\top) \quad (5)$$

where $\mathbf{W} \in \mathbb{R}^{d \times d}$ is a learnable parameter and \mathbf{k}_a is the knowledge representation (Sect. 3.2). In order to achieve the early stop mechanism, a stop action can be added to action space in implementation.

3.4 Modules Integration

In the training phase of the recurrent selection mechanism, the actions of selecting next knowledge are sampled according to the probability given by the selection policy. Our model generates a sequence of knowledge for each dialogue history. We train the dialogue model with supervised learning, and we train the selector network via reinforcement learning and weekly supervised learning.

Supervised Learning for Response Generation. The ground-truth response has been given, so we train the conversation model via supervised learning. Consistent with most Seq2Seq models, the training objective is to minimize the negative log likelihood (NLL):

$$\mathcal{L}_u = -\frac{1}{z} \sum_{t=1}^z \log p(y_t) \quad (6)$$

Reinforcement Learning for Selection Policy. Based on the assumption that the ground-truth knowledge labels are not available, supervised learning cannot be used to train the knowledge selection model, it is natural to train it via reinforcement learning.

First of all, the accumulated reward for taking action a at state s is denoted as $R(s, a)$, which is derived in a recursive manner:

$$R(s, a) = e^{-\text{PPL}(s, a)/\gamma} + R(s', a') \quad (7)$$

where γ is a constant and we empirically set it to 10. $\text{PPL}_{(s,a)}$ denotes the perplexity of ground-truth response Y in current step and $R(s', a')$ denotes the next state-action pair. We use perplexity as reward to make the knowledge selector more inclined to select knowledge that can generate high-quality responses.

The selection policy network can be trained by maximizing the expected accumulated reward through the policy gradient algorithm [13]:

$$\mathcal{J} = \mathbb{E}_{a \sim \pi} [\tilde{R}(s, a)] \quad (8)$$

where $\tilde{R}(s, a) = R(s, a) - b$ and $b \approx \mathbb{E}[R(s, a)]$ is the baseline that is used to reduce the variance of gradient estimation [2]. To be consistent with the notations in response generation, we denote the loss function of selection policy as \mathcal{L}_k , which is the negative expected accumulated reward \mathcal{J} in Eq. 8: $\mathcal{L}_k = -\mathcal{J}$. Thus, the gradient of \mathcal{L}_k over a series of action-pair \mathcal{B} is given by:

$$\nabla \mathcal{L}_k = - \sum_{(s, a) \in \mathcal{B}} \nabla \log \pi(a | s) R(s, a) \quad (9)$$

Weakly Supervised Learning for Selector. Valuable knowledge usually has a higher text similarity with the ground-truth response, so we add an additional selection loss to the selector when selecting the first knowledge. We take the response as the query, use the TF-IDF algorithm to score the knowledge in \mathcal{K} , mark the index of knowledge with the highest score as a^+ , and the additional loss is defined as:

$$\mathcal{L}_s = -\log \pi(a^+ | s) \quad (10)$$

Training and Inference. The training procedure is show in Algorithm 1 (for ease of understanding, assume batch size is 1 and only optimize one step). The overall loss is defined as $\mathcal{L} = \mathcal{L}'_u/o + \mathcal{L}_k + \lambda \mathcal{L}_s$. Weakly supervised labels may not be correct, so we put a smaller weight λ on \mathcal{L}_s .

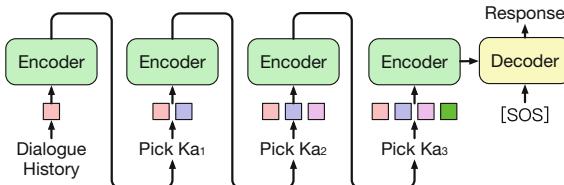


Fig. 2. Sketch of inference.

During inference, the knowledge selector take the best action (instead of sampling) at step c according to the selection policy:

$$a^* = \arg \max_{a \in \mathcal{A}_c} \pi(a | s) \quad (11)$$

Algorithm 1. Optimization Step

Input: $U, Y, \mathcal{K} = \{K_i\}_{i=1}^r$ and number of selections o ;
Output: Generation loss \mathcal{L}'_u , selection loss \mathcal{L}_k and \mathcal{L}_s ;

```

1: Encode knowledge independently according to Eq. 4 and get the knowledge representation  $\{\mathbf{k}_i\}_{i=1}^r$ ;
2: Initialize  $s = \phi$ ,  $\mathcal{B} = \phi$ ,  $\mathcal{L}'_u = 0$ ;
3: for  $c$  in  $\{1, \dots, o\}$  do
4:   Construct  $X_c$  from  $U$  and  $s$  according to Sect. 3.1;
5:   Encode  $X_c$  to obtain  $\mathbf{X}_c$  and  $\mathbf{v}_c$  according to Eq. 2;
6:   Decode and calculate  $\mathcal{L}_u$  according to Eq. 6;
7:    $\mathcal{L}'_u \leftarrow \mathcal{L}'_u + \mathcal{L}_u$ ;
8:   if  $c = 1$  then
9:     Calculate  $\mathcal{L}_s$  according to Eq. 10;
10:    end if
11:    if  $c < o$  then
12:      Calculate  $\pi$  according to Eq. 5;
13:      Sample an action  $a$  using Gumbel-Max Trick;
14:       $\mathcal{B} \leftarrow \mathcal{B} \cap \{(s, a)\}$ ,  $s \leftarrow s \cap \{a\}$ ;
15:    end if
16:  end for
17: Calculate  $\mathcal{L}_k$  according to Eq. 8;
18: return  $\mathcal{L}'_u$ ,  $\mathcal{L}_s$  and  $\mathcal{L}_k$ ;

```

After the action a^* is taken, a new knowledge is taken from the knowledge pool and appended to the input sequence, and so on, until reaching the upper bound of knowledge selection o . Because there is no need to calculate the reward during inference, in order to save overhead, the decoder only decodes when $c = o$. The sketch of inference is shown in Fig. 2.

4 Experiments

4.1 Datasets

We conduct experiments on Wizard-of-Wikipedia [3] and Holl-E [11]. Wizard-of-Wikipedia contains 18,430 training dialogues, 1,948 validation dialogues and 1,933 test dialogues on 1365 topics. And test set is split into two subsets according to topics, which are Test Seen with 965 dialogues and Test Unseen with 968 dialogues whose topics are never seen in training and validation set. There are about 61 sentences on average in the knowledge pool per turn, which are retrieved from Wikipedia based on the context. Holl-E contains 7,228 training dialogues, 930 validation dialogues and 913 test dialogues. Each dialogue is assigned with about 60 documents on average as the knowledge pool. Here, we use the modified version [5] which fits for knowledge selection.

It should be noted that in this paper, we focus on the scenarios where ground-truth knowledge is unknown. Thus, we do not use the ground-truth knowledge labels.

4.2 Models for Comparison

We compare our Knoformer with the following baselines:

- **KnowledGPT**: Another knowledge-based dialogue system based on BERT and GPT-2 which can select multiple external knowledge via reinforcement learning [14]. KnowledGPT does not provide complete source code and experimental results on other datasets, so we can only compare with it on the Wizard-of-Wikipedia dataset.
- **TF-IDF**: TF-IDF is a commonly used algorithm in information retrieval. We perform TF-IDF to sort all documents in knowledge set (dialogue context as query), and concatenate dialogue context and top-20 of them as the input.
- **PIPM/KDBTS**: A latent variable model that uses specially designed Posterior Information Prediction Module (PIPM) to select knowledge and Knowledge Distillation Based Training Strategy (KDBTS) to train the decoder with the knowledge selected from the prior distribution [1]. It is an improvement of PostKS and SKT, but still can only select one knowledge.

For fair comparison, we only use the knowledge selector of the above methods, and use BART [6] as the unified dialogue response generator. It should be noted that most pre-trained language models (e.g., BERT, GPT-2, BART) has a limit of max number of input tokens they can handle, so for BART+TF-IDF, text exceeding the maximum length will be truncated, and only keep the first 384 tokens. The source code of KIC and some other methods is not available, so we will not compare with them.

4.3 Implement Details

We implement our model over PyTorch framework. The parameters of dialogue module and knowledge encoder are initialized with BART-base and BERT-base respectively. We train our model using AdamW [10] optimizer with a batch size of 16 and learning rate 5e-5 at 3 epochs on a NVIDIA QUADRO RTX 8000 machine, and other hyperparameters are detailed in Table 1. When decoding, the number of beams is between 1 to 5. We adopt widely used public evaluation toolkit NLTK to evaluate the model performance.

Table 1. Hyperparameters over different datasets.

	Wizard seen	Wizard unseen	Holl-E
Max context length	384	384	256
Max knowledge length	64	64	48
Max response length	48	48	48
Max number of selection	6	3	2
λ	0.3	0.7	0.7

4.4 Automatic Evaluation

We adopt three automatic metrics include BLEU, Div, and corpus-level unigram F1 for evaluation. BLEU is well-known machine translation evaluation metric, and generated response with higher BLEU/F1 is closer to the ground-truth response and has preferable fluency. Div- n reflects the n -gram diversity of text [7], and response with higher Div- n could present more information.

Table 2. Automatic evaluation results on Wizard-of-Wikipedia test seen. BART in the first row means no knowledge is used.

Model	F1	BLEU-1	BLEU-2	BLEU-3	Div-1	Div-2
BART	24.0	22.4	10.2	5.0	6.0	23.8
BART+KnowledGPT	26.2	25.6	13.6	8.8	7.8	24.6
BART+TF-IDF	24.8	23.3	11.0	5.9	7.1	29.8
BART+PIPM	25.0	23.6	11.3	6.1	7.0	28.7
Knoformer	27.4	26.4	14.3	9.2	7.9	32.1

Table 3. Automatic evaluation results on Wizard-of-Wikipedia test unseen.

Model	F1	BLEU-1	BLEU-2	BLEU-3	Div-1	Div-2
BART	23.1	21.6	9.5	4.7	4.5	19.6
BART+KnowledGPT	24.7	24.6	12.6	7.8	4.9	23.6
BART+TF-IDF	23.5	22.3	10.0	5.4	5.1	21.4
BART+PIPM	24.0	23.0	10.5	5.6	4.7	20.8
Knoformer	25.4	24.8	12.6	8.0	5.1	23.1

Tables 2 and 3 shows the automatic evaluation results on Wizard, and Table 4 shows the results on Holl-E. We have the following observations: (1) Our Knoformer significantly surpasses all baselines in most evaluation metrics of all datasets, which means our knowledge selection module is more targeted and can select more valuable knowledge; (2) External knowledge is of great help to improve performance. If external knowledge is removed, performance will decline; (3) Only using prior experience (TF-IDF with context as query) to select knowledge is not effective.

4.5 Human Evaluation

Besides automatic evaluation, we also recruit three human annotators to do qualitative analysis on response quality. For each corpus, we randomly sample

Table 4. Automatic evaluation results on Holl-E.

Model	F1	BLEU-1	BLEU-2	BLEU-3	Div-1	Div-2
BART	24.7	20.6	11.5	5.2	4.0	16.1
BART+TF-IDF	27.8	25.0	11.0	5.9	7.1	28.3
BART+PIPM	36.3	33.4	27.5	24.6	4.6	20.8
Knoformer	39.5	37.9	31.0	28.4	7.0	25.8

200 samples, and each sample contains the dialog history, response, and external knowledge set. The annotators then judge the quality of the responses from three aspects, including context coherence, language fluency and response diversity, and assign a score in $\{0, 1, 2\}$ to each response for each aspect. Each response receives 3 scores per aspect, and the agreement among the annotators is measured via Fleiss' kappa [4]. The human evaluation result is shown in Table 5, and we observe that responses from our Knoformer are more fluent and more contextually coherent than those from baselines.

Table 5. Human evaluation results on Wizard-of-Wikipedia.

Models	Wizard test seen				Wizard test unseen			
	CC	LF	RD	Kappa	CC	LF	RD	Kappa
BART+KnowledGPT	1.78	1.80	1.64	0.61	1.72	1.74	1.66	0.62
BART+PIPM	1.73	1.75	1.65	0.59	1.69	1.70	1.62	0.61
Ours	1.80	1.84	1.69	0.60	1.74	1.77	1.66	0.61

4.6 Analysis

Ablation Study. In order to explore the importance of components in Knoformer, we conducted ablation experiments on Wizard-of-Wikipedia valid set, and the results are summarized in Table 6. First of all, we remove the loss item \mathcal{L}_k and \mathcal{L}_s of knowledge selection, making the selection of knowledge completely random. Results show that meaningless selection will harm the performance of the model. Secondly, we remove \mathcal{L}_s only, and performance is also degraded, which means that valuable knowledge does have a higher similarity with the ground-truth response, and building the link between knowledge and response directly helps improve model performance. Besides, we change the joint training to separate training, which means that selector and dialogue module are trained alternately. Results indicate that joint training can make better use of the feedback of dialogue model. In addition, when replacing our reward function with that of KnowledGPT (only return a reward for all actions in an episode), the performance drops significantly, which means that gives a reward to each action can effectively punish bad actions and enhance the knowledge selection.

Table 6. Results on wizard valid set.

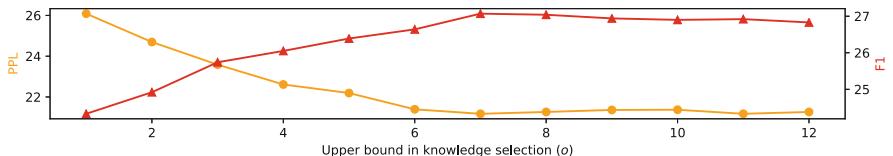
Models	Wizard test seen			Wizard test unseen		
	F1	BLEU-1	BLEU-3	F1	BLEU-1	BLEU-3
Ours	27.1	26.5	9.2	25.3	24.6	8.2
$-\mathcal{L}_k, \mathcal{L}_s$	24.6	23.1	5.9	23.6	22.1	5.3
$-\mathcal{L}_s$	26.8	26.3	8.5	24.9	24.1	7.6
-joint	26.9	26.3	9.1	25.1	24.3	8.0
R^\dagger	26.2	25.7	8.2	24.1	23.4	7.3

Case Study. Table 7 shows a case in Wizard test unseen. Our model select two pieces of knowledge and incorporate them into response. Compared with baselines and even reference, response generated by Knoformer is more informative.

Table 7. A case from test unseen of Wizard-of-Wikipedia.

Context	A: Elvis was such an amazing singer, but he also an incredible musician and actor B: Yes, The King. Did you know he sold more than 1 billion units in 20 years
Reference	I didn't know that! He was born in 1935 and died in 1977 though
Selected Knowl.	1. Elvis Aaron Presley (January 8, 1935 - August 16, 1977) was an American singer... 2. ... he is often referred to as the King of Rock and Roll or simply the King
(Ours)	Yes, he was king of rock and roll. He was born in 1935 and died in 1977
(BART+TF-IDF)	Yes, he sold over 100 million records worldwide. (BART+PIPM) I did not know that. I do know that he founded the Wall Street firm in 1960

Impact of o . To explore the influence of the number of knowledge choices on model performance, we vary the value of o in $\{1, 2, \dots, 12\}$ and report the evaluation results in Fig. 3. The smaller o , the smaller the probability that ground-truth knowledge will be captured. When o reaches a certain extent, the performance improvement is very weak or even slightly decreased, implying that the noise in the knowledge pool will interfere with the generation of responses.

**Fig. 3.** The performance of the model with different o in Wizard valid set (seen).

5 Conclusion

In this paper, we propose Knoformer, a knowledge-grounded dialogue generation model. Evaluation results on four benchmarks indicate that our model can significantly outperform state-of-the-art methods.

Acknowledgments. This work is supported by the National Key R&D Program of China (No. 2018YFC0830701), the National Natural Science Foundation of China (No. 61572120), the Fundamental Research Funds for the Central Universities (No. N181602013 and No. N171602003), Ten Thousand Talent Program (No. ZX202000035), and Liaoning Distinguished Professor (No. XLYC1902057).

References

- Chen, X., et al.: Bridging the gap between prior and posterior knowledge selection for knowledge-grounded dialogue generation. In: EMNLP (2020)
- Clark, K., Manning, C.D.: Deep reinforcement learning for mention-ranking coreference models. In: EMNLP (2016)
- Dinan, E., Roller, S., Shuster, K., Fan, A., Auli, M., Weston, J.: Wizard of Wikipedia: knowledge-powered conversational agents. In: ICLR (2019)
- Fleiss, J.: Measuring nominal scale agreement among many raters. *Psychol. Bull.* **76**, 378–382 (1971)
- Kim, B., Ahn, J., Kim, G.: Sequential latent knowledge selection for knowledge-grounded dialogue. In: ICLR (2020)
- Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: ACL (2020)
- Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B.: A diversity-promoting objective function for neural conversation models. In: NAACL (2016)
- Lian, R., Xie, M., Wang, F., Peng, J., Wu, H.: Learning to select knowledge for response generation in dialog systems. In: IJCAI (2019)
- Lin, X., Jian, W., He, J., Wang, T., Chu, W.: Generating informative conversational response using recurrent knowledge-interaction and knowledge-copy. In: ACL (2020)
- Loshchilov, I., Hutter, F.: Fixing weight decay regularization in adam. CoRR (2017)
- Moghe, N., Arora, S., Banerjee, S., Khapra, M.M.: Towards exploiting background knowledge for building conversation systems. In: EMNLP (2018)
- Vaswani, A., et al.: Attention is all you need. In: NeurIPS (2017)
- Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* **8**, 229–256 (1992)
- Zhao, X., Wu, W., Xu, C., Tao, C., Zhao, D., Yan, R.: Knowledge-grounded dialogue generation with pre-trained language models. In: EMNLP (2020)
- Zhou, K., Prabhumoye, S., Black, A.W.: A dataset for document grounded conversations. In: EMNLP (2018)



Multi-intent Attention and Top-k Network with Interactive Framework for Joint Multiple Intent Detection and Slot Filling

Xu Jia^(✉), Jiaxin Pan, Youliang Yuan, and Min Peng^(✉)

School of Computer Science, Wuhan University, Wuhan, China
`{jia_xu,pjx_1997,2020282110194,pengm}@whu.edu.cn`

Abstract. Multiple intent detection and slot filling are essential components of spoken language understanding. Existing methods treat multiple intent detection as a multi-label classification task. However, multi-label classification methods focus on the correlation between different intents and set the threshold to select the high probability intents. These methods will cause the model to miss part of the correct intents. In this paper, to address this issue, we introduce Multi-Intent Attention and Top-k Network with Interactive Framework (MIATIF) for joint multiple intent detection and slot filling. In particular, we model the multi-intent attention to obtaining the relation between the utterance and intents. Meanwhile, we propose the top-k network to encode the distribution of different intents and accurately predict the number of intents. Experimental results on two publicly available multiple intent datasets show substantial improvement. In addition, our model saves 64%–72% of training time compared to the current state-of-the-art graph-based model.

Keywords: Interactive framework · Multiple intent detection · Multi-intent attention · Top-k network

1 Introduction

Intent detection and slot filling are significant parts of spoken language understanding [13]. In an utterance, intents and slots always exist a strong correlation. For instance, the slot of movie name “*paris by night*” and the intent “*Search-ScreeningEvent*” correspond to each other in the query “*Rate if tomorrow comes and what time will paris by night aired*”. To model the relation between intents and slots, dominant models [4, 5, 12, 16, 23] adopt joint models to build the relationship between the two tasks. Though achieving promising performances, previous works only focus on the single-intent task. However, the utterances in reality dialogue scenarios express more than a single intent [3]. For example, in Fig. 1, the whole sentence corresponds to the intent “*RateBook*” and the intent “*SearchScreeningEvent*”.

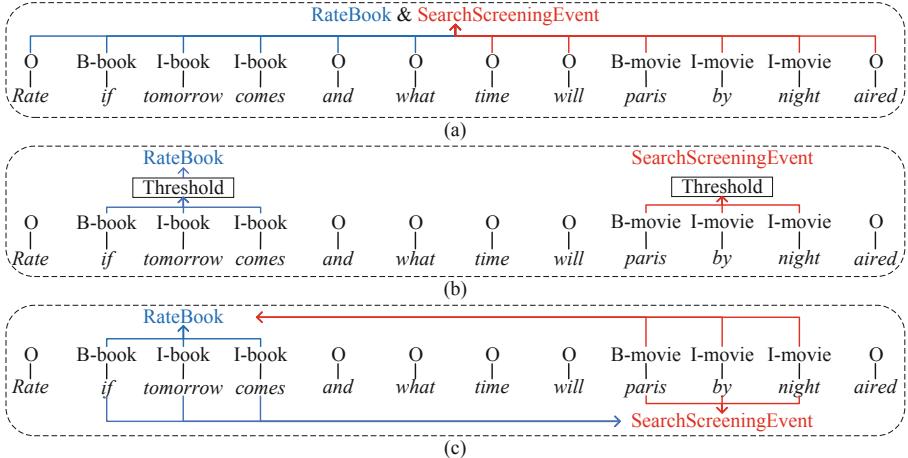


Fig. 1. Prior works treat multiple intents as an entire intent (a) or use multi-label classification methods to filter intents under the threshold (b). Our method discards the threshold and uses the attention between the utterance and intents to determine the final multiple intents (c).

To solve the problem of multiple intents in an utterance, the prior models directly combine multiple intents into a single one, which is shown in Fig. 1(a). However, these models do not guide each word to capture the features corresponding to different intents [17]. To better perform multiple intent detection and slot filling, [3] and [17] achieve promising performance by using multi-label classification methods to consider two tasks jointly, as shown in Fig. 1(b). The multi-label classification methods mainly utilize latent relevance among labels. However, the core of multiple intent detection is to distinguish the irrelevance of different intents. In addition, the method of setting the threshold can only select intents with higher probability. Depicted in Fig. 1(b), the intent “*RateBook*” which is above the threshold can be selected. In Fig. 1(c), these tokens “*paris by night*” not only focus on the intent “*SearchScreeningEvent*”, but also reduce the relevance on the intent “*RateBook*”.

There are two challenges in multiple intent detection: 1) How to distinguish the features of different intents. 2) How to predict the number of multiple intents rather than setting the threshold. To solve these two problems, we propose a **Multi-Intent Attention and Top-k Network with Interactive Framework (MIATIF)**. In particular, we use an interactive framework based on the vanilla transformer to improve the performance of both multiple intent detection and slot filling. We introduce multi-intent attention to capture the relation between the utterance and intents, which helps distinguish different intents’ features. Meanwhile, we construct the top-k network to predict the number of intents by encoding the distribution of different intents. This network can replace the method of setting a threshold to avoid missing low probability intents.

To summarize, the contributions of this paper are: 1) We propose a Multi-Intent Attention and Top-k Network with Interactive Framework to jointly solve

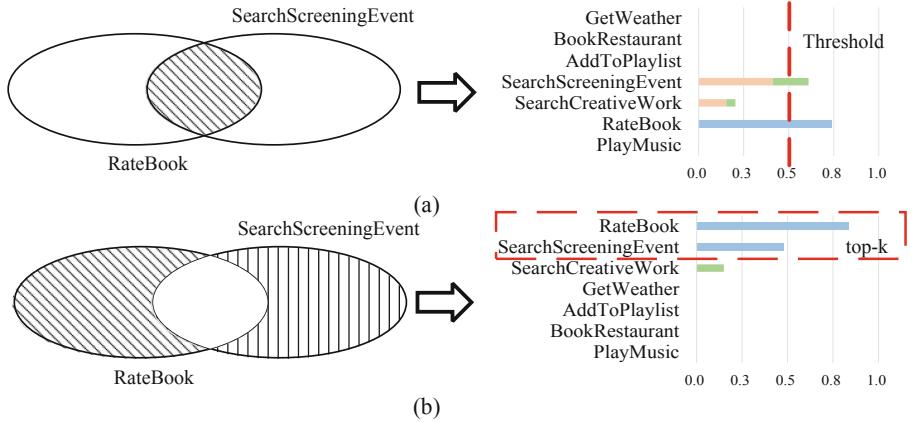


Fig. 2. (a) Multi-label classification methods set the threshold to select high probability labels. (b) Multiple intent detection methods pay more attention to the difference of intents.

the problem of multiple intent detection and slot filling. 2) We introduce multi-intent attention to distinguish the features between different intents. The top-k network predicts the number of intents by encoding the distribution of different intents. 3) We evaluate the performance of our model on two publicly available dialogue datasets. Our model shows improve overall accuracy performance 3.1% and 1.3% on two datasets and save 64%–72% training time compared to the current state-of-the-art method.

2 Problem Definition

Current works treat multiple intent detection as a multi-label classification task. Models select the intents which have high probability by setting the threshold. However, this paper argues that multiple intent detection and multi-label classification are essentially different tasks. As shown in Fig. 2(a), multi-label classification exploits the association between two labels to improve label probability to avoid being filtered by the threshold. In most cases, there is not a strong correlation between the different intents in an utterance. Therefore, the model needs to focus on the features between different intents in multiple intent detection. In this paper, we redefine the task of multiple intent detection.

We define an utterance $U = (w_1, w_2, \dots, w_L)$ consists of a sequence of L words. Multiple intent detection needs to decide the multiple intent label $Y^I = (Y_1^I, \dots, Y_{\hat{k}}^I)$ with \hat{k} possible intents. We should learn a function $f_I : U \rightarrow Y^I$ from sufficient training samples that achieve the mapping from utterance to multiple intents. In most multi-label classification models, $f_I(U) = \{Y^I | Sim(U, \hat{Y}^I) > \delta\}$ will be derived, where $Sim(U, \hat{Y}^I)$ evaluate the relevance scores of all intents and utterance, and δ is the threshold value. In multiple

intent detection, we first learn a top-k function $f_k : U \rightarrow k$, which utilizes the representation of the different intents in the utterance to predict the number of intents. We set the function $f_I(U) = \{Y^I | Top(Sim(U, \hat{Y}^I), f_k(U))\}$, where $Top(Y, k)$ denotes taking the top k values from Y . We train the model to find the best parameter set α that maximizes the likelihood:

$$\arg \max_{\alpha} P(Y^I | f_I(U); \alpha). \quad (1)$$

3 Model

The architecture of our model is shown in Fig. 3, which consists of the multi-intent attention and the top-k network based on the interactive framework.

3.1 Interactive Framework

In single intent detection and slot filling tasks, the interactive framework improves the performance by model the bidirectional connection between the intents and slots [18]. We firstly perform an interactive framework based on the vanilla transformer [20] to multiple intent detection. In the context feature encoder, We adopt the BiLSTM to encode each utterance U to produce a series of hidden states $H = (h_1, h_2, \dots, h_L)$. We use H^C to represent the output of the context feature encoder. Then, we get the explicit multiple intents and slots representation and put them into the interactive framework to make a mutual interaction. We randomly initialize the parameters as intent embedding matrix $W_F^I \in R^{d \times N_I}$ and slot embedding matrix $W_F^S \in R^{d \times N_S}$ (d represents the dimension of hidden states; N_I and N_S represent the number of intents and slots, respectively).

In practice, we use W_F^I and W_F^S to obtain H^I and H^S , respectively:

$$H^I = H^C + softmax(H^C \cdot W_F^I) \cdot W_F^I, \quad (2)$$

$$H^S = H^C + softmax(H^C \cdot W_F^S) \cdot W_F^S. \quad (3)$$

Furthermore, we map the matrix H^I and H^S to queries (Q^I, Q^S) , keys (K^S, K^I) and values (V^S, V^I) by using different linear projections. Finally, we treat Q^S as queries, K^I as keys, and V^I as values and obtain new slot representations incorporating intent information. The new slot representations:

$$\hat{H}^S = H^S + softmax(\frac{Q^S K^I}{\sqrt{d}}) V^I. \quad (4)$$

Similarly, we obtain the new intent representations:

$$\hat{H}^I = H^I + softmax(\frac{Q^I K^S}{\sqrt{d}}) V^S. \quad (5)$$

The interactive framework enables sharing the features of intents and slots. It can avoid the phenomenon of an utterance with correct slots and wrong intent or correct intent and wrong slots.

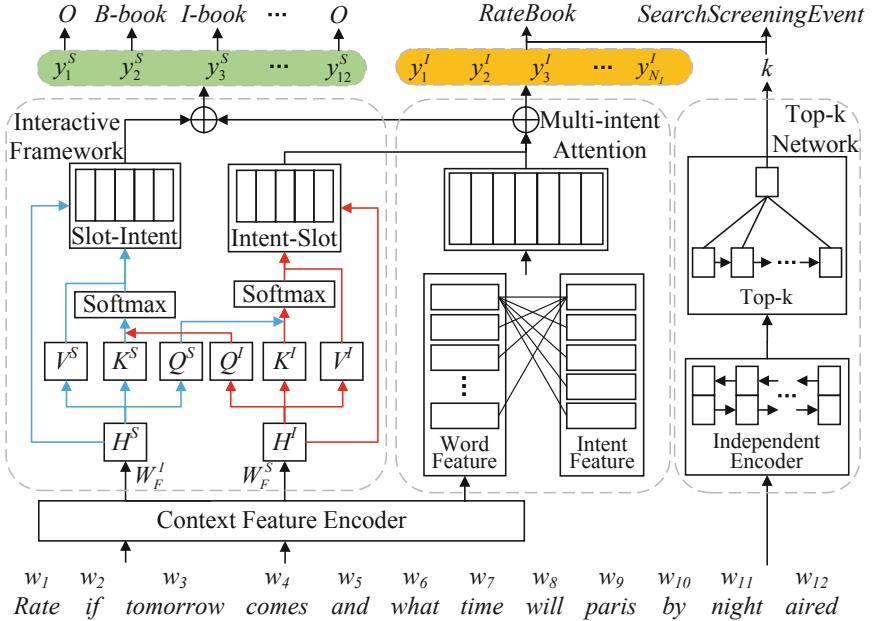


Fig. 3. The illustration of multi-intent attention and top-k network with interactive framework (MIATIF). Multi-intent attention and top-k network distinguish the features of different intents and accurately predict the number of intents.

3.2 Multi-intent Attention and Top-k Network

In this paper, the core contribution is the multi-intent attention and the top-k network. Firstly, multi-intent attention can build the relationship between the utterance and intents, distinguishing features of different intents by the text semantics of utterance. Then, to predict the number of intents, we introduce an independent encoder to encode the different distribution features. We take the representations of different intents and the top-k to predict the number of intents in the utterance.

Multi-intent Attention. The text of intents usually has specific semantics [22]. To make use of the semantic information of multiple intents, we need to obtain the intent embedding matrix $E^I \in R^{N_I \times d}$ in the same latent d -dim space with the words.

After obtaining the hidden states H^C from the context feature encoder and the intent embedding E^I , we can explicitly determine the semantic relation between each pair of words and intents. Attention weights are computed by the dot product between H^C , E^I , and output $A^I \in R^{L \times N_I}$:

$$A^I = \text{softmax}(H^C \cdot E^I); \hat{A}^I = A^I \cdot E^I, \quad (6)$$

where $\hat{A}^I \in R^{L \times d}$ is the relationship between each pair of words and intents. The representation \hat{A}^I is based on multiple intents and words in an utterance. Thus, we call it multi-intent attention.

Top-k Network. In this paper, one challenge we focus on is how to replace the threshold method to predict the number of multiple intents \hat{k} accurately. We propose the top-k network to accomplish multiple intent detection by encoding the distribution of different intents in an utterance. Unlike the method from [10], we do not bother to predict some conjunctions to determine whether it is a multiple intent problem. Therefore our method has universal application scenarios for predicting the number of intents rather than focusing on some special tokens. In particular, to avoid integrating the context features, we use an independent encoder to encode multi-intent distribution. The same with context feature encoder, we can get H^M from the output of the independent encoder.

Then, we use a unidirectional LSTM as the top-k decoder, which predicts the number of multiple intents. The intent distribution vector H^M will be fed to the decoder to predict the number of multiple intents. At each step i , the decoder state s_i^k is calculated by previous decoder state s_{i-1}^k , the previous number of multiple intents k_{i-1} and the aligned encoder hidden state h_i^M :

$$s_i^k = \text{LSTM}(s_{i-1}^k, k_{i-1}, h_i^M); k = \lfloor \sum_{i=1}^L (W_i^n \cdot s_i^k + b_i) + \frac{1}{2} \rfloor, \quad (7)$$

where $\lfloor \cdot \rfloor$ indicates round down.

3.3 Decoder

In multi-intent attention and top-k network with the interactive framework, we have obtained intent representation \hat{H}^I , slot representation \hat{H}^S , the multi-intent attention \hat{A}^I , and the number of multiple intents k . In this section, we build an intent decoder and a slot decoder, respectively.

Intent Decoder. We concatenate the intent representation \hat{H}^I and the multi-intent attention \hat{A}^I as the representation of the final inputs:

$$\tilde{H}^I = \hat{H}^I \oplus \hat{A}^I, \quad (8)$$

where $\tilde{H}_I \in R^{T \times 2d}$ and \oplus is an operation for concatenating two vectors.

We use a unidirectional LSTM as the multiple intent detection decoder:

$$s_i^I = \text{LSTM}(s_{i-1}^I, y_{i-1}^I, \tilde{h}_i^I). \quad (9)$$

Then the decoder state s_i^I is utilized for multiple intent detection:

$$y^I = \sigma(\text{LeakyReLU}(W_1^I s_i^I + b_1^I) W_2^I + b_2^I), \quad (10)$$

where W_1^I, W_2^I are trainable parameters of the intent decoder, $y^I = \{y_1^I, \dots, y_{n_I}^I\}$ is the intent output of the utterance and σ represents the activation function.

We use the number of multiple intents k in each utterance during inference instead of setting a threshold. The final result O^I is generated by intent output y^I and the number of multiple intents k . We get the top-k largest intent distributions as the final output. For example, if the $y^I = \{0.7, 0.1, 0.3, 0.9, 0.5, 0.1, 0.2\}$ and the k is 2, we predict intents $O^I = \{1, 4\}$.

Slot Decoder. For the slot filling decoder, we similarly use another unidirectional LSTM as the slot filling decoder. To ensure the performance of the slot filling task, we leverage multiple intent features to guide the slot prediction. At the decoding step i , the decoder state s_i^S can be formalized as:

$$s_i^S = LSTM(s_{i-1}^S, y_{i-1}^S, \hat{h}_i^S \oplus \tilde{h}_i^I). \quad (11)$$

Similarly, the decoder state s_i^S is utilized for slot filling:

$$y_i^S = softmax(W_d s_i^S); O_i^S = argmax(y_i^S), \quad (12)$$

where O_i^S is the slot label of the i -th word in the utterance.

3.4 Joint Training

Following [4, 16, 17], we adapt a joint model to consider the three tasks and update parameters by joint optimizing. The intent detection, slot filling, and top-k loss functions are:

$$\mathcal{L}^I = - \sum_{m=1}^{n_I} (\hat{y}_m^I \log(y_m^I) + (1 - \hat{y}_m^I) \log(1 - y_m^I)), \quad (13)$$

$$\mathcal{L}^S = - \sum_{i=1}^{n_W} \sum_{j=1}^{n_W} \hat{y}_i^{(j,S)} \log y_i^{(j,S)}, \quad (14)$$

$$\mathcal{L}^k = |k - \hat{k}|, \quad (15)$$

where \hat{y}^I , \hat{y}^S and \hat{k} are the gold intent label, gold slot label, and the gold number of intents, respectively.

The final joint objective is formulated as:

$$\mathcal{L} = \mathcal{L}^I + \mathcal{L}^S + \mathcal{L}^k. \quad (16)$$

4 Experiments

4.1 Datasets

Since other single intent datasets cannot evaluate multi-intent models, we evaluate the performance of our model on the only two publicly available multiple intent datasets, MixATIS and MixSNIPS. Both datasets are used in our paper following the same format and partition as in [17].

MixATIS and MixSNIPS datasets are collected from the ATIS [6] and SNIPS [2] which are widely used in SLU task, respectively. [17] utilizes conjunctions to connect sentences with different intents. The number of intents in the datasets is no more than 3, and the ratio between 1–3 intents is 3 : 5 : 2. MixATIS has 18000 utterances for training, 1000 utterances for validation, and 10000 utterances for testing. MixSNIPS has 45000 utterances for training, 2500 utterances for validation, and 2500 utterances for testing. In the training set, MixATIS has 17 different intents, and MixSNIPS has 7.

Table 1. Slot filling and intent detection results on two multi-intent datasets

Model	MixATIS				MixSNIPS			
	Slot(F1)	Intent(F1)	Intent(Acc)	Overall	Slot(F1)	Intent(F1)	Intent(Acc)	Overall
Attention BiRNN [12]	86.6	—	71.6	38.7	89.4	—	94.1	62.2
Slot-Gated [4]	88.1	—	65.7	38.9	87.8	—	96	56.5
SF-ID [5]	87.7	—	63.7	36.2	89.6	—	96.3	59.3
Stack-propagation [16]	87.4	79	71.9	41	93.2	97.6	94.6	71.9
Joint multiple ID-SF [3]	87.5	80.6	73.1	38.1	91	98.2	95.7	66.6
AGIF [17]	88.1	81.2	75.8	44.5	94.5	98.6	96.5	76.4
MIATIF	88.0	78.6	76.0	47.6	94.6	98.6	97.1	77.7

4.2 Implementation Details

The encoder and decoder hidden units are 256 and 128 in all datasets, respectively. We use Adam to optimize the parameters in our model and adapt the suggested hyper-parameters for optimization. For all experiments, we pick the model which the sentence-level accuracy works best on the dev set and then evaluate it on the test set. The epochs are 200 and 100, and the dropout rates are 0.3 and 0.4 for MixATIS and MixSNIPS, respectively. Part of the code uses the MindSpore Lite tool [1].

4.3 Main Results

Following [4] and [17], we use Slot(F1), Intent(F1), Intent(Acc) and Overall to evaluate the performance of slot filling, intent detection and sentence-level accuracy. We adopt the top-k network to predict the number of multiple intents, and the results are 98.6% and 99.6% in the MixATIS and MixSNIPS datasets. Table 1 shows the other experimental results of the proposed models on the MixATIS and MixSNIPS datasets. Among the baselines, [4, 5, 12, 16] are the classical model for single intent, [3, 17] achieve state-of-the-art on multiple intent.

We have the following observations from the results: 1) Our model outperforms baseline and achieves promising performances. On the MixATIS dataset, our model achieves 0.2% and 3.1% absolute gains on Intent(Acc) and Overall, respectively. On the MixSNIPS dataset, our model achieves the best results on all metrics, where it improves 0.6% on Intent(Acc) and 1.3% on Overall. The improvement indicates that our model successfully solves the challenge of multiple intent detection and improves the performance of both tasks. 2) The high accuracy of the number of intents reaching 98% has been shown that the top-k network can be relatively reliable. So it ensures that our model will not filter the part of correct intents and only select high probability intents. 3) Compared to the improvement in Intent(Acc), the improvement in Overall is more significant on both datasets. It is because we select the model which has the best performance of Overall on the dev sets. Also, we use the interactive framework to make the sentence-level accuracy perform better by fully interacting with the features of slots and intents in the utterances. 4) The improvements of our

model on the Slot(F1) and Intent(F1) are not significant. The reason is that *AGIF* extracts intent features and builds a graph structure to guide slot filling. Meanwhile, they use the threshold to select high probability intents, resulting in higher Intent(F1). However, the graph structure is time-consuming. We use the multi-intent attention to obtain an acceptable slight decrease of Slot(F1) while improving Intent(Acc) and Overall performance.

Table 2. Comparison of training time

Model	MixATIS		MixSNIPS	
	Epoch(s)	All(h:m:s)	Epoch(s)	All(h:m:s)
AGIF	207.5	5:45:50	473.3	6:34:27
MIATIF	74.2	4:07:10	131.5	3:39:14

To show the efficiency of our model, we compare training time with *AGIF*. Table 2 shows the results on the two datasets, where Epoch(s) indicates the average seconds consumed in one epoch and All(h:m:s) represents the time required to complete the full training. As every epoch, our model saves 64% and 72% time consumption, respectively. Although our epoch is twice of *AGIF*, our model still saves 29%–44% of the total training time consumption.

4.4 Ablation Study

In this section, we set up the following ablation experiments to study the impact of our model. The result is shown in Table 3.

Effectiveness of Interactive Framework. For the first time, we apply the interactive framework from single intent detection to multiple intent detection. To verify the validity of the framework, we remove the interactive framework from the model and replace H^I and H^S with H^C . It means that we only get the context feature from the encoder and directly input it into the decoder without incorporating the features of intents and slots. We name it as *without interaction*. From the result, Slot(F1) performances both drop 1.0%, and Intent(Acc) performance drops 0.6% and 0.8%. It results in overall performances drop of 4.4% and 2.5%. The decline Overall is significant without the interactive framework, indicating that the interactive framework plays a key role in sentence-level accuracy. Slot(F1) drops significantly due to the lack of intent features. We introduce multi-intent attention, so Intent(Acc) decreases insignificantly. It verifies that incorporating the intent and slot features is useful for improving the performance of both two tasks.

Effectiveness of Multi-intent Attention. We remove the multi-intent attention and utilize the output H^I of the interactive framework to the intent decoder. We name it as *without multi-intent attention*. From the result, Overall performances both drop 2.9% on the two datasets. We believe the main reason is

Table 3. Ablation experiments on the MixATIS and MixSNIPS datasets

Model	MixATIS				MixSNIPS			
	Slot(F1)	Intent(F1)	Intent(Acc)	Overall	Slot(F1)	Intent(F1)	Intent(Acc)	Overall
W/o interactive	87.0	78.1	75.4	43.2	93.6	98.1	96.3	75.2
W/o multi-intent attention	87.1	78.5	74.6	44.7	94.0	98.1	96.2	74.8
W/o top-k network	87.8	80.3	74.7	44.2	94.5	98.7	96.7	75.9
MIATIF	88.0	78.6	76.0	47.6	94.6	98.6	97.1	77.7

the decline in Intent(Acc). Since the lack of multi-intent attention, the model cannot distinguish features between different intents. Also, the interactive framework will pass the error to the slot filling, which leads to the decline of Slot(F1) slightly.

Effectiveness of Top-k Network. Instead of adopting the top-k network, we utilize the threshold to predict the multiple intents. We define it as *without top-k network*. This structure is similar with [3] and [17], which perform the multiple intent detection as the multi-label classification. From the result, we observe the overall performances drop 2.4% and 1.8% on the two datasets. We attribute it to the fact that the top-k network can avoid missing useful features of intents. Meanwhile, we observe that the Intent(F1) improves on both datasets due to threshold replacement. Since setting threshold only selects high probability intents, it leads to higher performance on Intent(F1).

5 Related Work

In the current works, intent detection(ID) is usually considered a classification task and slot filling(SF) as a sequence labeling task. So traditional machine learning methods are often used on these two tasks [9, 19]. In recent years, various neural architectures have achieved the state-of-the-art [5, 7, 12, 13, 15, 21]. Due to the strong correlation between the two tasks, the joint model is the currently effective method. The initial works use loss function via backpropagation to verify the parameter of the sharing encode module [12, 24]. The later models utilize the features of intent detection to enhance the features of slot filling [4, 11, 16], and establish the connection between the two tasks using gate mechanism or graph structure [5, 14, 18].

Although the above joint models have handled both tasks simultaneously, the current single-intent scenario cannot represent the multi-intent scenario. [3] proposes the task of multiple intent detection and introduces the slot-gated mechanism based on token-level to capture the features between intents and slots. To push forward the research of multi-intent SLU, [17] releases two large-scale multi-intent datasets MixATIS and MixSNIPS, based on ATIS and SNIPS. Then [17] introduces an intent-slot graph construction to model the relation between multi-intent and slot filling tasks. Previous works treat multiple intent detection as a multi-label classification task and achieve the promising performance [3, 8, 17]. Therefore, the above works ignore the differences between different intents. And

the threshold only selects intents with high probability. This paper introduces the multi-intent attention and the top-k network to accomplish multiple intent detection and slot filling tasks jointly.

6 Conclusion and Future Work

In this paper, we propose Multi-Intent Attention and Top-k Network with Interactive Framework (MIATIF) for joint multiple intent detection and slot filling. Our model first introduces an interactive framework based on the vanilla transformer in multiple intent detection. Then, to better exploit the features of different intents, we propose multi-intent attention. Furthermore, we utilize the independent encoder to alleviate the mixed context features on multiple intents, and the top-k predicts the number of intents. Our model improves performance for overall accuracy on the MixATIS and MixSNIPS of 3.1% and 1.3%, respectively. Simultaneously, our model saves 64%–72% of training time compared to the current state-of-the-art model while achieving better results. In the future, we also want to introduce pre-trained models to improve the performance using MindSpore.

Acknowledgment. We thank anonymous reviewers for their precious comments. This research is supported by MindSpore, the National Key R&D Program of China under Grant No. 2018YFC1604003, General Program of Natural Science Foundation of China (NSFC) under Grant No. 61772382 and No. 62072346, Key R&D Project of Hubei Province under Grant No. 2020BAA021 and Science and Technology Plan of Wuhan under Grant No. 2020010601012168.

References

1. Mindspore. <https://www.mindspore.cn/> (2020)
2. Coucke, A., et al.: Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. arXiv preprint [arXiv:1805.10190](https://arxiv.org/abs/1805.10190) (2018)
3. Gangadharaiah, R., Narayanaswamy, B.: Joint multiple intent detection and slot labeling for goal-oriented dialog. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), pp. 564–569 (2019)
4. Goo, C.W., et al.: Slot-gated modeling for joint slot filling and intent prediction. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), pp. 753–757 (2018)
5. Haihong, E., Niu, P., Chen, Z., Song, M.: A novel bi-directional interrelated model for joint intent detection and slot filling. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 5467–5471 (2019)
6. Hemphill, C.T., Godfrey, J.J., Doddington, G.R.: The ATIS spoken language systems pilot corpus. In: Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania (1990)

7. Hou, Y., et al.: Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 1381–1393 (2020)
8. Hou, Y., et al.: Few-shot learning for multi-label intent detection. In: The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI) (2021)
9. Huang, J., et al.: A probabilistic method for emerging topic tracking in microblog stream. *World Wide Web* **20**(2), 325–350 (2017)
10. Kim, B., Ryu, S., Lee, G.G.: Two-stage multi-intent detection for spoken language understanding. *Multimed. Tools Appl.* **76**(9), 11377–11390 (2017)
11. Li, C., Li, L., Qi, J.: A self-attentive model with gate mechanism for spoken language understanding. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 3824–3833 (2018)
12. Liu, B., Lane, I.: Attention-based recurrent neural network models for joint intent detection and slot filling. In: Proceedings of the 17th Annual Conference of the International Speech Communication Association (INTERSPEECH), pp. 685–689 (2016)
13. Louvan, S., Magnini, B.: Recent neural methods on slot filling and intent classification for task-oriented dialogue systems: a survey. In: Proceedings of the 28th International Conference on Computational Linguistics (COLING), pp. 480–496 (2020)
14. Peng, H., Shen, M., Jiang, L., Dai, Q., Tan, J.: An interactive two-pass decoding network for joint intent detection and slot filling. In: Zhu, X., Zhang, M., Hong, Yu., He, R. (eds.) NLPCC 2020. LNCS (LNAI), vol. 12431, pp. 69–81. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60457-8_6
15. Peng, M., et al.: Personalized app recommendation based on app permissions. *World Wide Web* **21**(1), 89–104 (2018)
16. Qin, L., Che, W., Li, Y., Wen, H., Liu, T.: A stack-propagation framework with token-level intent detection for spoken language understanding. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2078–2087 (2019)
17. Qin, L., Xu, X., Che, W., Liu, T.: Towards fine-grained transfer: an adaptive graph-interactive framework for joint multiple intent detection and slot filling. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings (EMNLP), pp. 1807–1816 (2020)
18. Qin, L., et al.: A co-interactive transformer for joint slot filling and intent detection. In: ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8193–8197. IEEE (2021)
19. Raymond, C., Riccardi, G.: Generative and discriminative algorithms for spoken language understanding. In: Eighth Annual Conference of the International Speech Communication Association (INTERSPEECH) (2007)
20. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems (NIPS), pp. 5998–6008 (2017)
21. Wu, J., et al.: Joint learning of word and label embeddings for sequence labelling in spoken language understanding. In: 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 800–806. IEEE (2019)
22. Xiao, L., Huang, X., Chen, B., Jing, L.: Label-specific document representation for multi-label text classification. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 466–475 (2019)

23. Zhang, C., et al.: Joint slot filling and intent detection via capsule neural networks. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 5259–5267 (2019)
24. Zhang, X., Wang, H.: A joint model of intent determination and slot filling for spoken language understanding. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI), vol. 16, pp. 2993–2999 (2016)



Enhancing Long-Distance Dialogue History Modeling for Better Dialogue Ellipsis and Coreference Resolution

Zixin Ni^{1,2} and Fang Kong^{1,2(✉)}

¹ Laboratory for Natural Language Processing, Soochow University, Suzhou, China
20194227053@stu.suda.edu.cn, kongfang@suda.edu.cn

² School of Computer Science and Technology, Soochow University, Suzhou, China

Abstract. Previous work on dialogue-specific ellipsis and coreference resolution usually concatenates all dialogue history utterances into a single sequence. It may mislead the model to attend to inappropriate parts and to copy from wrong utterances when the dialogue history is long. In this paper, we aim to model dialogue history from multiple granularities and take a deep look into the semantic connection between the dialogue history and the omitted or coreferred expressions. To achieve this, we propose a speaker highlight dialogue history encoder and a top-down hierarchical copy mechanism to generate the complete utterances. We conduct dozens of experiments on the CamRest676 dataset, and the experimental results show that our methods are expert in long-distance dialogue history modeling and can significantly improve the performance of ellipsis and coreference resolution in the dialogue task.

Keywords: Ellipsis recovery · Coreference resolution · Dialogue history

1 Introduction

Dialogue is known to be an activity generated by different speakers, and the expression of each speaker is usually based on the understanding of dialogue histories. When using dialogue systems, people tend to avoid repetitive expressions, which results in the notorious omission and coreference problems. Table 1 presents two typical examples on ellipsis and coreference phenomena in multi-turn dialogue. For example, the expression ‘*Yu Garden*’ is omitted in Q3 to avoid repetition (Ellipsis), and the pronoun ‘*they*’ in Q2 refers to ‘*The Curry Prince*’ in A1 (Coreference). Dialogue ellipsis and coreference resolution as a preprocessing sub-task of the multi-turn dialogue task, it can well resolve coreference and ellipsis phenomena in dialogue history to help machine speakers better understand the user’s intent and thus generate more reasonable responses.

In most cases, the omitted or coreferred expressions are nearly from either the dialogue history utterances or the incomplete utterances. Observing this, previous researches heavily employ pointer nets (Su et al. [12], Zhang et al. [16]) or sequence-to-sequence models with copy mechanisms (Quan et al. [9])

Table 1. Examples of dialogue ellipsis and coreference resolution

Turn	Speaker	Dialogue
Q1	usr	I'm looking for a place to eat in the east side of the city
A1	sys	The Curry Prince is on the east side of town
Q2	usr	Do they serve chinese food?
A2	sys	No they do not serve Chinese food, however he Yu Garden does and is in the east side
Q3	usr	What is the address?
A3	sys	Yu Garden's address is 529 Newmarket Road Fen Ditton. Is there anything else I can help you with today?
Q4	usr	Yes, what is the price range?
A4	sys	Yu Garden is in the expensive price range
User utterances after resolution		
Q2		Does The Curry Prince (they) serve chinese food?
Q3		What is the address of the Yu Garden ?
Q4		Yes, what is the price range of the Yu Garden ?

for dialogue ellipsis and coreference resolution. To our knowledge, almost all previous work treats dialogue history utterances indiscriminately, that is, they simply concatenate all the user and system utterances in previous dialogue turns into a single sequence as the history information. However, it is usually difficult for neural models to capture the omitted and coreferred expressions when the dialogue history is very long. Moreover, since the content to be recovered usually comes from the history utterances that are relevant to the current utterance, it is not targeted to take all the history information into consideration.

To tackle these problems, we introduce an encoder-decoder architecture for multi-turn dialogue as our baseline system. Based on it, we propose two approaches to better capture the semantic information from dialogue history, especially in the case of long-distance dialogue history. First, simulating the process of dialogue interaction, we introduce a speaker highlight dialogue history encoder for better global representation of dialogue history and to permit the model to view the dialogue history of each speaker independently. Second, we propose a top-down hierarchical copy mechanism to select and copy relevant expressions from relevant history utterances at different levels. Experimental results show that our proposed methods have good adaptability to the long-distance dialogue history, and can achieve competitive performance in dialogue ellipsis and coreference resolution when compared with the baseline systems.

2 Model

Following Quan et al. [9], we formulate dialogue ellipsis and coreference resolution as a sequence-to-sequence generative problem. Given the n -th user utterance $U_n = (u_1, u_2, \dots, u_s)$ and its dialogue history $H = \{(U_1, R_1), (U_2, R_2), \dots, (U_{n-1}, R_{n-1})\}$

corresponding to all the previous dialogue turns, where R_i represents the system response of the i -th turn, the goal is to recover the ellipsis or coreference for the current utterance U_n . More formally, the dialogue ellipsis and coreference resolution task can be formulated as $((H, U_n) \rightarrow U_c)$ where each token of U_c is generated from current user utterance or its dialogue history.

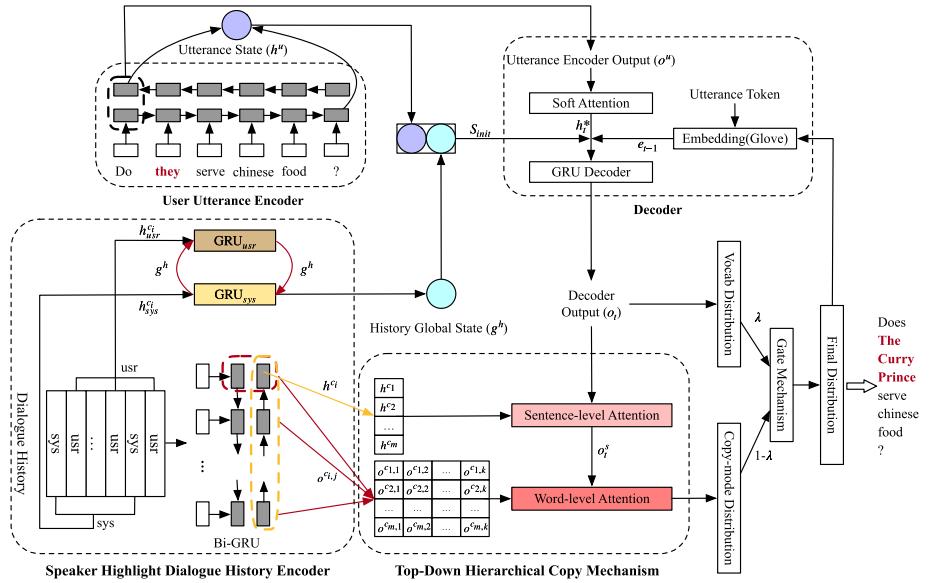


Fig. 1. The encoder-decoder architecture of the proposed dialogue ellipsis and coreference resolution model.

Figure 1 illustrates the dialogue ellipsis and coreference resolution model which mainly consists of four components: (i) user utterance encoder; (ii) speaker highlight dialogue history encoder (SH-DHE); (iii) decoder; (iv) top-down hierarchical copy mechanism (TDH-CM).

2.1 User Utterance Encoder

During the encoding process, the input user utterance of length s is first transformed into a sequence of D -dimensional word embeddings, obtaining $E_u = (e_1^u, e_2^u, \dots, e_s^u)$. Then the word embedding sequence is fed to a bi-directional GRU network to capture the context features of the sequence. Concretely, each token e_i^u of the user utterance is obtained through the concatenation of both forward and backward hidden states, i.e., $h_i^u = [\overrightarrow{h_i^u}, \overleftarrow{h_i^u}]$. And the representation of the user utterance is obtained through the concatenation of the last hidden states in the two directions, as shown in Eq. (3).

$$o^u, h^u = \text{BiGRU}(E_u, \theta) \quad (1)$$

$$o^u = [h_1^u, h_2^u, \dots, h_s^u] \quad (2)$$

$$h^u = [\overrightarrow{h_s^u}, \overleftarrow{h_1^u}] \quad (3)$$

2.2 Speaker Highlight Dialogue History Encoder

In the literature, previous studies usually connect all the user and system utterances in series as the input of the dialogue history encoder. However, utterances in dialogue history usually contain some useless expressions and even noises, and encoding the entire connected utterances with RNNs is really a challenge, especially when the dialogue history is long. To tackle the above-mentioned problem, we propose to encode each utterance separately in this work.

Local Representation. Local representation means encoding each utterance in dialogue history. Given an utterance in the dialogue history H (with $M = 2(n-1)$ utterances), we get representation o^{c_i} through a GRU encoder that shares the same architecture as the user utterance encoder. And the encoder outputs and local representations of the utterances in the dialogue history H can be written as $\{o^{c_1}, o^{c_2}, \dots, o^{c_M}\}$ and $\{h^{c_1}, h^{c_2}, \dots, h^{c_M}\}$ respectively.

Global Representation. So as to better encode the dialogue history to get its global representation g^h , we propose a speaker highlight mechanism to simulate the interactive process of human-to-machine dialogue enlightened by Shi et al. [11]. In this work, our proposed model also makes a sequential scan of each utterance in H . For instance, the dialogue history path of Q_4 is $Q_1 \rightarrow A_1 \rightarrow Q_2 \rightarrow A_2 \rightarrow Q_3 \rightarrow A_3$, as shown in Table 1. Among them, Q_1, Q_2 and Q_3 are from the same speaker *usr* and A_1, A_2 and A_3 are from the same speaker *sys*. We feed the local representations from the same speaker to the same encoder, so as to highlight the previous dialogue history from the same speaker and help our model to better understand the development of the dialogue history involving this speaker.

Since there are only two speakers in human-to-machine dialogue, we use two GRU cells to encode them separately. Let $h_{a_i}^{c_i}$ denote the local utterance representation h^{c_i} from the speaker a_i . The entire process cyclically inputs the local utterance representation $h_{a_i}^{c_i}$ into its corresponding encoder according to the dialogue history path, and the global representation is built incrementally. We calculate the global representation of dialogue history as following:

$$g^h = \begin{cases} 0 & i = 0 \\ \text{GRU}_{\text{usr}}(h_{a_i}^{c_i}, g^h) & a_i = \text{usr}, i \in \{1, \dots, M\} \\ \text{GRU}_{\text{sys}}(h_{a_i}^{c_i}, g^h) & a_i = \text{sys}, i \in \{1, \dots, M\} \end{cases} \quad (4)$$

where *usr* and *sys* denote the input of GRU cell comes from user utterance and system response, respectively.

After that, we concatenate the representation h^u of the user utterance and global representation of dialogue history g^h as the initial state, S_{init} , of the decoder.

$$S_{init} = h^u \oplus g^h \quad (5)$$

where \oplus denotes the vector concatenation operation.

2.3 Decoder

In the decoder, we use a single-layer unidirectional GRU to generate probability distribution over the vocabulary V . Following Bahdanau et al. [1], we use the previous hidden state s_{t-1} and the representation h_i^u of token u_i from user utterance encoder to calculate the attention weights at the time t , noted by a^t . The attention weights represent the contribution of each token to the omission recovery of the current position.

$$attn_i^t = v^T \tanh(w_1 h_i^u + w_2 s_{t-1} + b_1) \quad (6)$$

$$a^t = \text{softmax}(attn_i^t) \quad (7)$$

where v , w_1 , w_2 , and b_1 are learnable parameters. Then the attention distribution a^t is used to calculate a weighted sum of the representation h_i^u , which is known as context vector h_t^* .

$$h_t^* = \sum_i a^t h_i^u \quad (8)$$

Then we feed the context vector h_t^* , the previous decoder state s_{t-1} , and the word embedding e_{t-1} of the previously generated word into the single-layer unidirectional GRU to update the decoder state s_t at timestep t . The decoder output o_t is then concatenated with the context vector h_t^* to produce probability distribution over the vocabulary V . The generate-mode distribution to predict token y_t is calculated as:

$$o_t, s_t = \text{GRU}([e_{t-1}; h_t^*], s_{t-1}) \quad (9)$$

$$P_{vocab}(y_t | y_{1:t-1}) = w_3 h_t^* + w_4 o_t + b_2 \quad (10)$$

where w_3 , w_4 , and b_2 are learnable model parameters.

2.4 Top-Down Hierarchical Copy Mechanism

For the proposed TDH-CM model, it should be noted that, we use a coarse-grain to fine-grain approach to calculate the probabilities for tokens copied from the dialogue history, which are parts of the omitted or coreferred expressions. As described before, the SH-DHE architecture yields two kinds of memories: (i) Sentence-level memories, i.e., local utterance representations $\{h^{c_1}, h^{c_2}, \dots, h^{c_M}\}$ which concatenates the final hidden states of both forward and backward GRUs for each utterance. (ii) Word-level memories, i.e., utterance encoder outputs $\{o^{c_1}, o^{c_2}, \dots, o^{c_M}\}$ where each o^{c_i} with $i \in \{\overrightarrow{1, \dots, M}\}$ is a set of concatenated hidden states at step $j \in \{1, \dots, k\}$, $o^{c_{i,j}} = [\overrightarrow{h_{i,j}}, \overleftarrow{h_{i,j}}]$.

Firstly, we compute the relevance between the decoder output o_t and sentence-level utterance representation h^{c_i} , so that figure out which utterances mention the relevant omitted or coreferred expressions. After feeding it to a softmax layer, we obtain a sentence-level attention distribution $a_{t,i}^s$ as:

$$r_i^s = o_t^T h^{c_i} \quad (11)$$

$$a_{t,i}^s = \text{softmax}(r_i^s) \quad (12)$$

The sentence-level attention distribution $a_{t,i}^s$ is used to calculate a weighted sum of local utterance representations h^{c_i} , known as sentence-level information h_t^s . Then we update the representation of the decoder output by combining o_t with sentence-level information h_t^s through a linear transformation as:

$$h_t^s = \sum_{i=1}^M a_{t,i}^s \cdot h^{c_i} \quad (13)$$

$$o_t^s = w_5[o_t; h_t^s] + b_3 \quad (14)$$

where w_5 and b_3 are learnable parameters, and the update decoder output o_t^s will be used in subsequent word-level inference.

Secondly, we aim to capture the token sequence from relevant utterances which are parts of the omitted or coreferred expressions. Thus, we map the updated decoder output o_t^s into the space of the word-level memories $o^{c_{i,j}}$ by measuring the similarity as following:

$$P_{copy}(y_t|y_{1:t-1}) = w_6(o_t^s \odot o^{c_{i,j}}) + b_4 \quad (15)$$

where w_6 and b_4 are learnable parameters. This similarity gives a word-level attention which is treated as the probabilities for tokens copied from the dialogue history.

Finally, we aim to enhance our model to learn whether to copy tokens from U_n or history utterances at different steps. To achieve this, we use a gate mechanism to adaptively fuse the generation and copy mode to the final probability distribution. The final probability distribution is calculated as following:

$$\lambda = \sigma(w_7[h_t^*; e_{t-1}; s_t] + b_5) \quad (16)$$

$$P(y_t|y_{1:t-1}) = \lambda P_{vocab}(y_t|y_{1:t-1}) + (1 - \lambda)P_{copy}(y_t|y_{1:t-1}) \quad (17)$$

where w_7 and b_5 are learnable parameters, and σ denotes the sigmoid function.

3 Experiment

3.1 Data and Metrics

The CamRest676 corpus (Quan et al. [9]) is a publicly available data that annotates the resolution of ellipsis and coreference in multi-turn dialogue. The corpus

contains 676 dialogues with 2744 user utterances. Among them, 1174 ellipsis versions and 1209 coreference versions are created from the user utterance. According to our statistics, each utterance contains around 12.51 word units on average, and each dialogue is composed of around 4 dialogue turns on average. Following previous studies, we use 80% of the data as a training set and the remaining 20% as a validation set.

We adopt the automatic metrics BLEU, EM, one word precision, recall, and F1 score as the main evaluation methods. Among them, BLEU evaluates the similarity between the incomplete utterance and the golden ones at the n-gram level. The exact match rate (EM) is the strictest evaluation metric that measures whether the generated utterances match the golden ones or not. For the EM score, we report on the complete and incomplete utterances separately to see the difference, denoted as EM1 and EM2.

3.2 Experimental Settings

We optimized the following parameters during training: the learning rate was 0.001, the decay parameter was 0.5, the size of hidden states was 128. We employed the 50-dimension word embeddings provided by Glove and did not fine-tune the pre-trained vectors during training. The vocabulary size V was set to 800 and the batch size was 8. In order to prevent our model from over-fitting, we adopted dropout after embedding, Bi-GRU and decoder, respectively, and the dropout rate was 0.5. We employed Adam as the optimizer for model learning with the early stopping strategy used to avoid the problem of over-fitting, and the patience value was 12. It is worth mentioning that we used the standard cross-entropy loss as the loss function to train the entire model.

3.3 Experimental Results

In this paper, we compare our proposed system with the following competitive models on the CamRest676 corpus:

- **GECOR:** an end-to-end generative model proposed by Quan et al. [9] that uses the popular copy mechanism to recover omission. They use a sequence-to-sequence model as their baseline which simply connects all dialogue history utterances in series as the input of history encoder, known as GECOR-basic. Notably, we also reproduce their results over different system settings where GECOR1 means the model with the copy mechanism and GECOR2 means the model with the gated copy mechanism.
- **Baseline:** a sequence-to-sequence model which first encodes each utterance of dialogue history separately, and then concatenates each local representation in dialogue history and user utterance representation as the initial state of the decoder.

It is noted that Liu et al. [6] achieves state-of-the-art on CamRest676, which uses a series of edit operations (i.e. substitute and insert) on the incomplete

utterance, but our model is based on the generative framework so we will not compare it here. Following Quan et al. [9], we train our model on three types of datasets: the ellipsis dataset which only annotates the ellipsis version utterances, the coreference dataset which only annotates the coreference version utterances, and the mixed dataset which randomly selects a version for each user utterance from the ellipsis and coreference version. The quantitative evaluation results of the above models are shown in Tables 2, 3, and 4.

First, compared with GECOR-basic, our introduced Baseline system can improve the performance of both dialogue ellipsis and coreference resolution on all the seven metrics. This indicates that our approach of encoding the dialogue history utterances separately can effectively capture the long-distance dependency information hidden within the global dialogue history.

Table 2. Performance comparison on the ellipsis dataset. † denotes the duplicated systems.

	EM	EM1	EM2	BLEU	F1	Prec.	Rec.
GECOR-basic †	51.72	71.33	28.15	73.81	91.45	92.71	90.23
GECOR1 †	67.32	92.70	37.82	83.21	96.37	98.46	94.36
GECOR2 †	65.20	90.53	34.87	83.11	96.40	98.31	94.57
Baseline	56.87	78.28	30.34	75.18	91.70	92.90	90.53
Final	67.86	92.83	36.86	83.97	95.94	98.24	93.75
w/o SH-DHE	66.79	91.81	35.86	82.63	95.61	98.18	93.17
w/o TDH-CM	57.03	79.11	29.49	76.57	91.87	92.66	91.09

Table 3. Performance comparison on the coreference dataset.

	EM	EM1	EM2	BLEU	F1	Prec.	Rec.
GECOR-basic †	52.72	71.16	32.79	75.98	91.45	91.97	90.93
GECOR1 †	71.15	90.98	47.80	85.41	96.32	97.95	94.74
GECOR2 †	70.31	92.45	44.56	85.30	96.80	98.28	95.36
Baseline	60.08	80.66	36.29	79.90	92.84	93.28	92.41
Final	73.58	94.53	49.37	87.66	96.54	97.90	95.22
w/o SH-DHE	71.65	93.12	46.86	87.03	96.24	97.83	94.71
w/o TDH-CM	60.67	81.75	36.29	80.96	93.40	94.06	92.74

Second, since we also use a gate mechanism in our system as GECOR2 did, we mainly compare it with GECOR2. And the results (lines 4 and 6) show that our approach can greatly improve the performance on EM and BLEU by 2.66 points and 0.86 points respectively on the ellipsis dataset, 3.27 points and 2.36 points respectively on the coreference dataset, and 4.82 points and 1.51 points

respectively on the mixed dataset. The increments over GECOR2 suggest the strong capabilities of our model in establishing a dependency relationship within dialogue history. Moreover, we also note that the increments on the coreference resolution is higher than those on the ellipsis resolution which indicate that the model is better for learn due to more explicit pronoun guidance for the referent case. In addition, the EM2 metric focuses on incomplete user utterances which mainly evaluates whether the model fills the omitted content is accurate or not. And the improvements indicate that our model can effectively recover the omitted and coreferred information of incomplete user utterances.

Nevertheless, our model only achieves performance similar to GECOR2 on some metrics (e.g., F1, Prec., and Rec.). And one possible reason could be that: our proposed method is tailored for long-distance dialogue history modeling, while these short dialogue history scenarios will weaken the ability of our model in distinguishing between positive and negative examples. To figure out this, we will provide a deep analysis on this problem in Sect. 3.4.

Table 4. Performance comparison on the mix dataset.

	EM	EM1	EM2	BLEU	F1	Prec.	Rec.
GECOR-basic †	50.29	71.38	27.78	73.78	90.60	91.48	89.74
GECOR1 †	65.26	91.82	38.90	83.09	95.59	97.73	93.54
GECOR2 †	64.09	89.43	37.30	82.83	96.00	97.95	94.12
Baseline	55.85	77.54	31.43	76.87	91.88	93.11	90.68
Final	68.91	94.57	40.00	84.34	95.81	97.67	94.02
w/o SH-DHE	65.64	92.39	35.51	82.76	95.50	97.61	93.48
w/o TDH-CM	57.58	80.43	31.84	78.28	92.36	93.47	91.28

Table 5. Results of the split data in short and long dialogue history.

Data	Model	EM	EM1	EM2	BLEU	F1	Prec.	Rec.
Short	GECOR2 †	59.79	88.36	32.66	82.61	95.72	97.98	93.56
	Ours	62.18	90.43	35.35	81.86	95.59	97.99	93.31
Long	GECOR2 †	77.27	90.82	38.24	82.85	95.24	98.78	91.95
	Ours	81.95	96.00	39.39	85.86	96.13	97.94	94.39

3.4 Ablation Study

Effects of the SH-DHE and TDH-CM Methods. Although the baseline system has a good performance in capturing the global information of dialogue history when compared with GECOR-basic, it still suffers from the long distance of the dialogue history. To tackle this issue, we propose to model the dialogue history at multiple levels during encoding and decoding, and the resulting system

can extract much richer semantic information from the dialogue history. To show the effectiveness of our approaches, we conduct an ablation study on different ways of combining the three kinds of datasets. As shown in columns 5, 6, 7, and 8 of Tables 2, 3, and 4, the Baseline system enhanced with the SH-DHE and TDH-CM modules can significantly improve the performance on all the seven metrics. With this in mind, we intuitively believe that the SH-DHE module integrates the potential information of dialogue interaction into the global representation of dialogue history, and also enhances the model’s ability to distinguish the information of different speakers. Moreover, since the TDH-CM module could excavate the multi-granularity information in the dialogue history, the semantic expressions related to the omitted and coreferred information can be accurately highlighted.

Additionally, comparing the Baseline with TDH-CM we find that combining the SH-DHE and TDH-CM modules can increase the EM and BLEU scores by 1.07 points and 1.34 points respectively on the ellipsis dataset, 1.93 points and 0.63 points respectively on the coreference dataset, and 3.27 points and 1.58 points respectively on the mixed dataset. The results indicate that our proposed approach does enhance the system’s capability of modeling dialogue history and capturing the missing semantic information accurately.

Performance on Long Dialogue History Understanding. In this paper, we explicitly argue that our approaches enable the model to learn better representation of the long dialogue history and capture the omitted or coreferred information from the history expressions. To prove this, we present the performance of our model on short and long dialogs for comparison. According to our statistics, each utterance contains around 12.51 tokens and each dialogue in CamRest676 consists of about 4 dialog turns. On this basis, we divide the ellipsis dataset according to the number of turns in the dialogue history, where the texts with less than 4 dialogue turns are regarded as short dialogs, and vice versa as long dialogs. The results in Table 5 show that our proposed approach can bring significant performance improvements to the GECOR2 system when the dialogue history is longer than 4 dialog turns, which suggests the great effectiveness

	I	'm	looking	for	a	place	to	eat	in	the	east	side	of	the	city	.		
Q1	3.26	3.74	3.51	2.97	2.41	2.43	1.51	0.41	0.87	0.82	1.70	2.84	1.14	0.86	1.31	0.34		
A1	The Curry Prince	is on	the	east	side	of town	.											
1.90	4.13	3.65	2.78	1.49	1.78	2.47	0.31	0.03	0.45									
Q2	Does the Yu Garden serve	chinese	food	?														
2.34	1.47	4.97	3.23	1.98	2.56	1.39												
A2	No they do not	serve	Chinese	food	,	however	the	Yu Garden	does	and	is	in	the	east	side	.		
4.16	3.19	2.18	2.27	2.48	1.85	3.25	3.56	2.87	2.22	5.28	4.09	2.67	1.90	1.13	0.67	1.22	2.16	0.67
Q3	What is the address	?																
2.77	1.77	1.21	1.39	0.57														
A3	Yu Garden 's address	is	address_SLOT	.	Is there anything else I	can help you with today	?											
5.49	5.01	3.97	3.37	1.84	2.62	3.07	3.22	3.93	3.24	2.99	2.18	2.23	1.49	1.84	0.61	0.57		

Incomplete utterance Q4: Yes , what is the price range (**of the Yu Garden**) ?

Fig. 2. Heatmap visualization of TDH-CM for examples in Table 1. The higher attention weights are colored in red, and others are colored in blue. The right pattern shows the distribution of sentence-level attention and the left pattern shows the distribution of word-level attention. (Color figure online)

of our approach in long-distance history modeling. By contrast, our performance improvement is limited when the dialog is short, which explains why our system obtains similar results as GECOR2 on some metrics.

4 Case Study

To demonstrate the interpretability of our model, we give the heatmap visualization of the TDH-CM for the examples in Table 1, as shown in Fig. 2. In the example, ‘*of the Yu Garden*’ is the omitted expression in the incomplete utterance Q4. Referring to the sentence-level attention on the right, the utterance that contains omitted and coreferred information receives more attention, i.e., A3, A2, and Q2. We can also see that the TDH-CM method ignores Q2 and pays more attention to A3. In other words, the TDH-CM method can detect both short and long distance dependencies, and the utterances in dialogue history that are close to the incomplete utterance obtain more attention.

Similarly, the word-level attention models the interaction between the tokens in dialogue history and the omitted and coreferred expressions in the incomplete utterance. In this example, the model correctly distributes higher attention weights to the omitted tokens, such as ‘*Yu Garden*’ in Q2 with the score of 5.28 and ‘*Yu Garden*’ in A3 with the score of 5.49. Hence, we have reason to believe that the TDH-CM method we use is stable and useful for dialogue ellipsis and coreference resolution in multi-turn dialogue.

5 Related Work

Previous work on English ellipsis recovery focus on verb phrase, and scholars established a theoretical system by analyzing linguistic rules. Nielsen et al. [8] first proposed an end-to-end computable system based on machine learning techniques. Drawing lessons from Nielsen et al. [8], Liu et al. [7] introduced a unified framework that combines target detection, antecedent head resolution, and antecedent boundary detection. Our work is closely related to the coreference resolution task, which aims at clustering mentions that refer to the same physical entities. Lee et al. [5] presented a fully differentiable approximation to high-order inference based on end-to-end coreference resolution model proposed by Lee et al. [4]. Joshi et al. [2] used unsupervised contextualized representations to enhance the coreference resolution. Wu et al. [13] first formulated coreference resolution as a machine reading task.

Until recently, there exists researches on dialogue ellipsis and coreference resolution. Kumar et al. [3] used the framework of sequence-to-sequence learning to generate complete questions from a non-sentential question, given previous question and answer. Su et al. [12] introduced a Transformer-based utterance rewriting architecture using the pointer network to recover all coreferred and omitted information. Quan et al. [9] first attempted to provide both solution and dataset for ellipsis and coreference resolution in multi-turn dialogue.

Nevertheless, one drawback is that almost all previous works are limited to short texts or one-shot dialogues and treat all dialogue history utterances indiscriminately, which is not proper for the dialogue ellipsis and coreference resolution task. Since the omitted and coreferred tokens usually related to a few previous dialogue histories. In recent years, several work provided studies on context to improve the performance of dialogue systems and response selection tasks (Xing et al. [14], Zhang et al. [15], Shan et al. [10]). The motivation of this paper is how to effectively extract and aggregate the relevant expressions in dialogue history. Different from previous work, our model can enhance long-distance dialogue history modeling and focus on the relevant expressions at multiple levels.

6 Conclusion

In this paper, we proposed a baseline model to separately encode the utterances in the long-distance history of multi-turn dialogue. Based on it, we used a speaker highlight dialogue history encoder and a top-down hierarchical copy mechanism to well capture the omitted and coreferred information in dialogue history at multiple levels. Experimental results demonstrated that our approach can significantly improve both dialogue ellipsis and coreference resolution quality in long-distance multi-turn dialogue. Compared with previously proposed methods, our resulting model is much more competitive. And we will extend our proposed model on other domains such as human-to-human conversations or Chinese multi-turn dialogue corpus in future work.

Acknowledgments. The authors would like to thank the anonymous reviewers for the helpful comments. We also thank Longyin Zhang and Jinfeng Wang for their helpful discussions. This work was supported by Project 61876118 under the National Natural Science Foundation of China and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

- Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: Proceedings of the 3rd International Conference on Learning Representations (2015)
- Joshi, M., Chen, D., Liu, Y., Weld, D.S., Zettlemoyer, L., Levy, O.: SpanBERT: improving pre-training by representing and predicting spans. Trans. Assoc. Comput. Linguist. **8**, 64–77 (2020)
- Kumar, V., Joshi, S.: Non-sentential question resolution using sequence to sequence learning. In: Proceedings of COLING 2016, pp. 2022–2031 (2016)
- Lee, K., He, L., Lewis, M., Zettlemoyer, L.: End-to-end neural coreference resolution. In: EMNLP (2017)
- Lee, K., He, L., Zettlemoyer, L.: Higher-order coreference resolution with coarse-to-fine inference. arXiv preprint [arXiv:1804.05392](https://arxiv.org/abs/1804.05392) (2018)
- Liu, Q., Chen, B., Lou, J.G., Zhou, B., Zhang, D.: Incomplete utterance rewriting as semantic segmentation. arXiv preprint [arXiv:2009.13166](https://arxiv.org/abs/2009.13166) (2020)

7. Liu, Z., Gonzalez, E., Gillick, D.: Exploring the steps of verb phrase ellipsis. In: Proceedings of NAACL (2016)
8. Nielsen, L.A.: A corpus-based study of verb phrase ellipsis. In: Proceedings of the 6th Annual CLUK Research Colloquium, pp. 109–115 (2003)
9. Quan, J., Xiong, D., Webber, B., Hu, C.: GECOR: an end-to-end generative ellipsis and co-reference resolution model for task-oriented dialogue. arXiv preprint [arXiv:1909.12086](https://arxiv.org/abs/1909.12086) (2019)
10. Shan, Y., et al.: A contextual hierarchical attention network with adaptive objective for dialogue state tracking. arXiv preprint [arXiv:2006.01554](https://arxiv.org/abs/2006.01554) (2020)
11. Shi, Z., Huang, M.: A deep sequential model for discourse parsing on multi-party dialogues. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 7007–7014 (2019)
12. Su, H., et al.: Improving multi-turn dialogue modelling with utterance rewriter. arXiv preprint [arXiv:1906.07004](https://arxiv.org/abs/1906.07004) (2019)
13. Wu, W., Wang, F., Yuan, A., Wu, F., Li, J.: CorefQA: coreference resolution as query-based span prediction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6953–6963 (Jul 2020)
14. Xing, C., Wu, Y., Wu, W., Huang, Y., Zhou, M.: Hierarchical recurrent attention network for response generation. In: Proceedings of the AAAI Conference on Artificial Intelligence (2018)
15. Zhang, H., Lan, Y., Pang, L., Guo, J., Cheng, X.: ReCoSa: detecting the relevant contexts with self-attention for multi-turn dialogue generation. arXiv preprint [arXiv:1907.05339](https://arxiv.org/abs/1907.05339) (2019)
16. Zhang, X., Li, C., Yu, D., Davidson, S., Yu, Z.: Filling conversation ellipsis for better social dialog understanding. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 9587–9595 (2020)



Exploiting Explicit and Inferred Implicit Personas for Multi-turn Dialogue Generation

Ruifang Wang^{1,2}, Ruifang He^{1,2(✉)}, Longbiao Wang^{2(✉)}, Yuke Si²,
Huanyu Liu², Haocheng Wang², and Jianwu Dang^{2,3}

¹ State Key Laboratory of Communication Content Cognition, People's Daily Online, Beijing, China

² Tianjin Key Laboratory of Cognitive Computing and Application, College of Intelligence and Computing, Tianjin University, Tianjin, China

{ruifang_wang,rfhe,longbiao_wang,siyuke,huanyuliu,haochengwang}@tju.edu.cn

³ Japan Advanced Institute of Science and Technology, Ishikawa, Japan
jdang@jaist.ac.jp

Abstract. Learning and utilizing personas in open-domain dialogue have become a hotspot in recent years. The existing methods that only use predefined explicit personas enhance the personality to some extent, however, they cannot easily avoid persona inconsistency and weak diversity responses. To address these problems, this paper proposes an effective model called Exploiting Explicit and Inferred Implicit Personas for Multi-turn Dialogue Generation (**EIPD**). Specifically, 1) an explicit persona extractor is designed to improve persona consistency; 2) Taking advantage of the von Mises-Fisher (**vMF**) distribution in modeling directional data (e.g., the different persona state), we introduce the implicit persona inference to increase diversity; 3) during the generation, the persona response generator fuses the explicit and implicit personas in the response. The experimental results on the ConvAI2 persona-chat dataset demonstrate that our model performs better than commonly used baselines. Further analysis of the ablation experiments shows that EIPD can generate more persona-consistent and diverse responses.

Keywords: Persona-based dialogue generation · Implicit personas · vMF

1 Introduction

With the development of open-domain dialogue system, great progress has been achieved in many fields, such as intelligent assistants, customer service and chatbots [3]. The end-to-end network [14] has been proven effective for generative dialogue systems. However, it is still difficult to build more engaging and realistic conversations owing to the lack of interlocutor personas.

Several efforts have been made to explore the abilities of personas for facilitating response generation [7]. [20] introduced a novel dataset, PERSONA-CHAT,

Personas	
1. <i>My skin is olive colored.</i> 3. <i>My eyes are green.</i> 5. <i>I want to be a librarian.</i>	2. My purse has a picture of a skunk on it. 4. <i>I wear glasses that are cateye.</i>
Dialogue1 Context1: Hi, how are you? please tell me about yourself ! Context2: Hello, blonde hair, blue eyes. and yourself ? Response: Yes , I've olive skin color with green eyes and cateye glasses.	Dialogue2 Context1: You sound very pretty ! what else do you like ? Response: Books ! which probably explains why I'm studying to become a librarian. you ?

Fig. 1. The two dialogues from ConvAI2 persona-chat, where the same colors of sentences imply that the sentences are related to each other in the conversation.

where each dialogue is assigned a character description using 5 sentences as a persona profile. We define these persona profiles as **explicit personas**. Then, [12, 13] generated responses with this kind of personas. However, in real conversations, sometimes the repliers answer with explicit personas directly, and sometimes they answer with some useful information that can be inferred from the explicit personas and context, which we define as **implicit personas**. Specifically, as shown in Fig. 1, the two dialogues are associated with the same explicit personas. The response in Dialogue1 is directly associated with explicit personas, ‘*My skin is olive colored. My eyes are green. I wear glasses that are cateye.*’. It describes the image of the speaker which are consistent with the context. Therefore, how to capture context-relevant personas is essential in persona-based dialogue. However, in Dialogue2, the response not only mentions the persona ‘*I want to be a librarian.*’ but also explain the reason why the speaker wants to be a librarian. This kind of information does not appear in the context and explicit personas, but it can be inferred from persona-based context. This indicates it is possible to use implicit personas in some responses. Although some persona-based dialogue methods have been proposed, the following challenges still exist: 1) In multi-turn dialogue, as shown in Fig. 1, the response is related to some contextual personas, and previous methods cannot effectively capture the key explicit personas, which is not conducive to persona consistency. 2) In the persona-based dialogue, the attractive responses are not only persona-consistent but also diverse, while the existing methods mainly focus on persona consistency. 3) Previous methods usually take explicit personas into consideration, but neglect that both explicit and implicit personas mentioned above can interact with each other in one model at the same time.

To tackle these challenges, we propose a model called Exploiting Explicit and Inferred Implicit Personas for Multi-turn Dialogue Generation (EIPD), which consists of three components. Specifically, the explicit persona extractor mainly adopts a transformer encoder to acquire some explicit personas relevant to the context. Second, the implicit persona inference module employs the von Mises-Fisher (vMF) distribution, which is suitable for modeling directional data to

reason the implicit personas and improve the response diversity. Third, the persona-response generator is designed to guide the implicit personas and fuse the two kinds of personas to generate the response. Finally, the ConvAI2 persona-chat dataset is used to evaluate the effectiveness of proposed model. We summarize the contributions of this work as follows:

- It is the first time that an effective framework for multi-turn dialogue generation takes two kinds of personas into consideration simultaneously.
- An implicit personas inference module with an vMF distribution is devised to reason the implicit personas.
- The persona generator is used to supervise the generation of implicit personas.
- The experimental results demonstrate that our model can generate responses with more diversity and persona consistency compared with baseline results.

2 Related Work

2.1 Persona-Based Dialogue Model

In open-domain dialogue generation, the persona-based dialogue model has attracted an increasing number of researchers' attention. Recent works focus on improve the persona-based dialog generation performance as well as persona consistency. [11] assigned a desired identity to chatbot which can generate coherent response. [20] constructed a persona-chat dataset with different speaker profiles. Based on this dataset, [13] proposed an Reinforcement Learning framework to improve persona consistency of response. Besides these works using speaker profiles, other works using implicit information to achieve it. [7] used pretrained speaker embeddings and dialogue context to boost informative and diverse response. [10] proposed a multi-task learning approach that incorporated speaker characteristics to train the neural conversation models. Despite the success of using implicit persona in conversation, they are still difficult to learn implicit personas displayed by the speakers automatically.

2.2 von Mises-Fisher Distribution

The von Mises-Fisher(vMF) distribution represents a latent hyperspherical space which can model directional data better. Considering this characteristic, the vMF distribution is introduced into some NLP works. Both [1] and [9] integrated vMF into a topic model to explore the semantic consistency and to improve the performance. [18] replaced Gaussian distribution with vMF distribution in CVAE and discovered that the ‘collapse’ problem can also be alleviated. [5] used vMF distribution to draw the context word vectors to improve the embedding models. Different from these works, we apply vMF distribution in the Conditional Variational Autoencoder(CVAE) framework to infer the implicit personas.

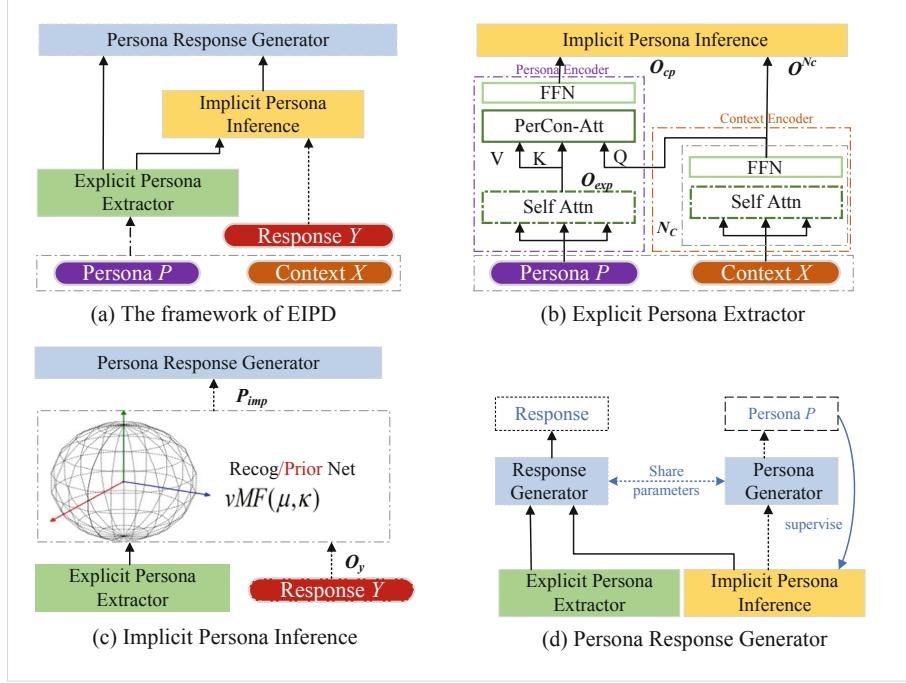


Fig. 2. The framework of the EIPD model, including explicit persona extractor, implicit persona inference and persona response generator. The process represented by the dotted line only occurs during the training.

3 The Proposed Model

A persona-based dialogue system generates responses with context and personas. Our problem is formulated as follows: context $X = \{x_1, x_2, \dots, x_m\}$, each utterance $x_i = (w_{i,1}^x, w_{i,2}^x, \dots, w_{i,M_i}^x)$, a set of explicit personas $P_{exp} = \{p_1, p_2, \dots, p_n\}$, each persona $p_i = (w_{i,1}^p, w_{i,2}^p, \dots, w_{i,N_i}^p)$, and response $Y = \{w_1^y, w_2^y, \dots, w_k^y\}$. Given X , the implicit personas P_{imp} are explored by the implicit persona inference module with the supervision of explicit personas. By leveraging the context, explicit personas, and implicit personas, the goal is to generate a diverse and persona-consistent response Y . We drop the subscript of P_{exp} for simplicity.

As shown in Fig. 2(a), the whole framework can be divided into three modules: (1) Explicit Persona Extractor, (2) Implicit Persona Inference, and (3) Persona Response Generator.

3.1 Explicit Persona Extractor

Following Transformer [15], this component (Fig. 2(b)), which includes a context encoder and a persona encoder, takes context and explicit personas as the input and extracts the most relevant explicit personas to improve persona consistency.

Context Encoder: We use the transformer encoder to encode the context X . The multi-head self-attention is defined as $\text{MultiHead}(Q, K, V)$, where Q , K , and V represent query, key, and value, respectively. The encoder is composed of N_c layers. The encoding of context is as follows:

$$H_c^n = \text{MultiHead}(O_c^{n-1}, O_c^{n-1}, O_c^{n-1}) \quad (1)$$

$$O_c^n = \text{FFN}(H_c^n) \quad (2)$$

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (3)$$

where $n \in (2, N_c)$. H_c^n and O_c^n are the n -th layer output of the multi-head self-attention and feed-forward network, respectively. In the first layer, O_c^1 represents the word embedding and positional embedding of the input. Following [15], we also add layer normalization to the sub layers, and we can finally obtain the context representation O^{N_c} after N_c layers.

Persona Encoder: According to the examples in Fig. 1, we observe the following: 1) The response Y is often related to some personas p_i and contexts X_j . 2) The relevance between X and p_i is beneficial to generate an informative and consistent response. Therefore, we want to consider them. Specifically, we use another multi-head self-attention to encode the explicit personas. O_{exp} represents the output of this attention mechanism. We then use PerCon-Attention which takes O^{N_c} as query, O_{exp} as key and value to compute the contextual explicit persona hidden vector O_{cp} based on the following equations:

$$H_{cp} = \text{PerConAtt}(O^{N_c}, O_{exp}, O_{exp}) \quad (4)$$

$$O_{cp} = \text{FFN}(H_{cp}) \quad (5)$$

3.2 Implicit Persona Inference

According to Fig. 1, we can see that the personas shown in the response are not entirely extracted from the given explicit personas. We therefore employ an inference module using vMF distribution to reason the implicit personas (Fig. 2(c)) for the personalized and diverse responses.

Since different speakers express different implicit personas, this information can be represented in different directions in the semantic space. The vMF distribution [18] can model directional data better, therefore, we introduce it into the CVAE framework. Specifically, in the CVAE framework, the prior network $p_\theta(z|X, P)$ and the recognition network $q_\varphi(z|X, P, Y)$ are used to sample the latent variable z , namely, implicit personas, and can be written as p_{imp} . In our settings, p_{imp} follows the vMF distribution, specifically the prior network $p_\theta(p_{imp}|X, P) \sim vMF(\mu_{prior}, \kappa_{prior})$ and the posterior network $q_\varphi(p_{imp}|X, P, Y) \sim vMF(\mu_{pos}, \kappa_{pos})$.

VMF Distribution: The von Mises-Fisher distribution is defined over a hypersphere of unit norm, depending on the direction vector $\mu \in R^m$ with $\|\mu\|=1$ and a concentration parameter $\kappa \in R_{\geq 0}$, where m denotes the dimension of the word vectors. The Probability Density Function of the vMF distribution for a random unit vector $z \in R^m$ is defined as:

$$f_m(p_{imp}; \mu, \kappa) = C_m(\kappa) \exp(\kappa \mu^T p_{imp}) \quad (6)$$

$$C_m(\kappa) = \frac{\kappa^{m/2-1}}{(2\pi)^{m/2} I_{m/2-1}(\kappa)} \quad (7)$$

where $C_m(\kappa)$ is the normalization constant and $I_{m/2-1}$ stands for the modified Bessel function of the first kind at order v . Inspired by NVSRN [2], we encode Y into representations O_y , set κ_{prior} and κ_{pos} as constants and compute μ_{prior} , μ_{pos} as:

$$\mu_{pos}^{\sim} = f_{pos}([O^{N_c}, O_{cp}, O_y]) \quad (8)$$

$$\mu_{pos} = \mu_{pos}^{\sim} / \|\mu_{pos}^{\sim}\| \quad (9)$$

$$\mu_{prior}^{\sim} = f_{prior}([O^{N_c}, O_{cp}]) \quad (10)$$

$$\mu_{prior} = \mu_{prior}^{\sim} / \|\mu_{prior}^{\sim}\| \quad (11)$$

where f_{prior} and f_{pos} are two transformations and $\|\cdot\|$ denotes the 2-norm used to ensure the normalization. Since the prior $p_{\theta}(p_{imp}|X, P)$ follows the $vMF(\mu_{prior}, \kappa_{prior})$ rather than $vMF(\cdot, 0)$, the KL divergence will be computed as:

$$\begin{aligned} \mathcal{L}_{KL} &= KL(q_{\varphi}(p_{imp}|X, Y, P) || p_{\theta}(p_{imp}|X, P)) \\ &= (m/2 - 1) \log \frac{\kappa_{pos}}{\kappa_{prior}} + \log \frac{I_{m/2-1}(\kappa_{prior})}{I_{m/2-1}(\kappa_{pos})} \\ &\quad - \kappa_{prior} \mu_{prior} \mu_{pos}^{-1} \frac{I_{m/2}(\kappa_{pos})}{I_{m/2-1}(\kappa_{prior})} + \kappa_{pos} \frac{I_{m/2}(\kappa_{pos})}{I_{m/2-1}(\kappa_{prior})} \end{aligned} \quad (12)$$

Sampling Technique for vMF: Following the implementation of [4], we use the rejection sampling scheme to sample $w \in [-1, 1]$, and then the latent variable p_{imp} is derived from $p_{imp} = w\mu + v\sqrt{1-w^2}$, where v is a randomly sampled unit vector tangent to the hypersphere at μ .

3.3 Persona Response Generator

This component comprises a response generator and a persona generator (Fig. 2(d)). Considering the interaction between the two kinds of personas, we use the two generators to further enhance the modeling of directional data and better fuse the implicit and explicit personas.

Persona Generator: To strengthen the supervision for implicit personas, during this process, we employ an RNN decoder that receives implicit persona p_{imp} as the initial hidden state and then generates tokens sequentially under the probability distributions:

$$p_{\theta_p}(P|p_{imp}) = \prod_i^n \prod_{j=1}^{N_i} p(w_{i,j}|P_{<i}, w_{i,<j}) \quad (13)$$

where n is the number of turns of explicit personas; N_i is the length of the i -th utterance p_i . During this process, the loss function is:

$$\mathcal{L}_p = \mathbf{E}_{q_\varphi(p_{imp}|X, P, Y)} [\log p_{\theta}(P|p_{imp})] \quad (14)$$

Response Generator: Finally, conditioned based on explicit personas, implicit personas and context, we employ a response decoder to generate the response Y :

$$p_{\theta_g}(Y|X, P, p_{imp}) = \prod_{i=1}^k p_{vocab}(w_{y,i}) \quad (15)$$

where p_{vocab} is the vocabulary's probability distribution; $p_{vocab}(w_{y,i})$ is the probability of the word $w_{y,i}$; k is the length of the response Y . In general, the ELBO in the decoder can be rewritten as:

$$\mathcal{L}_r = \mathbf{E}_{q_\varphi(p_{imp}|X, Y, P)} [\log p_{\theta}(Y|p_{imp}, X, P)] - \mathcal{L}_{KL} \quad (16)$$

3.4 Training Objective

In the EIPD model, the overall objective is:

$$\mathcal{L} = \lambda \mathcal{L}_p + (1 - \lambda) \mathcal{L}_r \quad (17)$$

where the hyperparameter λ is used to control the balance between response generator and persona generator.

4 Experiments

4.1 Experimental Settings

Dataset: We use the released ConvAI2 persona-chat dataset, which is an extended version of PERSONA-CHAT [20]¹, to verify our proposed method. The dataset consists of 164,356 utterances in 10,981 dialogues, and each speaker has at least 4 persona profiles. We randomly split the data into the training, validation, and test sets, which respectively contain 67112, 8395, and 4478 dialogues.

¹ <http://convai.io/>.

Baselines: We compared the proposed EIPD model with five commonly used baseline models. **S2SAP**: the Seq2Seq model, which integrates context and persona as the input [20]. **CVAE**²: an RNN-based model that exploits latent variables to improve the diversity of the response [21]. **Trans**³: the transformer model [15] that concatenates personas and context as the input. **PerCVAE**⁴: a memory augmented CVAE model that uses multi-hop attention to exploit the persona information to improve the response quality [12]. **TransferTransfo**⁵: a finetuned GPT2 that takes personas and dialogue context as the input [16] (Table 1).

Table 1. Objective (on the left) and subjective evaluation (on the right) results with respect to the ConvAI2 persona-chat dataset. Results in bold represent the best scores. In the subjective evaluations, the percentages of each kind of response are calculated by combining the evaluations from three annotators together. The Kappa scores of all models are higher than 0.4, which indicates that the three annotators reach a fair agreement.

Model	Dist-1	BLEU-1	BLEU-2	F1	G1	G2	G3	G4	G3&4
S2SAP	0.0151	0.1467	0.1439	0.2309	38.25	29.67	27.35	4.73	32.08
CVAE	0.0165	0.1356	0.1502	0.1903	37.25	25.00	26.00	11.75	37.75
Trans	0.0267	0.1531	0.1621	0.1921	32.25	25.50	29.00	13.25	44.25
PerCVAE	0.0374	0.2047	0.1858	0.2404	21.43	18.73	39.25	20.59	59.84
TransferTransfo	0.0332	0.2532	0.2249	0.1973	20.34	17.14	42.73	19.79	62.52
EIPD	0.0388	0.2263	0.2323	0.2452	18.36	14.75	44.75	22.14	66.89

Parameters: For the RNN-based models, we set word embeddings to the size of 300. The encoder is a 2-layer GRU structure with a hidden size of 600. For the Transformer, the size of word embedding is set to 512, and the numbers of layers of encoder and decoder are set to 3 and 1. Besides, the number of heads in multi-head attention is 8, and the inner-layer size of the feed-forward network is 2048. In our model, the parameters of the explicit persona extractor are the same as those of Transformer. The dimension of the latent variable is set to 180. We use the Adam algorithm to update the parameters with a learning rate of 0.0001. The batch size is set to 32. An early-stop strategy is used to obtain the best model. Our model is implemented using the Tensorflow framework. We conduct all experiments on a GPU.

Evaluations: In our experiments, we use Dist-1, BLEU-1/2 and F1 to evaluate our method. In addition to the automatic metrics, we recruit three human annotators familiar with the NLP tasks to judge the quality of the generated

² <https://github.com/snakeztc/NeuralDialog-CVAE>.

³ <http://github.com/atselousov/transformerchatbot>.

⁴ <https://github.com/vsharecodes/percvae>.

⁵ <http://github.com/huggingface/transfer-learning-conv-ai>.

responses. We sampled 200 context-response-persona triples from the above models. They are required to provide 4-graded judgements according to the following criteria: **G1**: The generated response is not grammatically correct, is irrelevant to the semantics of context or is inconsistent with the given personas. **G2**: The generated response is fluent and weakly related to the context, such as some generic responses. **G3**: The generated response is fluent and relevant to the context semantics and slightly consistent with the personas. **G4**: The generated response is not only fluent and semantically relevant but also consistent with the given personas.

Table 2. Performances of model ablation. EIPD is significantly better than the ablation approaches.

Model	Dist-1	BLEU-1	BLEU-2	F1
D	0.0007	0.1248	0.1365	0.2038
IPD	0.0301	0.1465	0.1526	0.2186
EPD	0.0354	0.2142	0.2053	0.2439
EIPD _{Gau}	0.0345	0.2171	0.2064	0.2348
EIPD _{pd}	0.0363	0.2121	0.2105	0.2208
EIPD	0.0388	0.2263	0.2323	0.2452

4.2 Experimental Results

Objective and Subjective Evaluations: For objective evaluation, (1) Dist-1 is the ratios of distinct unigrams which can reflect the diversity of the generated response. It can be found that the performance of S2SAP is the worst because it only roughly combines the explicit personas. PerCVAE surpassed other baselines due to the exploitation of explicit personas. Compared with the baselines, EIPD outperforms them, which indicates that the proposed model can generate diverse responses. (2) BLEU-1/2 evaluates how many n-grams ($n = 1, 2$) in the generated responses overlap with them in the ground truth. EIPD performs better than baselines except for TransferTransfo in BLEU-1, and we speculate that the reason may be that the pretrained language model contains semantic information. (3) For F1, the score of EIPD is higher than others, demonstrating that the model can generate more accurate information.

For subjective evaluation, the responses generated by EIPD are more engaging as compared to the responses from all baselines. It can be determined that the percentage of diverse and persona-consistent responses (the grade ‘G3&4’) is 66.89%, obviously higher than others, which indicates that EIPD can generate persona-consistent responses. Additionally, the percentage of ‘G2’ is declining, while, the percentage of ‘G3’ is rising. This proves that EIPD has the ability to generate context-relevant responses, and alleviate the problem of generic responses at the same time. Among the baselines, the results of S2SA perform poorly since it the model does not take any kind of personas into consideraiton.

By adding explicit personas or global information, the performance of these models improve gradually, yet still worse than our model.

Ablation Analysis: To investigate the effects of specific modules in EIPD, we ablated our model through several different approaches: **D**: A generative dialog model without explicit and implicit personas. **EPD**: It removes the implicit persona inference, that is, the model does not use implicit personas. **IPD**: It replaces the explicit persona extractor with the RNN to represent the explicit personas. **EIPD_{Gau}**: This model replaces the vMF distribution with the Gaussian distribution. **EIPD_{pd}**: This approach deletes the persona generator, so the generation of implicit personas loses the supervision of the explicit personas.

As shown in Table 2, from D, IPD, EPD, EIPD_{Gau}, and EIPD_{pd} to EIPD, every step yields an observed improvement on the automatic metrics. EIPD achieves the best performance among all the methods. Specifically, compared with D, the improvements of EPD and IPD on all metrics imply that the explicit persona extractor can capture the explicit personas related to some context, and the implicit persona inference module can obtain the implicit personas inferred from the given context and explicit personas. Furthermore, we note that EIPD performs better than EIPD_{Gau} on all metrics, which proves that the vMF distribution is more useful than the Gaussian distribution in this framework. Specifically, the implicit persona inference module can reason the more rational implicit personas with vMF distribution, and this phenomenon is consistent with the characteristics of vMF, which is good at modeling directional data, such as the personalities of different speakers. In addition, the performance of EIPD_{pd} is inferior to EIPD, which verifies that the persona generator can facilitate the generation of persona-consistent and diverse responses.

Table 3. An example of dialogue with the personas ‘Black coffee is my addiction. My favorite hobby is gardening. My family gets together every Saturday. My husband died last year.’ in ConvAI2 persona-chat dataset.

Context

A>Hello how are you doing?

B:I’m good how are you?

A:Good thanks. So what is life like for you?

SASP:I’m a student and I work a lot and a lot.

CVAE:I am, as little since I am super excited about me, school.

Trans:I like to play games with friends what about you?

PerCVAE:I would like to talk about you. Would you like a happy person?

TransferTransfo:That’s so good, I prefer to read.

Golden:I am currently struggling in school.

EIPD-1:It is ok. I do not get much done unless I work on my garden.

EIPD-2:It is ok now. I like to walk outside and explore the outdoors.

Case Study: According to Table 3, we can determine that the baseline models often generate some fluent but irrelevant and weak personalized responses. For comparison, we use the EIPD to generate different responses through implicit persona inference, and we find that the responses are related to the personas ‘*My favorite hobby is gardening*’. The first response directly answers the speaker’s attitude about *gardening*, and the second response expands the information about the given personas.

5 Conclusion and Future Work

In this paper, we propose an effective EIPD for multi-turn persona-based dialogue. To the best of our knowledge, we are the first to fuse the explicit personas and implicit personas to generate more realistic responses. It uses an explicit persona extractor to improve the persona consistency, and employs an implicit persona inference module with vMF distribution to improve the diversity. Finally, the persona response generator is used to fuse personas and generate the response. Experimental results on ConvAI2 persona-chat dataset demonstrate the effectiveness of our model and verify the importance of implicit personas. In the future, we would like to use knowledge graphs and pretrained language model to strengthen the inference of implicit personas.

Acknowledgement. This work was supported by the National Key RD Program of China under Grant 2018YFB1305200, the National Natural Science Foundation of China under Grant (61771333, 61976154), the Tianjin Municipal Science and Technology Project under Grant 18ZXZNGX00330, and the State Key Laboratory of Communication Content Cognition, People’s Daily Online (No. A32003).

References

1. Batmanghelich, K., Saeedi, A., Narasimhan, K., Gershman. S.: Nonparametric spherical topic modeling with word embeddings. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp. 537–542 (2016)
2. Chang, J., et al.: NVSRN: a neural variational scaling reasoning network for initiative response generation. In: 2019 IEEE International Conference on Data Mining, pp. 51–60 (2019)
3. Chen, H., Liu, X., Yin, D., Tang, J.: A survey on dialogue systems: recent advances and new frontiers. ACM SIGKDD Explor. Newslett. **19**(2), 25–35 (2017)
4. Guu, K., Hashimoto, T.B., Oren, Y., Liang, P.: Generating sentences by editing prototypes. Trans. Assoc. Comput. Linguist. **6**, 437–450 (2018)
5. Jameel, S., Schockaert, S.: Word and document embedding with vMF-mixture priors on context word vectors. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 3319–3328 (2019)
6. Kingma, D., Mohamed, S., Jimenez, D., Welling, M.: Semi-supervised learning with deep generative models. In: Advances in Neural Information Processing Systems, pp. 3581–3589 (2014)

7. Kottur, S., Wang, X., Carvalho, V.: Exploring personalized neural conversational models. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-2017, pp. 3728–3734 (2017)
8. Li, J., Galley, M., Brockett, C., Spithourakis, G.P., Gao, J., Dolan, B.: A persona based neural conversation model. arXiv preprint [arXiv:1603.06155](https://arxiv.org/abs/1603.06155) (2016)
9. Li, X., Chi, J., Li, C., Ouyang, J., Fu, B.: Integrating topic modeling with word embeddings by mixtures of vMFs. In: COLING 2016, 26th International Conference on Computational Linguistics, pp. 151–160 (2016)
10. Luan, Y., Brockett, C., Dolan, B., Gao, J., Galley, M.: Multi-task learning for speaker-role adaptation in neural conversation models. In: Proceedings of the Eighth International Joint Conference on Natural Language Processing, pp. 605–614 (2017)
11. Qian, Q., Huang, M., Zhao, H., Xu, J., Zhu, X.: Assigning personality identity to a chatting machine for coherent conversation generation. arXiv preprint [arXiv:1706.02861](https://arxiv.org/abs/1706.02861) (2017)
12. Song, H., Zhang, W., Cui, Y., Wang, D., Liu, T.: Exploiting persona information for diverse generation of conversational responses. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-2019, pp. 5190–5196 (2019)
13. Song, H., Zhang, H., Hu, J., Liu, T.: Generating persona consistent dialogues by exploiting natural language inference. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, pp. 8878–8885 (2020)
14. Sutskever, I., Vinyals, O., Le, Q.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, pp. 3104–3112 (2014)
15. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
16. Wolf, T., Sanh, V., Chaumond, J., Delangue, C.: Transfertransfo: a transfer-learning approach for neural network based conversational agents. arXiv preprint [arXiv:1901.08149](https://arxiv.org/abs/1901.08149) (2019)
17. Wu, B., et al.: Guiding variational response generator to exploit persona. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 53–65 (2020)
18. Xu, J., Durrett, G.: Spherical latent spaces for stable variational autoencoders. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 4503–4513 (2018)
19. Yao, K., Zweig, G., Peng, B.: Attention with intention for a neural network conversation model. arXiv preprint [arXiv:1510.08565v3](https://arxiv.org/abs/1510.08565v3) (2015)
20. Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., Weston, J.: Personalizing dialogue agents: I have a dog, do you have pets too? arXiv preprint [arXiv:1801.07243](https://arxiv.org/abs/1801.07243) (2018)
21. Zhao, T., Zhao, R., Eskenazi, M.: Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 654–664 (2017)



Few-Shot NLU with Vector Projection Distance and Abstract Triangular CRF

Su Zhu¹, Lu Chen^{2(✉)}, Ruisheng Cao², Zhi Chen², Qingliang Miao¹, and Kai Yu^{1,2(✉)}

¹ AISpeech Co., Ltd., Suzhou, China

su.zhu@aispeech.com

² X-LANCE Lab, Department of Computer Science and Engineering,

Shanghai Jiao Tong University, Shanghai, China

{chenlusz,kai.yu}@sjtu.edu.cn

Abstract. Data sparsity problem is a key challenge of Natural Language Understanding (NLU), especially for a new target domain. By training an NLU model in source domains and applying the model to an arbitrary target domain directly (even without fine-tuning), few-shot NLU becomes crucial to mitigate the data scarcity issue. In this paper, we propose to improve prototypical networks with vector projection distance and abstract triangular Conditional Random Field (CRF) for the few-shot NLU. The vector projection distance exploits projections of contextual word embeddings on label vectors as word-label similarities, which is equivalent to a normalized linear model. The abstract triangular CRF learns domain-agnostic label transitions for joint intent classification and slot filling tasks. Extensive experiments demonstrate that our proposed methods can significantly surpass strong baselines. Specifically, our approach can achieve a new state-of-the-art on two few-shot NLU benchmarks (Few-Joint and SNIPS) in Chinese and English without fine-tuning on target domains.

Keywords: Few-shot learning · Natural language understanding

1 Introduction

Natural language understanding (NLU) is a critical component of conversational dialogue systems, converting user’s utterances into the corresponding semantic representations for a specific narrow domain (e.g., *booking hotel*, *searching flight*). Typically, the NLU module in goal-oriented dialogue systems contains two sub-tasks: intent classification and slot filling [13], as shown in Fig. 1. Intent classification is typically treated as a sentence classification problem, and slot filling is treated as a sequence labeling problem in which contiguous sequences of words are tagged with semantic labels (slots).

Recently, motivated by commercial applications like Amazon Alexa, Apple Siri, Google Assistant, and Microsoft Cortana, great interest has been attached to rapid domain transfer and adaptation with only a few samples. Few-shot learning approaches [5,20] become appealing in this scenario [7,8,22], where a general (domain-agnostic) model is learned from existing domains and transferred to new domains rapidly with merely few examples (e.g., in one-shot learning, only one example for each new class). The few examples sketch a new domain for the model.

The similarity-based few-shot learning methods have been widely analyzed on classification problems [16, 17, 20, 22], which classify an item according to its similarity with the representation of each class. These methods learn a domain-general encoder to extract feature vectors for items in existing domains and utilize the same encoder to obtain the representation of each new class from very few labeled samples (*support set*). This scenario has been successfully adopted in the slot filling task [8]. Nonetheless, it is still a challenge to design appropriate word-label similarity metrics for better generalization capability.

In this work, a vector projection distance is proposed to improve prototypical networks for few-shot NLU (joint intent classification and slot filling). To eliminate the impact of unrelated label vectors but with large norms, we exploit projections of contextual word embeddings on each normalized label vector as the word-label similarity. Meanwhile, the half norm of each label vector is utilized as a threshold, which can help reduce false-positive errors. To better model the intent classification and slot filling jointly, we also propose an abstract triangular CRF with abstract label transitions which can be shared across domains.

Our methods are evaluated on two few-shot NLU benchmarks (Few-Joint and SNIPS) in Chinese and English, respectively. Experimental results show that our methods can outperform various few-shot learning baselines and achieve state-of-the-art performances without fine-tuning on target domains. Our contributions are summarized as follows:

- We propose a vector projection distance to improve prototypical networks for few-shot NLU, which leads to better generalization capability of NLU models.
- We propose an abstract triangular CRF to model the intent classification and slot filling jointly, learning abstract label transitions across domains.
- We conduct extensive experiments with different distance functions and ablation studies to validate the effectiveness of our methods.

2 Related Work

The similarity-based few-shot learning aims to learn an effective distance metric [16, 20]. It can be simpler and more efficient than other meta-learning methods [6, 15].

For few-shot learning in the natural language processing community, researchers pay more attention to classification tasks, such as text classification [22]. Recently, few-shot learning for NLU task becomes popular and appealing. Fritzler et al. [7] explored few-shot NER with the prototypical network. Hou et al. [8] exploited the TapNet and label dependency transferring for both slot filling tasks. Yu et al. [23] explored retrieval-based methods for intent classification and slot filling tasks in few-shot settings. We are the first to utilize vector projections as word-label similarities in few-shot NLU. Triangular CRF has been applied in single domain NLU [21], while the transition weights of source domains can not be used in the target domain directly. We propose the abstract triangular CRF to share the underlying factors of transitions among different domains.

Several methods choose to train NLU models on source domains and keep fine-tuning on a target domain [1, 9, 12]. However, keep fine-tuning will produce different model parameters for new domains, which is not efficient and economical. Results show that our methods can beat these strong baselines even without fine-tuning.

sentence (\mathbf{x})	Show	me	flights	from	<i>Shanghai</i>	to	<i>New</i>	<i>York</i>
slots (\mathbf{y})	o	o	o	o	B-FromCity	o	B-ToCity	I-ToCity
intent (\mathbf{z})	Find_Flight							

Fig. 1. An example of intent and slot annotation (IOB format) in domain FlightTravel.

3 Problem Formulation

Intent classification and slot filling are major tasks of NLU in task-oriented dialogue systems. An intent is a purpose or a goal that underlies a user-generated sentence. Therefore, intent classification can be seen as a sentence classification problem. Slot filling aims to automatically extract a set of attributes or “slots” with the corresponding values. It is typically treated as a sequence labeling problem. An example of data annotation is provided in Fig. 1. The user’s intent is to find flights. For slot annotation, it follows the popular inside/outside/beginning (IOB) schema.

Let $\mathbf{x} = (x_1, \dots, x_{|\mathbf{x}|})$ denote an input sentence (i.e., word sequence), \mathbf{z} denote its intent label, and $\mathbf{y} = (y_1, \dots, y_{|\mathbf{x}|})$ denote its output sequence of slot tags, where $|\mathbf{x}|$ is the sentence length. For each domain \mathcal{D} , it includes a set of $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ pairs, i.e., $\mathcal{D} = \{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}, \mathbf{z}^{(n)})\}_{n=1}^{|\mathcal{D}|}$, where $|\mathcal{D}|$ is the total sample number.

In the few-shot scenario, the NLU model is trained on several source domains $\{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_M\}$, and then directly evaluated on a new target domain \mathcal{D}_t which only contains few labeled samples (*support set*). The support set, $\mathcal{S} = \{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}, \mathbf{z}^{(n)})\}_{n=1}^{|\mathcal{S}|}$, usually includes K examples (K-shot) for each of N labels (N-way). Thus, the few-shot NLU task is to find the best slot sequence \mathbf{y}^* and intent \mathbf{z}^* jointly, given an input query \mathbf{x} in target domain \mathcal{D}_t and its corresponding support set \mathcal{S} ,

$$\mathbf{y}^*, \mathbf{z}^* = \arg \max_{\mathbf{y}, \mathbf{z}} p_\theta(\mathbf{y}, \mathbf{z} | \mathbf{x}, \mathcal{S}) \quad (1)$$

where θ refers to parameters of the NLU model, the $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ pair and the support set are in the target domain, i.e., $(\mathbf{x}, \mathbf{y}, \mathbf{z}) \sim \mathcal{D}_t$ and $\mathcal{S} \sim \mathcal{D}_t$.

The few-shot NLU model is trained on the source domains to minimise the error in predicting slots and intents jointly conditioned on the support set,

$$\theta^* = \arg \min_{\theta} \sum_{m=1}^M \sum_{\substack{(\mathbf{x}, \mathbf{y}, \mathbf{z}) \sim \mathcal{D}_m \\ \mathcal{S} \sim \mathcal{D}_m}} -\log p_\theta(\mathbf{y}, \mathbf{z} | \mathbf{x}, \mathcal{S}) \quad (2)$$

4 Our Proposed Few-Shot NLU Model

In this section, we will introduce our prototypical networks for the few-shot NLU task, which is improved with vector projection distance and abstract triangular Conditional Random Field (CRF). The main architecture of our model is illustrated in Fig. 2. Our model consists of two parts: support set reader and semantic parser. The support set

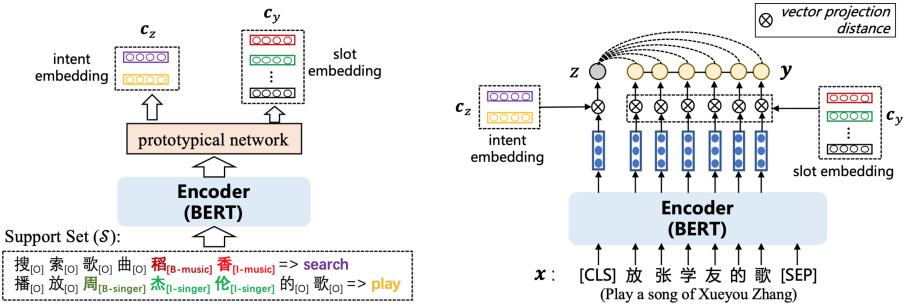


Fig. 2. The architecture of our proposed few-shot NLU model consists of two parts: Support Set Reader (the left part) and Semantic Parser (the right part).

reader exploits a BERT encoder to compute embeddings of all sentences in the support set, and it applies a prototypical network to get the central vector of each intent and slot label (i.e., intent and slot embeddings). The semantic parser also utilizes a BERT encoder to extract word embeddings of the input sentence. It then calculates intent and slot logits by measuring vector projection distance between word and label embeddings. Finally, an abstract triangular CRF is applied to predict intent and slot labels jointly. BERT encoders in the support set reader and semantic parser are shared.

4.1 Support Set Reader

Obviously, an NLU model cannot make predictions for unknown labels. Thus, it is essential to extract label features from the support set, which contains a minimal annotation set for all intents and slots of the new domain.

For the support set, $\mathcal{S} = \{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}, z^{(n)})\}_{n=1}^{|\mathcal{S}|}$, a contextual word embedding function E is applied onto each support sentence \mathbf{x} to get dense features, i.e., $E(\mathbf{x})$. Generally, E can be a kind of sequence model, like BLSTM [14], Transformer [19]. In this paper, we adopt a pre-trained BERT model [4] as E , i.e.,

$$(\mathbf{e}_0, \mathbf{e}_1, \dots, \mathbf{e}_{|\mathbf{x}|}) = E(\mathbf{x}) = \text{BERT}([\text{CLS}], \mathbf{x}) \quad (3)$$

where $[\text{CLS}]$ is a special token to get whole sentence embedding (i.e., \mathbf{e}_0), and \mathbf{e}_i refers to the BERT embedding of each input word, $i = 1, \dots, |\mathbf{x}|$.

Following prototypical networks [16] in the image classification field, the prototype of each intent or slot (label embedding) is defined as the mean vector of the embedded supporting points belonging to it.

$$\mathbf{c}_z = \frac{1}{N_z} \sum_{n=1}^{|\mathcal{S}|} \mathbb{I}\{z^{(n)} = z\} E(\mathbf{x}^{(n)})_0 \quad (4)$$

$$\mathbf{c}_y = \frac{1}{N_y} \sum_{n=1}^{|\mathcal{S}|} \sum_{i=1}^{|\mathbf{x}^{(n)}|} \mathbb{I}\{y_i^{(n)} = y\} E(\mathbf{x}^{(n)})_i \quad (5)$$

where $\mathbb{I}\{\cdot = \cdot\}$ is an indicator function, $N_z = \sum_{n=1}^{|S|} \mathbb{I}\{z^{(n)} = z\}$ is the number of sentences labeled with intent z in the support set, and $N_y = \sum_{n=1}^{|S|} \sum_{i=1}^{|x^{(n)}|} \mathbb{I}\{y_i^{(n)} = y\}$ is the number of words labeled with slot y in the support set.

4.2 Semantic Parser

The semantic parser also exploits a BERT encoder to calculate contextual word embeddings of the input sentence, then predicts the intent and slot labels jointly with label embeddings and the abstract triangular CRF.

Linear Conditional Random Field (CRF) [14, 18] considers the correlations between slots in neighborhoods, while the triangular CRF also considers the correlations between intent and slot. Thus, the triangular CRF can jointly decode the most likely slot sequence and intent class given the input sentence. The posterior probability of joint intent z and slot sequence \mathbf{y} is computed via:

$$\psi_\theta(\mathbf{y}, z, \mathbf{x}, \mathcal{S}) = f_E(z, \mathbf{x}, \mathcal{S}) + \sum_{i=1}^{|x|} (f_{T_{IS}}(z, y_i) + f_{T_{SS}}(y_{i-1}, y_i) + f_E(y_i, \mathbf{x}, \mathcal{S})) \quad (6)$$

$$p_\theta(\mathbf{y}, z | \mathbf{x}, \mathcal{S}) = \frac{\exp(\psi_\theta(\mathbf{y}, z, \mathbf{x}, \mathcal{S}))}{\sum_{y', z'} \exp(\psi_\theta(\mathbf{y}', z', \mathbf{x}, \mathcal{S}))} \quad (7)$$

where $f_E(z, \mathbf{x}, \mathcal{S})$ is the emission score of the intent, and $f_E(y_i, \mathbf{x}, \mathcal{S})$ is the emission score of the slot at the i -th step. $f_{T_{IS}}(z, y_i)$ is the transition score between intent z and slot y_i , and $f_{T_{SS}}(y_{i-1}, y_i)$ is the transition score between two adjacent slots.

Emission Score and Vector Projection Distance. The emission scorer independently assigns each word a score with respect to each label y_i , which is defined as a word-label similarity function:

$$f_E(z, \mathbf{x}, \mathcal{S}) = \text{SIM}(E(\mathbf{x})_0, \mathbf{c}_z) \quad (8)$$

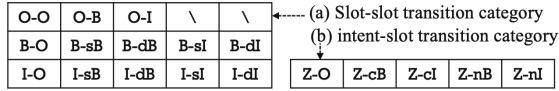
$$f_E(y_i, \mathbf{x}, \mathcal{S}) = \text{SIM}(E(\mathbf{x})_i, \mathbf{c}_{y_i}) \quad (9)$$

For the word-label similarity function, we propose to exploit vector projections of word embeddings \mathbf{x}_i on each normalized label vector \mathbf{c}_k :

$$\text{SIM}(\mathbf{x}_i, \mathbf{c}_k) = \mathbf{x}_i^\top \frac{\mathbf{c}_k}{\|\mathbf{c}_k\|} - \frac{1}{2} \|\mathbf{c}_k\| \quad (10)$$

Different from the dot product used in [8], it can help eliminate the impact of \mathbf{c}_k 's norm to avoid the circumstance where the norm of \mathbf{c}_k is large enough to dominate the similarity metric. In order to reduce false-positive errors, the half norm of each label vector is utilized as an adaptive bias term. It is called VPB, and the version without the bias is named VP.

A simple interpretation for the above vector projection distance is to learn a distinct linear classifier for each label. We can rewrite the above formulas as a linear model:

**Fig. 3.** Abstract categories of label transitions consists of slot-slot and intent-slot transitions.**Table 1.** Definitions of abstract transition categories. X and Y refer to arbitrary two different slot names.

Category	Description	Category	Description
O-O	$y_{i-1} = O \& y_i = O$	I-sB	$y_{i-1} = I-X \& y_i = B-X$
O-B	$y_{i-1} = O \& y_i = B-X$	I-dB	$y_{i-1} = I-X \& y_i = B-Y$
O-I	$y_{i-1} = O \& y_i = I-X$	I-sI	$y_{i-1} = I-X \& y_i = I-X$
B-O	$y_{i-1} = B-X \& y_i = O$	I-dI	$y_{i-1} = I-X \& y_i = I-Y$
B-sB	$y_{i-1} = B-X \& y_i = B-X$	Z-O	z is intent, $y_i = O$
B-dB	$y_{i-1} = B-X \& y_i = B-Y$	Z-cB	$y_i = B-X$, z and y_i co-occurred
B-sI	$y_{i-1} = B-X \& y_i = I-X$	Z-cI	$y_i = I-X$, z and y_i co-occurred
B-dI	$y_{i-1} = B-X \& y_i = I-Y$	Z-nB	$y_i = B-X$, z and y_i do not co-occurred
I-O	$y_{i-1} = I-X \& y_i = O$	Z-nI	$y_i = I-X$, z and y_i do not co-occurred

$$\text{SIM}(\mathbf{x}_i, \mathbf{c}_k) = \mathbf{x}_i^\top \mathbf{w}_k + b_k \quad (11)$$

where $\mathbf{w}_k = \frac{\mathbf{c}_k}{\|\mathbf{c}_k\|}$ and $b_k = -\frac{1}{2}\|\mathbf{c}_k\|$. The weights are normalized as $\|\mathbf{w}_k\| = 1$ to improve the generalization capability of the few-shot model. Experimental results indicate that vector projection is an effective choice compared to dot product, cosine similarity, squared Euclidean distance, etc.

Transition Score and Abstract Triangular CRF. The transition score between two slots captures temporal dependencies of slots in consecutive time steps, and the transition score between an intent and a slot captures task dependencies of intent classification and slot filling. The Transition Score is learnable scalar for each label pair. We classify all label pairs (slot-slot and intent-slot) into abstract categories that are domain-agnostic to share the underlying factors of transitions among different domains. Transition scores of label pairs belong to the same abstract category are shared.

Following [8], we design 13 abstract categories for slot-slot transitions as shown in Fig. 3 (a), where B (I) refers to any slot starting with ‘B’ (‘I’), sB (sI) means a slot containing the same name with the previous slot, and dB (dI) means a slot which contains a different name with the previous slot.

For intent-slot transitions, we define 5 abstract categories by exploiting intent-slot co-occurrence relations in the support set \mathcal{S} . As shown in Fig. 3 (b), Z means any intent, cB (cI) are slots which start with ‘B’ (‘I’) and co-occur with the intent in a same support sample, and nB (nI) refers to slots not co-occurring with the intent.

The abstract slot-slot ($f_{T_{ss}}(y_{i-1}, y_i)$) and intent-slot ($f_{T_{is}}(z, y_i)$) transitions are defined in Table 1.

5 Experiment

We evaluate the proposed method on the natural language understanding task of 1/3/5-shot setting, which transfers knowledge from source domains (training) to an unseen target domain (testing) containing only 1/3/5-shot support set.

5.1 Settings

Dataset. We conduct experiments on two public datasets: SNIPS [3] (in English) and Few-Joint [10] (in Chinese). For SNIPS, we use the data split¹ of 5-shot setting without intent classification task. For Few-Joint, we utilize the 1-shot, 3-shot and 5-shot settings, which contains both intent classification and slot filling tasks. They are in the *episode* data setting [20], where each episode contains a support set (1/3/5-shot) and a batch of labeled samples.

The SNIPS dataset consists of 7 domains with different slots (totally 53 slots): Weather (We), Music (Mu), PlayList (Pl), Book (Bo), Search Screen (Se), Restaurant (Re), and Creative Work (Cr). We select one target domain for evaluation, one domain for validation, and utilize the rest domains as source domains for training.

FewJoint is a joint NLU dataset used in the few-shot learning contest of SMP2020- ECDT Task-1². It contains 59 multi-intent domains, 143 different intents, and 205 different slots. We follow the original data split, that there are 45 domains for training, 5 domains for validation and 9 domains for evaluation.

Evaluation. Three metrics are used for evaluation: Intent Accuracy, Slot F₁-score³, Joint Accuracy. The Joint Accuracy evaluates the sentence level accuracy, which considers one sentence is correct only when all its slots and intent are correct.

We take the average score of all evaluation domains as the final result. To mitigate the bias of different random seeds and conduct robust evaluation, we run each experiment for 5 times with different random seeds and report the average score of 5 random seeds for all results.

Training Details. As SNIPS is in English, we use the uncased Bert-base [4] as the BERT encoder to extract contextual word embeddings. For Few-Joint in Chinese, we use Chinese-bert-base⁴ and Chinese-roberta-wwm-ext⁵.

The models are trained using ADAM [11] and updated after each episode. We fine-tune BERT with layer-wise learning rate decay (rate is 0.9), i.e., the parameters of the l -th layer get an adaptive learning rate $1e-5 * 0.9^{(L-l)}$, where L is the total number of layers in BERT. For the abstract triangular CRF transition parameters, they are initialized as zeros, and large learning rates of 5e-3 and 1e-3 are applied for Few-Joint and SNIPS datasets, respectively. The models are trained for 10 iterations, and we save the parameters with the best average score on the validation domains.

¹ <https://atmahou.github.io/attachments/ACL2020data.zip>.

² <https://smp2020.aconf.cn/smp.html>.

³ CoNLL evaluation script: <https://www.clips.uantwerpen.be/conll2000/chunking/output.html>.

⁴ <https://github.com/google-research/bert>.

⁵ <https://github.com/ymcui/Chinese-BERT-wwm>.

Table 2. Scores of 1/3/5-shot NLU tasks on Few-Joint dataset. * means Chinese-bert-base is used as the BERT encoder, while † means Chinese-roberta-wwm-ext is used.

Model	1-shot			3-shot			5-shot		
	int. acc	slot F ₁	joint acc.	int. acc	slot F ₁	joint acc.	int. acc	slot F ₁	joint acc.
JointTransfer	41.83	26.89	12.27	—	—	—	57.50	29.00	18.81
Meta-JOSFIN	57.92	29.26	15.00	—	—	—	78.91	53.88	36.63
SepProto	66.35	27.24	10.92	72.30	34.11	16.40	75.64	36.08	15.93
JointProto	58.52	29.49	9.64	78.46	40.37	23.65	70.93	39.47	14.48
ConProm+FT	61.24	42.02	24.63	—	—	—	78.33	62.34	40.25
ConProm+FT+TR	63.67	42.44	27.72	—	—	—	78.43	69.44	46.54
Our method (VP)*	69.09	60.69	40.75	81.32	73.15	58.47	86.70	77.42	66.41
Our method (VPB)*	68.03	60.95	40.29	80.58	75.05	60.30	85.77	78.43	67.04
Our method (VP) [†]	70.66	63.55	42.23	84.19	75.78	60.12	88.29	78.33	65.08
Our method (VPB) [†]	69.21	63.09	41.26	82.79	76.85	60.11	87.15	80.54	67.36

Table 3. Slot F₁ scores of 5-shot slot filling task on SNIPS dataset. Scores of 7 target domains and the average are reported.

Model	We	Mu	Pl	Bo	Se	Re	Cr	Avg.
L-ProtoNet+CDT+PWE [8]	74.68	56.73	52.20	78.79	80.61	69.59	67.46	68.58
L-TapNet+CDT+PWE [8]	71.64	67.16	75.88	84.38	82.58	70.05	73.41	75.01
Retriever [23]	82.95	61.74	71.75	81.65	73.10	79.54	51.35	71.72
Our method (VP)	79.88	67.77	78.08	87.68	86.59	79.95	75.61	79.37
Our method (VPB)	82.91	69.23	80.85	90.69	86.38	81.20	76.75	81.14

5.2 Baselines

JointTransfer is a domain transferred NLU model based on the JointBERT [2], which consists of a shared BERT encoder with intent classification and slot filling layers. It is first pre-trained on source domains and then fine-tuned on the support set of the target domain.

Meta-JOSFIN [1] is a meta-learning model based on the MAML [6]. The meta-learner model is also a joint NLU model similar to JointTransfer. It learns initial parameters on source domains, which can fast adapt to the target domain after only a few updates.

SeqProto is a prototypical-based NLU model with BERT embedding that learns intent classification and slot filling separately. During the experiment, it is pre-trained on source domains and then directly applies to target domains without fine-tuning.

JointProto [12] is all the same as SepProto except that BERT encoders for intent classification and slot filling sub-tasks are shared.

ConProm+FT [9] is a contrastive prototype merging network, which learns to bridge metric spaces of intent and slot on data-rich domains, and then adapt the bridged metric space to a specific few-shot domain. “+FT” means fine-tuning on the support set similar to Meta-JOSFIN.

ConProm+FT+TR [9] adds Transition Rules (+TR) between slot tags, which bans illegal slot prediction, such as ‘I’ tag after ‘O’ tag.

Table 4. Comparing different distance functions on 1/3/5-shot settings of Few-Joint dataset.

SIM(\mathbf{x}, \mathbf{c})	1-shot			3-shot			5-shot		
	int. acc	slot F ₁	joint acc.	int. acc	slot F ₁	joint acc.	int. acc	slot F ₁	joint acc.
VP	70.66	63.55	42.23	84.19	75.78	60.12	88.29	78.33	65.08
VPB	69.21	63.09	41.26	82.79	76.85	60.11	87.15	80.54	67.36
Dot	69.93	49.54	33.98	80.84	59.30	47.32	84.40	61.04	50.03
Euclidean	68.91	48.59	31.13	82.69	66.65	52.43	87.30	70.22	58.80
Cosine	56.44	21.51	16.53	61.63	29.76	23.73	74.02	31.11	29.82

Table 5. Ablation study of the abstract triangular CRF on 1/3/5-shot settings of Few-Joint dataset.

Model	1-shot			3-shot			5-shot		
	int. acc	slot F ₁	joint acc.	int. acc	slot F ₁	joint acc.	int. acc	slot F ₁	joint acc.
Our method (VP)	70.66	63.55	42.23	84.19	75.78	60.12	88.29	78.33	65.08
(-) w/o intent-slot	68.29	60.88	37.76	81.30	72.82	54.62	84.23	75.52	58.16
(-) w/o CRF	69.82	44.89	25.73	81.37	55.05	40.88	83.81	58.95	44.21
Our method (VPB)	69.21	63.09	41.26	82.79	76.85	60.11	87.15	80.54	67.36
(-) w/o intent-slot	68.07	61.64	38.22	81.63	74.47	56.59	84.66	78.98	62.61
(-) w/o CRF	68.60	47.93	28.82	81.90	61.07	47.96	85.42	69.22	55.60

5.3 Main Results

Table 2 shows main results on 1-shot, 3-shot and 5-shot settings of Few-Joint dataset, where results of baselines on 1-/5-shot are borrowed from [9], and results of 3-shot are borrowed from [10]. Our methods can outperform the previous results on intent accuracy, slot F₁ score and joint accuracy with large margins. Especially, our methods perform even better than the baselines which are fine-tuned on support sets of target domains, e.g., Meta-JOSFIN, ConProm+FT and ConProm+FT+TR. By comparing VP and VPB, we find that the adaptive bias item in Eq. (10) would be effective in 3- or 5-shot settings. For the pre-trained BERT encoder, the results show that Chinese-roberta-wwm-ext is better than Chinese-bert-base. Therefore, we will use Chinese-roberta-wwm-ext for the BERT encoder in the rest experiments on Few-Joint dataset.

Table 3 shows results on 5-shot slot filling of SNIPS dataset. Our method can significantly outperform the previous state-of-the-art models. If we incorporate the negative half norm of each label vector as a bias (VPB), the average slot F₁ score over 7 domains is dramatically improved. We speculate that 5-shot slot filling involves multiple support points for each slot, thus false-positive errors could occur more frequently if there is no threshold for predicting each label.

5.4 Analysis

Distance Functions. For the word-label similarity function SIM(\mathbf{x}, \mathbf{c}), we propose to used vector projection distance, as shown in Eq. (10). Here, we conduct contrastive experiments between our proposed vector projection distances (VP and VPB) and other

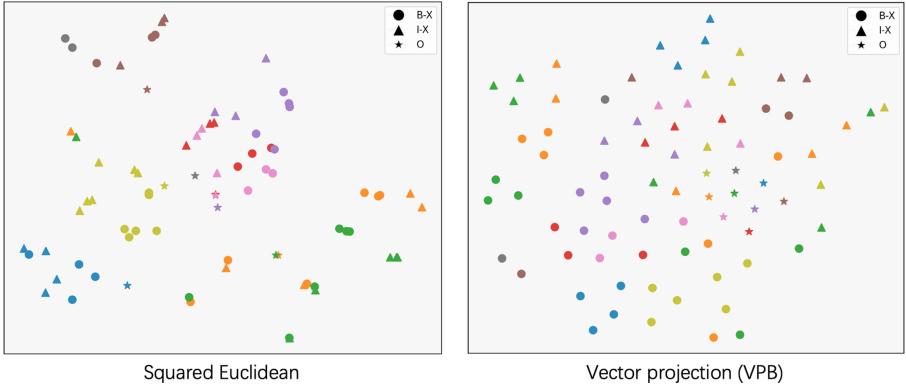


Fig. 4. Visualization of slot embedding distributions of squared Euclidean distance and VPB based NLU models in the 5-shot setting on Few-Joint test set (9 domains with different colours), by using TSNE (step = 1500).

variants including the dot product ($\mathbf{x}^\top \mathbf{c}$), squared Euclidean distance ($-\frac{1}{2} \|\mathbf{x} - \mathbf{c}\|^2$), and cosine function ($\frac{\mathbf{x}^\top \mathbf{c}}{\|\mathbf{x}\| \|\mathbf{c}\|}$). The results in Table 4 show that our methods can significantly outperform these alternative metrics. The cosine function can lead to really poor performances, which may be caused by its fixed value range (i.e., $\frac{\mathbf{x}^\top \mathbf{c}}{\|\mathbf{x}\| \|\mathbf{c}\|} \in [-1, 1]$).

To further understand how the vector projection distance affects extracting label embeddings from support sets, we visualize the slot embedding distributions in the metric space. As shown in Fig. 4, it is exciting to see that our method (VPB) can make slot embeddings more discriminative, while the squared Euclidean distance would lead to ambiguous points. We can also find that VPB can gather \circ of all test domains close together. VPB always keeps slots B-X in the lower-left part while makes slots I-X in the upper-right part.

Effectiveness of the Abstract Triangular CRF. We also conduct ablation studies to validate the effectiveness of the abstract triangular CRF, as shown in Table 5. From the results, we can find that performances drop with significant margins if we remove intent-slot transitions (“w/o intent-slot”). When the intent-slot transitions are removed, intents are predicted via a softmax function. Meanwhile, if we remove both intent-slot and slot-slot transitions from the abstract triangular CRF (i.e. “w/o CRF”), performances decrease further. The results on both VP and VPB show that the proposed abstract triangular CRF can improve the joint accuracy significantly and be effective for joint intent classification and slot filling.

From Table 5, we can also find that intent-slot transitions are more effective in the 5-shot setting. The reason may be that 1-shot or 3-shot setting is not sufficient for obtaining complete intent-slot co-occurrences of a target domain. Since the absence of slot-slot transitions can lead to a larger decrease, it seems that slot-slot transitions are more important than intent-slot transitions for joint accuracy.

O-O, 1.02	O-B, 1.22	O-I, -1.79	\	\	Z-O, 0.75	Z-cB, 0.32	Z-cl, 0.19
B-O, -1.27	B-sB, -1.67	B-dB, -0.70	B-sI, 1.75	B-dI, -1.73	\	Z-nB, -1.04	Z-nI, -0.75
I-O, -0.53	I-sB, -2.20	I-dB, -0.27	I-sI, 1.47	I-dI, -1.64			

Fig. 5. Abstract transition weights of our method (VPB) in the 5-shot setting on Few-Joint dataset.**Table 6.** Compare different learning rates of the abstract triangular CRF on 1/3/5-shot settings of Few-Joint dataset.

Model	lr	1-shot			3-shot			5-shot		
		int. acc	slot F ₁	joint acc.	int. acc	slot F ₁	joint acc.	int. acc	slot F ₁	joint acc.
Our method (VP)	5e-3	70.66	63.55	42.23	84.19	75.78	60.12	88.29	78.33	65.08
	3e-3	70.50	63.43	42.02	84.37	75.09	59.79	87.71	77.78	63.78
	1e-3	70.18	57.62	36.40	83.79	68.58	52.80	85.97	72.26	57.03
	5e-4	69.78	54.87	32.91	82.56	66.29	50.17	84.93	68.28	52.31
Our method (VPB)	5e-3	69.21	63.09	41.26	82.79	76.85	60.11	87.15	80.54	67.36
	3e-3	68.94	62.79	40.51	82.66	76.44	59.34	86.77	79.61	65.81
	1e-3	68.52	59.73	37.60	83.12	73.78	57.52	86.23	78.22	63.28
	5e-4	67.92	56.08	34.77	82.64	71.27	55.78	85.61	76.04	60.28

We draw abstract CRF transition weights of our method (VPB) in Fig. 5. It learns several transition rules. For example, a slot beginning with ‘I’ after a different slot beginning with ‘B’ (i.e., B-dI) is penalized with a negative transition weight. The transition from an intent to a slot co-occurring with it (e.g., Z-cB and Z-cI) would be encouraged, while the transition from an intent to any slot never co-occurring with it in the support set (e.g., Z-nB and Z-nI) would be penalized. This shows how the abstract triangular CRF works in our methods.

Should Learning Rate of CRF Transitions Be Larger? The parameters of CRF Transitions are initialized from scratch, which is different from the BERT encoder. Therefore, the learning rate for CRF Transitions could be larger. Results of different learning rates (<{5e-3, 3e-3, 1e-3, 5e-4}) are shown in Table 6. The results demonstrate that large learning rates can improve performance effectively, like 3e-3 and 5e-3. We also find that the VPB function can outperform VP dramatically for a small learning rate (e.g., 1e-3 and 5e-4).

6 Conclusion

In this paper, we propose a vector projection distance and abstract triangular CRF for few-shot intent classification and slot filling tasks. The vector projection distance can be interpreted as a normalized linear model, which can improve the model generalization capability. The abstract triangular CRF learns domain-agnostic intent-slot and slot-slot transitions to model NLU tasks better jointly. Experimental results demonstrate that our method can significantly outperform strong baselines on Few-Joint and SNIPS datasets in few-shot settings.

Acknowledgements. We thank all the anonymous reviewers for their thoughtful comments. This work was supported by the GuSu Innovation Fund (ZXT20200003).

References

1. Bhathiya, H.S., Thayasilvam, U.: Meta learning for few-shot joint intent detection and slot-filling. In: ICMLT, pp. 86–92 (2020)
2. Chen, Q., Zhuo, Z., Wang, W.: BERT for joint intent classification and slot filling. arXiv preprint [arXiv:1902.10909](https://arxiv.org/abs/1902.10909) (2019)
3. Coucke, A., et al.: Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. arXiv preprint [arXiv:1805.10190](https://arxiv.org/abs/1805.10190) (2018)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL, pp. 4171–4186 (2019)
5. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE Trans. Pattern Anal. Mach. Intell. **28**(4), 594–611 (2006)
6. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML, pp. 1126–1135 (2017). [JMLR.org](https://jmlr.org)
7. Fritzler, A., Logacheva, V., Kretov, M.: Few-shot classification in named entity recognition task. In: SAC, pp. 993–1000 (2019)
8. Hou, Y., et al.: Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. In: ACL (2020)
9. Hou, Y., Lai, Y., Chen, C., Che, W., Liu, T.: Learning to bridge metric spaces: few-shot joint learning of intent detection and slot filling. In: ACL (Findings) (2021)
10. Hou, Y., et al.: Fewjoint: a few-shot learning benchmark for joint language understanding. arXiv preprint [arXiv:2009.08138](https://arxiv.org/abs/2009.08138) (2020)
11. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
12. Krone, J., Zhang, Y., Diab, M.: Learning to classify intents and slot labels given a handful of examples. In: NLP4ConvAI, pp. 96–108 (2020)
13. Liu, B., Lane, I.: Attention-based recurrent neural network models for joint intent detection and slot filling. In: INTERSPEECH, pp. 685–689 (2016)
14. Ma, X., Hovy, E.: End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In: ACL, pp. 1064–1074 (2016)
15. Munkhdalai, T., Yu, H.: Meta networks. In: ICML, pp. 2554–2563 (2017). [JMLR.org](https://jmlr.org)
16. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: NeurIPS, pp. 4077–4087 (2017)
17. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to compare: relation network for few-shot learning. In: CVPR, pp. 1199–1208 (2018)
18. Sutton, C., McCallum, A., et al.: An introduction to conditional random fields. Found. Trends® Mach. Learn. **4**(4), 267–373 (2012)
19. Vaswani, A., et al.: Attention is all you need. In: NeurIPS, pp. 5998–6008 (2017)
20. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. In: NeurIPS, pp. 3630–3638 (2016)
21. Xu, P., Sarikaya, R.: Convolutional neural network based triangular CRF for joint intent detection and slot filling. In: ASRU, pp. 78–83 (2013)
22. Yan, L., Zheng, Y., Cao, J.: Few-shot learning for short text classification. Multimed. Tools Appl. **77**(22), 29799–29810 (2018). <https://doi.org/10.1007/s11042-018-5772-4>
23. Yu, D., He, L., Zhang, Y., Du, X., Pasupat, P., Li, Q.: Few-shot intent classification and slot filling with retrieved examples. In: NAACL, pp. 734–749 (2021)



Cross-domain Slot Filling with Distinct Slot Entity and Type Prediction

Shudong Liu, Peijie Huang^(✉), Zhanbiao Zhu, Hualin Zhang, and Jianying Tan

College of Mathematics and Informatics, South China Agricultural University,
Guangzhou, China

{sudan,zhuzhanbiao,vol_chris,jytan}@stu.scau.edu.cn,
pjhuang@scau.edu.cn

Abstract. Supervised learning approaches have been proven effective in slot filling, but they need massive labeled training data which is expensive and time-consuming in a given domain. Recent models for cross-domain slot filling adopt transfer learning framework to cope with the data scarcity problem. However, these cross-domain slot filling models rely on the same encoder representation in different stages for slot entity task and slot type task, which decrease the performance of both tasks. Besides, these models treat different source domains equally and ignore the shared slot-related information in different domains, which may damage the performance of cross-domain learning. In this paper, we present a pipeline approach for cross-domain slot filling (**PCD**) by learning distinct contextual representations for slot entity identification and slot type alignment, and fusing slot entity information at the input layer of the slot type alignment model for incorporating global context. Moreover, we also present a simple yet effective instance weighting scheme (**Iw**) to our approach for better capturing the slot entities in the cross-domain setting. Experiments on multiple domains show that our approach achieves state-of-the-art performance in cross-domain slot filling. Ablation analysis and further experiments also prove the effectiveness of each part of our model, especially in the identification of slot entities.

Keywords: Spoken language understanding · Slot filling · Cross-domain learning · Instance weighting scheme · Zero-shot learning

1 Introduction

Spoken language understanding (SLU) is the core component of intelligent personal digital assistants (IPDAs) such as Microsoft Cortana, Google Assistant, Amazon Alexa, and Apple Siri [1]. It typically consists of intent detection and slot filling. Slot filling models capture useful semantic information which has been shown helpful for related NLP tasks.

Recently, supervised joint learning approaches have shown their effectiveness in slot filling [2–5]. Such joint models for intent detection and slot tagging have taken the state of the art of slot filling to a new level. However, such approaches are expensive and time-consuming due to the difficulties in

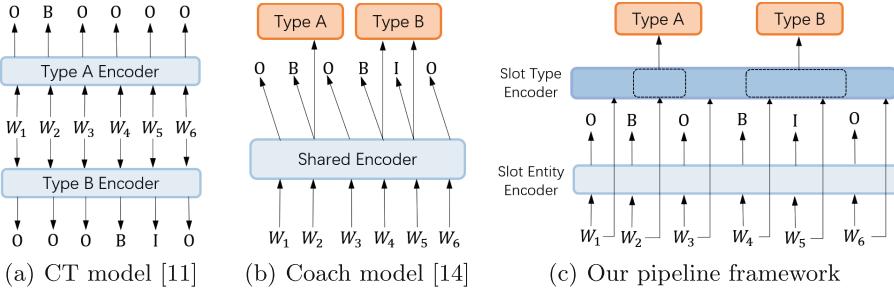


Fig. 1. Cross-domain slot filling frameworks.

collecting high-quality labeled training data with different domains. This limitation has motivated us to explore cross-domain slot filling for fast adaptation to new domains. Cross-domain adaptation copes with the data scarcity problem in low-resource target domains [6–10]. The key challenge of slot filling in a new domain is identifying unseen slot types without any supervision signals. Common approaches for cross-domain slot filling are focusing on employing slot description (e.g., the description of slot label *restaurant_type* is “restaurant type”) to predict unseen slots [11–15].

Existing cross-domain slot filling models can be classified into two main categories. As shown in Fig. 1 (a), the first part of work, such as the CT model [11], conducts slot filling individually for each slot type [12]. They generate word-level representations, then interact with the representation of each slot type description in semantic space. The final predictions are independent for each slot type based on the fused features. Besides, slot examples were also used to increase the robustness of domain adaptation [13]. However, such models exist a multiple prediction problem. Unlike the above models, as shown in Fig. 1 (b), Liu et al. [14] proposed a two-stage slot filling framework to avoid the multiple prediction problem and learn the general pattern of slot entities. They use the shared representation to identify whether the tokens are slot entities or not by a BIO (Begin/In/Out) 3-way classifier, and then predict their specific slot types based on slot descriptions. For example, given a movie-related utterance “*find andreas hofer at elevenses*”, the model will first capture the slot entity “*andreas hofer*” and then classify its label as “*movie_name*”. He et al. [15] further leveraged contrastive learning and adversarial attack to improve model robustness.

Though achieving the promising performance, these two-stage models still suffer from two issues: (1) They use the same encoder for identifying slot entities (by using BIO structure) in stage one and predicting the specific slot types for each entity in stage two. However, the information captured in cross-domain learning in slot entity identification and slot type alignment is different. The slot entity identification is to detect the entity boundary while the slot type alignment is to predict slot labels by contexts. The performance of these two-stage models drops in both tasks since affecting each other. (2) Such approaches treat each source domain corpus equally. However, in cross-domain learning,

different source domains have different contributions to the target domain, and some of them may even cause negative transfer problems [16]. For example, given target domain “GetWeather”, the model can get more improvements from “BookRestaurant” domain because of the location-related shared slots, but fewer improvements from the “PlayMusic” domain with no related shared slots at all.

To meet the above challenges, in this work, we propose a pipeline approach for cross-domain slot filling by learning distinct contextual representations for slot entities and slot types. The overall principle can be seen in Fig. 1 (c). It learns two independent encoders for the slot entity model and slot type model. To capture the entity information from the entity model in slot types prediction, we add boundary markers into the second encoder. In addition, we introduce an instance weighting scheme to control the contribution of different source domains to the target domain. The core idea is to compute the similarities between domains, which are used to adjust learning rates for the utterances of different domains. By doing so, the model tends to learn more shared-information in more similar domains, rather than in less similar domains.

Our main contributions are summarized as follows: (1) We propose a pipeline approach for cross-domain slot filling with distinct contextual representations for slot entities and slot types. (2) We introduce a simple yet effective instance weighting scheme for better capturing slot entities and alleviating the negative transfer problem. (3) Experiments in the zero-shot/few-shot settings on SNIPS and SMP-ECDT datasets show that our approach outperforms the state-of-the-art models. Ablation study and quantitative analysis also prove the effectiveness of the proposed model.

2 Our Approach

Figure 2 illustrates our pipeline model architecture by a sample user utterance “*find andreas hofer at elevenses*” and its corresponding slots. The pipeline model consists of a slot entity and a slot type model. The slot entity model predicts whether tokens are slot entities or not (BIO labels) and learns the slot entity pattern with the instance weighting scheme. The slot type model classifies the slot entities into related types with slot descriptions [11] and boundary markers.

2.1 Slot Entity Model

Following prior work, we utilize the BiLSTM-CRF structure [17] to encode the hidden states of tokens and predict the BIO labels. The input of the model is an utterance consisting of n tokens denoted as $W = [w_1, \dots, w_n]$. Let E be the embedding layer for utterances. We formulate the whole process as follows:

$$[\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n] = BiLSTM(E(W)) \quad (1)$$

$$[\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n] = CRF([\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n]) \quad (2)$$

where $[\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n]$ is the hidden layer and $[\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n]$ is the logits for the 3-way classification.

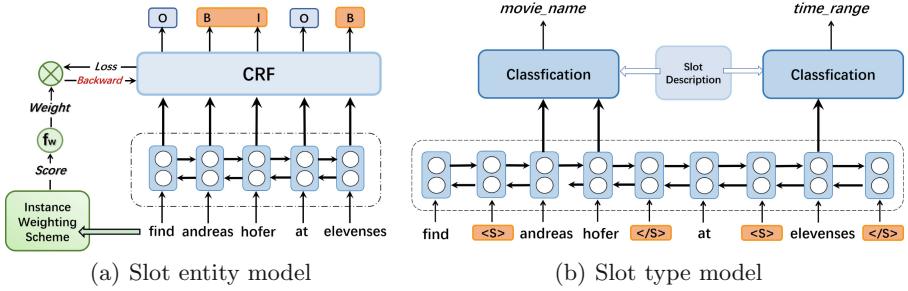


Fig. 2. The core architecture of our proposed pipeline model (PCD-Iw). Fig (a) displays the slot entity model with instance weighting scheme to identify whether the tokens are slot entities or not. Fig (b) shows the slot type model with boundary markers to match the specific slot types based on slot type descriptions. The boundary markers can be generated from the prediction results of the slot entity model.

2.2 Slot Type Model

The slot type model aims to classify the type of the slot entities predicted by the slot entity model. Prior work [14, 15] used the same encoder since it has captured the information about which parts are the entities to focus on in predicting slot types. However, due to the different granularity of the information to be captured by the two tasks in cross-domain setting, using the shared representation directly will damage the performance of the model. Hence, we build a new model for classifying slot types.

To capture the entity boundaries and highlight the slot entities, inspired by Zhong and Chen [18], we insert boundary markers at the input layer in this model. Specifically, given an input utterance W and a corresponding predicted slot entity, we define text markers as $\langle S \rangle$ and $\langle /S \rangle$, and insert them into the input utterance before and after the slot entities (Fig. 2 (b)). Let \widehat{W} denote this modified sequence with text markers inserted:

$$\widehat{W} = \dots \langle S \rangle, w_{START(i)}, \dots, w_{END(i)}, \langle /S \rangle \dots \quad (3)$$

By doing so, the position information and boundary of the slot entity can be explicitly used for the slot type prediction, which realizes the effect of the original shared encoder. We then apply another BiLSTM encoder on \widehat{W} to generate the context-aware representations:

$$[\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_{n+2m}] = BiLSTM(E(\widehat{W})) \quad (4)$$

where m denotes the number of the predicted slot entities in this utterance. We take its hidden states between the start and end markers($\langle S \rangle$ and $\langle /S \rangle$) to denote the slot representation. The representation \mathbf{r}_i of i^{th} slot entity can be denoted as:

$$\mathbf{r}_i = BiLSTM([\mathbf{h}_{START(i)}, \dots, \mathbf{h}_{END(i)}]) \quad (5)$$

Following Shah et al. [13] and Liu et al. [14], we sum the embedding of the slot description tokens as the description representation. Then we can obtain a slot description matrix $\mathbf{M}_{desc} \in \mathbb{R}^{n_s \times d_s}$ where n_s is the number of all the possible slot types and d_s is the dimension of slot description representation. Finally, we calculate the dot product as classification logits $s_i = \mathbf{r}_i \cdot \mathbf{M}_{desc}$ and get the cross-entropy loss.

2.3 Instance Weighting Scheme

Negative transfer often occurs in cross-domain learning because of the wide differences in the distribution of different domains. Especially in slot entities predicting, for the similar sentence structure, the slot entities that need to be captured in different domains are usually different. For example, slot *object.name* usually appears after the phrase “What is” in domain “SearchCreativeWork”. However, it will have a bad effect in the domain without slot *object.name*, leading to the prediction of redundant entities in the slot entity model.

Since we can get all the possible slots in the target domain (Table 1 and 2), we introduce a simple yet effective instance weighting scheme by using the ratio of shared slots. We quantify the similarity between the data of different source domains and the specified target domain. For a target domain (td), the scoring function for calculating the similarity of different source domains (sd) is as follows:

$$score(sd, td) = \frac{|Slot_{shared}|}{|Slot_{sd}|} \cdot \frac{|Slot_{shared}|}{|Slot_{td}|} \quad (6)$$

where $|Slot_{sd}|$ and $|Slot_{td}|$ are the numbers of slot types for the source domain and the target domain respectively, and $|Slot_{shared}|$ is the number of shared slots of the source domain and the target domain. For example, in Table 1, *timeRange* and *spatial_relation* are the shared slots of “GetWeather” and “FindScreeningEvent” domains. Then we define a function $f_w(\cdot)$ to transform the scores into weights as follows:

$$weight(sd, td) = f_w(score) = \alpha + \beta \cdot score(sd, td) \quad (7)$$

where α and β are the hyper-parameters and are used to tune the magnitude of similarity. For the utterances of different source domains, the learning rate is controlled by the similarity weight, which is computed as:

$$LR(sd, td) = \epsilon \cdot weight(sd, td) \quad (8)$$

where ϵ represents the initial learning rate for the source domain. Finally, $LR(sd, td)$ will be used to update the model parameters with the loss of the CRF layer in slot entity model.

3 Experiment

3.1 Dataset

To evaluate the efficiency of our proposed model, we conduct experiments on two benchmark datasets.

Table 1. Detailed statistics of SNIPS dataset.

Domain	Slots	
	Cross-domain shared	Domain-specific
AddToPlaylist	artist, playlist, music_item	playlist_owner, entity_name
BookRestaurant	country, state, timeRange, sort, spatial_relation, city	party_size_number, poi, restaurant_type, facility, party_size_description, served_dish, cuisine, restaurant_name
GetWeather	country, state, timeRange, city, spatial_relation	spacurrent_location, condition_description condition_temperature, geographic_poi
PlayMusic	sort, artist, playlist, music_item	year, album, genre, track, service
RateBook	object_type, object_name	object_part_of_series_type, rating_value, object_select, best_rating, rating_unit
SearchCreativeWork	object_type, object_name	-
FindScreeningEvent	timeRange, object_type, spatial_relation	object_location_type, movie_type, movie_name, location_name

Table 2. Detailed statistics of SMP-ECDT dataset.

Domain	Slots	
	Cross-domain shared	Domain-specific
cookbook	keyword	dishName, utensil, ingredient
epg	datetime_time, datetime_date, category, name, code, area	tvchannel
map	startLoc_poi, endLoc_poi, startLoc_city, endLoc_city, endLoc_province, endLoc_area, location_city, location_province, type, startLoc_area	location_area, location_poi
message	name, content, category, teleOperator	receiver, headNum
poetry	keyword, author, name	queryField, dynasty
train	startDate_date, category, startLoc_city, endLoc_city, startLoc_area, endLoc_area, startLoc_province, endLoc_province, startLoc_poi, startDate_time	-
video	name, category, timeDescr, area, popularity, artist	tag, scoreDescr

- **SNIPS.** We execute the experiments on the crowd-sourced benchmark corpus SNIPS [19] that widely used for slot filling. It is a public spoken language understanding dataset that contains 39 slot types across 7 domains (intents). The scheme corresponding to each domain is described in Table 1. To test our model, for each time, we choose one domain as the target domain and the other six domains as the source domains.
- **SMP-ECDT.** SMP-ECDT corpus^{1,2} consists of 29 domains and 124 slot labels. Due to the large number of domains and the small amount of data in each domain, we only selected the top 7 domains with the largest amount of data as target domains for the experiment. The statistics of these domains are listed in Table 2. For each time, we choose one domain as the target domain and the other 28 domains as the source domains.

¹ <http://conference.cipsc.org.cn/smp2019/evaluation.html>.² <https://github.com/OnionWang/SMP2019-ECDT-NLU>.

3.2 Baselines

We compare our model with the existing baselines:

- **Concept Tagger (CT).** Bapna et al. [11] utilized slot descriptions to fill slots for each slot type individually and cope with the unseen slot types.
- **Coarse-to-fine Approach (Coach).** Liu et al. [14] proposed a coarse-to-fine procedure with BIO 3-way classification and slot type prediction. It also further introduced a template regularization (TR) to improve the performance of similar or the same slot types. We use their best model Coach+TR to compare with but we call it Coach simply.
- **Contrastive Zero-Shot Learning with Adversarial Attack (CZSL-Adv).** He et al. [15] used contrastive loss to leverage auxiliary slot description information and introduced an adversarial attack (Adv) training strategy to improve model robustness. Since the paper does not provide the code of adversarial attack part, we only use the CZSL model to compare with in the experiments on SMP-ECDT dataset.

3.3 Implementation Details

For a fair comparison under cross-domain settings, we follow most of the set-ups in Liu et al. [14] and He et al. [15]. For all BiLSTM encoders, We set the hidden size to 200 and a dropout [20] rate to 0.3. Following Liu et al. [14], for every word in SNIPS, we concatenate the word-level [21] and character-level [22] embeddings. For SMP-ECDT datasets, we use the public Chinese pre-training character-level embeddings [23]. We combine the samples from all source domains for training, split 500 data samples in the target domain as the validation set for choosing the best model and the remainder are used for the test set.

Following Liu et al. [14], we used tokenized slot names as the slots descriptions of SNIPS (e.g., the description of slot label *restaurant_type* is “restaurant type”). For SMP-ECDT dataset, we also define a simple Chinese slot description for each slot type. For example, the slot description of “*TV_channel*” is “电视频道” (The Chinese word embedding of “TV channel”).

In instance weighting scheme, we set the hyper-parameters α and β to 0.2 and 15. We take the instance weighting scheme on the loss of the CRF layer. We use Adam optimizer [24] to optimize all parameters with a learning rate of 0.0005. We set the batch size to 32 and use the early stop of patience 5. All data shown in the following results are the average of several independent experiments.

3.4 Overall Results

We use F1 score to evaluate the performances on each domain. Table 3 and Table 4 show the experiment results of the proposed model on SNIPS and SMP-ECDT datasets respectively. PCD-Iw denotes our proposed model and PCD represents our model without instance weight scheme. Scores in each row represent the performance of the leftmost target domain.

Table 3. Slot F1-scores on SNIPS for different target domains under zero/few-shot learning settings. * indicates the significant improvement over all baselines ($p < 0.05$)

Training setting	Zero-shot						Few-shot on 50 samples					
	CT	Coach	CZSL	CZSL-Adv	PCD	PCD-Iw	CT	Coach	CZSL	CZSL-Adv	PCD	PCD-Iw
AddToPlaylist	38.82	50.90	53.29	53.89	52.84	55.83*	68.69	74.68	77.71	76.18	80.24	80.37*
BookRestaurant	27.54	34.01	37.97	34.06	36.84	38.41*	54.22	74.82	77.35	76.28	77.41*	76.59
GetWeather	46.45	50.47	48.70	52.24	56.04	59.80*	63.23	79.64	81.85	83.28	84.23	85.09*
PlayMusic	32.86	32.01	29.14	34.59	31.81	36.31*	54.32	66.38	65.59	68.17	66.44	69.76*
RateBook	14.54	22.06	29.55	31.53	30.26	28.25	76.45	84.62	84.31	87.22	89.16	89.49*
SearchCreativeWork	39.79	46.65	49.32	50.61	49.78	51.81*	66.38	64.56	66.41	66.49	70.00*	69.65
FindScreeningEvent	13.83	25.63	25.95	30.05	27.75	26.95	70.67	83.85	81.14	83.26	84.10	86.43*
Average F1	30.55	37.39	39.13	40.99	40.76	42.50*	64.85	75.51	76.34	77.27	78.80	79.63*

Table 4. Slot F1-scores on SMP-ECDT for different target domains under zero/few-shot learning settings. * indicates the significant improvement over all baselines ($p < 0.05$)

Training Setting	Zero-shot					Few-shot on 5 samples					
	Domain↓ Model→	CT	Coach	CZSL	PCD	PCD-Iw	CT	Coach	CZSL	PCD	PCD-Iw
cookbook		1.35	16.95	15.00	16.54	22.27*	3.47	38.07	43.18	48.62*	42.31
epg		9.50	18.84	20.54	25.41*	24.59	13.95	31.37	29.54	39.92	39.93*
map		16.75	22.15	23.42	22.95	26.66*	18.39	35.71	32.02	28.33	28.40
message		11.19	29.87	25.23	26.59	29.89*	30.86	33.87	34.86	31.79	36.63*
poetry		19.03	43.19	43.66	43.41	43.81	21.96	50.48	53.67	45.74	65.52*
train		84.58	85.71	85.09	83.96	84.05	84.95	85.16	85.14	86.65*	86.31
video		19.41	26.39	32.13	36.68*	32.53	22.14	30.56	30.82	34.42	35.07*
Average F1		23.21	34.73	35.01	36.50	37.69*	26.94	43.60	43.92	45.07	47.73*

From Table 3 and 4, we can observe that our model significantly outperforms all the baselines and achieves the state-of-the-art performance in the zero/few-shot settings. In the zero-shot setting, compared with the best prior work, PCD achieves 1.51% and 2.68% improvement on SNIPS dataset and SMP-ECDT dataset respectively. Moreover, since we did not use adversarial attack training to improve the robustness of the model, the PCD-Iw actually reaches 3.37% improvement on SNIPS dataset compared to the CZSL model (F1 score of 39.13). In the few-shot setting, PCD achieves 2.36% and 3.81% improvement in two datasets. In addition, without instance weighting scheme the PCD framework has also improved in every experiment setting. These results indicate the effectiveness of our proposed framework.

3.5 Analysis on Slot Entity Identification

Since our PCD-Iw approach especially the instance weighting scheme has a promotion effect on the identification of slot entities, we analyze this effect separately. The results are shown in Tables 5 and 6. The scores are calculated from our slot entity model and the first step in two-stage models [14, 15].

Table 5. BIO F1-scores on SNIPS for different target domains under zero/few-shot learning settings.

Training setting	Zero-shot				Few-shot on 50 samples			
	Domain↓ Model→	Coach	CZSL	PCD	PCD-Iw	Coach	CZSL	PCD
AddToPlaylist	57.06	57.43	61.77	65.43	79.08	80.98	84.93	85.17
BookRestaurant	59.49	59.51	60.39	65.29	82.56	83.75	84.63	87.06
GetWeather	57.14	59.76	66.24	71.15	79.95	84.96	89.58	90.28
PlayMusic	48.48	49.53	52.22	59.46	70.24	74.72	77.68	82.04
RateBook	32.23	38.13	34.39	35.87	86.69	89.56	89.10	90.08
SearchCreativeWork	48.88	48.27	49.97	54.66	66.69	67.90	70.41	71.47
FindScreeningEvent	37.73	40.71	44.24	46.43	84.02	85.19	84.93	88.22
Average F1	48.72	50.51	52.74	56.90	78.45	81.01	83.04	84.90

Table 6. BIO F1-scores on SMP-ECDT for different target domains under zero/few-shot learning settings.

Training setting	Zero-shot				Few-shot on 5 samples			
	Domain↓ Model→	Coach	CZSL	PCD	PCD-Iw	Coach	CZSL	PCD
cookbook	65.84	70.50	73.24	74.35	71.74	71.11	74.71	74.31
epg	29.72	34.89	41.46	42.70	38.61	35.28	49.19	49.28
map	53.15	57.00	55.95	57.32	56.21	52.63	54.50	56.25
message	38.35	33.50	36.17	44.58	39.16	40.73	42.06	46.47
poetry	51.15	53.08	51.41	52.05	53.50	54.68	52.69	74.86
train	92.13	89.70	91.04	89.20	91.26	89.65	93.00	93.57
video	32.39	41.58	42.23	42.51	35.12	36.79	40.91	44.15
Average F1	51.82	54.32	55.93	57.52	55.09	54.41	58.15	62.70

As can be seen from Table 5 and 6, our model achieves the state-of-the-art performance in almost all domains under zero/few-shot settings. In the zero-shot setting, PCD achieves 6.39% and 3.40% improvement on SNIPS dataset and SMP-ECDT dataset respectively. In the few-shot setting, PCD achieves 3.89% and 8.29% improvement in two datasets. The result also verifies our assumptions that instance weighting scheme can be used for alleviating the problem of negative transfer and improving the performance of capturing slot entities.

3.6 Ablation Study

From Table 3 and 4, we can see that compared with PCD-Iw model, the PCD model has a performance decline of 1.0%-2.5%, which indicates that both our PCD model and the instance weighting scheme have an improvement effect. As can be seen from Table 5 and 6, compared with the PCD-Iw model, the

performance of the PCD model in identifying slot entities decreases by 1.5% to 5.0%. The relatively high gap indicates that the instance weighting scheme has a more significant improvement in the identification of slot entities.

3.7 Analysis on Seen and Unseen Slots

Following the baselines setting, we also split the test set into “unseen” and “seen” parts. Table 7 shows the results on seen and unseen slots in two datasets. We can observe that our approach consistently outperforms the baselines both on the unseen and seen slots in the two settings and two datasets. Our pipeline model is to promote all the slot types and the instance weighting scheme also alleviates the problem of negative transfer. Therefore, our approaches generally improve on both unseen and seen slot types.

Table 7. Average F1-scores on SNIPS and SMP-ECDT for seen and unseen slots across all target domains.

Dataset	SNIPS				SMP-ECDT			
Setting	0 sample		50 samples		0 sample		5 samples	
	unseen	seen	unseen	seen	unseen	seen	unseen	seen
CT	27.10	44.18	62.05	69.64	11.85	30.95	18.29	34.64
Coach	34.09	51.93	76.49	80.16	18.98	44.15	31.45	44.78
CZSL	34.57	52.69	77.15	80.09	17.05	46.74	32.74	43.41
CZSL-Adv	36.35	55.43	78.48	79.36	—	—	—	—
PCD	35.79	55.63	78.84	80.75	20.73	48.70	29.84	46.29
PCD-Iw	36.98	56.96	80.61	81.66	21.12	49.08	39.76	49.17

4 Conclusions

In this paper, we propose a new pipeline approach with distinct slot entity and type prediction for cross-domain slot filling. Our approach consists of a slot entity identification model and slot type alignment model, which uses distinct contextual representations for learning and boundary markers for connecting two sub models. Moreover, we introduce an effective instance weighting scheme to control the contribution of different source domains by adjusting learning rates. Experiments show that our approach significantly outperforms existing cross-domain slot filing models, especially in the accuracy of slot entity identification.

Acknowledgements.. This work was supported by Natural Science Foundation of Guangdong Province (No. 2021A1515011864), Guangzhou Key Laboratory of Intelligent Agriculture (No. 201902010081), National Natural Science Foundation of China (No. 71472068), Educational Commission of Guangdong Province of China (No. 2020KTSCX01) and Innovation Training Project for College Students of Guangdong Province (No. S202110564051, No. S202010564169).

References

1. Sarikaya, R.: The technology behind personal digital assistants: an overview of the system architecture and key components. *IEEE Signal Process. Mag.* **34**(1), 67–81 (2017)
2. Liu, B., Lane, I.: Attention-based recurrent neural network models for joint intent detection and slot filling. In: Proceedings of the 17th Annual Conference of the International Speech Communication Association (INTERSPEECH 2016), San Francisco, CA, USA, 8–12 September, pp. 685–689 (2016)
3. Goo, C., et al.: Slot-gated modeling for joint slot filling and intent prediction. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018), New Orleans, Louisiana, USA, 1–6 June, Volume 2 (Short Papers), pp. 753–757 (2018)
4. Haihong, E., Niu, P., Chen, Z., Song, M.: A novel bi-directional interrelated model for joint intent detection and slot filling. In: Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL 2019), Florence, Italy, 28 July–2 August, Volume 1: Long Papers, pp. 5467–5471 (2019)
5. Qin, L., Che, W., Li, Y., Wen, H., Liu, T.: A stack-propagation framework with token-level intent detection for spoken language understanding. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019), Hong Kong, China, 3–7 November, pp. 2078–2087 (2019)
6. Jaech, A., Heck, L.P., Ostendorf, M.: Domain adaptation of recurrent neural networks for natural language understanding. In: Proceedings of the 17th Annual Conference of the International Speech Communication Association (INTERSPEECH 2016), San Francisco, CA, USA, 8–12 September, pp. 690–694 (2016)
7. Guo, J., Shah, D.J., Barzilay, R.: Multi-source domain adaptation with mixture of experts. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018), Brussels, Belgium, 31 October–4 November, pp. 4694–4703 (2018)
8. Lin, B.Y., Lu, W.: Neural adaptation layers for cross-domain named entity recognition. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018), Brussels, Belgium, 31 October–4 November, pp. 2012–2022 (2018)
9. Liu, Z., et al.: Zero-shot cross-lingual dialogue systems with transferable latent variables. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019), Hong Kong, China, 3–7 November, pp. 1297–1303 (2019)
10. Liu, Z., Winata, G.I., Fung, P.: Zero-resource cross-domain named entity recognition. In: Proceedings of the 5th Workshop on Representation Learning for NLP (RepL4NLP@ACL 2020), Online, 9 July, pp. 1–6 (2020)
11. Bapna, A., Tiir, G., Hakkani-Tür, D., Heck, L.P.: Towards zero-shot frame semantic parsing for domain scaling. In: Proceedings of the 18th Annual Conference of the International Speech Communication Association (INTERSPEECH 2017), Stockholm, Sweden, 20–24 August, pp. 2476–2480 (2017)
12. Lee, S., Jha, R.: Zero-shot adaptive transfer for conversational language understanding. In: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI 2019) Honolulu, Hawaii, USA, 27 January–1 February, pp. 6642–6649 (2019)

13. Shah, D.J., Gupta, R., Fayazi, A.A., Hakkani-Tür, D.: Robust zero-shot cross-domain slot filling with example values. In: Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL 2019), Florence, Italy, 28 July–2 August, Volume 1: Long Papers, pp. 5484–5490 (2019)
14. Liu, Z., Winata, G.I., Xu, P., Fung, P.: Coach: A coarse-to-fine approach for cross-domain slot filling. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020), Online, 5–10 July, pp. 19–25 (2020)
15. He, K., Zhang, J., Yan, Y., Xu, W., Niu, C., Zhou, J.: Contrastive zero-shot learning for cross-domain slot filling with adversarial attack. In: Proceedings of the 28th International Conference on Computational Linguistics (COLING 2020), Barcelona, Spain (Online), 8–13 December, pp. 1461–1467 (2020)
16. Pan, S.J., Yang, Q.: A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **22**(10), 1345–1359 (2010)
17. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: 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 (NAACL-HLT 2016), San Diego California, USA, 12–17 June, pp. 260–270 (2016)
18. Zhong, Z., Chen, D.: A frustratingly easy approach for entity and relation extraction. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2021), Online, 6–11 June, pp. 50–61 (2021)
19. Coucke, A., et al.: Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *CoRR* abs/1805.10190 (2018)
20. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014)
21. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguistics* **5**, 135–146 (2017)
22. Hashimoto, K., Xiong, C., Tsuruoka, Y., Socher, R.: A joint many-task model: growing a neural network for multiple NLP tasks. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017), Copenhagen, Denmark, 9–11 September, pp. 1923–1933 (2017)
23. Li, S., Zhao, Z., Hu, R., Li, W., Liu, T., Du, X.: Analogical reasoning on Chinese morphological and semantic relations. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, (ACL 2018), Melbourne, Australia, 15–20 July, Volume 2: Short Papers, pp. 138–143 (2018)
24. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015

Social Media and Sentiment Analysis



Semantic Enhanced Dual-Channel Graph Communication Network for Aspect-Based Sentiment Analysis

Zehao Yan, Shiguan Pang, and Yun Xue^(✉)

School of Physics and Electronic Engineering, South China Normal University,
Guangzhou, China
{yzh_scnu,psg-nlp,xueyun}@m.scnu.edu.cn

Abstract. Aspect-based sentiment analysis is a fine-grained task that aims to clarify the sentiment polarity of a given aspect in a sentence, whose main challenge is to model the relation between the aspects and its opinion words. Seeing that the analysis based on dependency tree has its deficiencies, a Semantic Enhanced Dual-channel Graph Communication Network is proposed to address such issues. In our model, the semantic information is captured to supplement syntactic features while the communication mechanism and the hierarchical attention module are employed to obtain the word representation. The working performance of the proposed model is evaluated on publicly available datasets. Experimental results reveal that our model significantly outperforms the baseline methods and achieves advanced results in ABSA tasks.

Keywords: Aspect-based sentiment analysis · Graph neural network · Dependency tree

1 Introduction

Aspect-based sentiment analysis (ABSA) is recently a major interest in the field of natural language processing (NLP) [14, 22]. As a fine-grained task, ABSA targets at identifying the sentiment polarity of a specific aspect in a given text. Within the same sentence, it is often the case that multiple aspects carrying sentiment are different. Therefore, the main challenge of this task is how to model the interaction between aspects and its opinion words.

On the task of ABSA, these years have witnessed the progressing of graph convolution network (GCN) due to its superiority in processing syntactic structure [7]. The GCN-based methods set the foundation of capturing the syntactic information from syntax dependency trees, with recent publications reporting their effectiveness in sentence encoding [13, 15, 24]. This becomes an ongoing trend where syntactic information is exploited as a primary basis for sentiment

Z. Yan and S. Pang—Equal contribution.

analysis. Specifically, the integration of GCN and syntax dependency tree provides an opportunity to precisely establish the relation between the aspect and its opinion words, which addresses the issue of long-distance dependency [9, 18].

The application of GCN-based methods are, however, still limited. For one thing, in sentences that are absent of syntactic features, the syntax structure analysis can result in low sentiment classification accuracy. According to Fig. 1, for the aspect ‘*psp*’, the syntax dependency tree contains a number of noises (e.g. Despite the connection, the word ‘*put*’ is irrelevant to ‘*psp*’). For another, both syntax and semantics in one sentence has an effect on each other, which gives rise to dealing with the interaction between them. As explained in [12], ‘Syntactic effects are difficult to distinguish from semantic effects, because in natural language, syntactic changes usually alter the meaning of the expression’.

In line with the significance of semantics, current research focuses on integrating the semantic information to enhance the establishing of syntactic structure. The computing of cosine similarity is one such approach, owing to its learning to construct the semantic similarity map [14, 25]. Notwithstanding, the cosine similarity has the deficiency of insensitivity to the absolute value of distance, which results in the meaningless connection between unrelated words. By contrast, the attention mechanism is highlighted considering its capability of resolving the relation between different words [3, 19, 22].

In this work, focusing on integrating the semantic information and the syntactic information, A Semantic Enhanced Dual-channel Graph Communication Network (SDGCN) for ABSA is thus proposed. An attention-based K-Head cosine similarity model, by applying attention mechanism to the basic semantic similarity graph, is proposed to characterize the semantic connection and the connection weight among words. A graph communication unit is also developed to learn the relation between syntax and semantics. Since each layer of GCN tends to obtain a specific representation, a hierarchical aspect-based attention module is devised and exploited in GCN, which captures and fuses the representation of aspects and contexts within each layer. Besides, we compare our models against baseline methods on publicly available datasets to investigate its working performance.

The contributions of this paper can be summarized as follows:

- Based on theory of cosine similarity, an attention-based K-Head cosine similarity is proposed and applied to construct the semantic similarity graph. In addition, a graph communication module that interactively learns the syntax and the semantics is devised to facilitate the integration of semantic information into syntactic structure.
- A hierarchical aspect-based attention module is developed to fully capture the representation of both the aspect and its contexts for further processing, which improves the encoding effectiveness.
- The experimental results reveal that SDGCN is a competitive alternative due to its efficiently integrating the semantic information and the syntactic information.

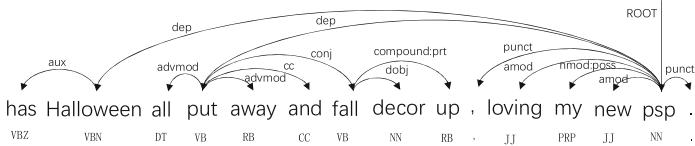


Fig. 1. Syntax dependency tree of aspect ‘psp’.

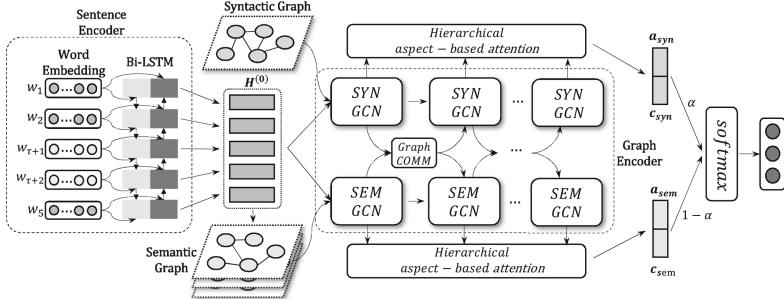


Fig. 2. Architecture of SDGCN

2 Related Work

So much is the significance of ABSA in NLP that an increasing number of methods concentrate on modeling the relation between aspect and opinion words and thus identifying its sentiment polarity. Generally, current ABSA methods can be further divided into two categories, i.e. semantic-based methods and syntactic-based methods.

Semantic-Based Methods: For the purpose of devising a model to tackle the texts that is distinctly human, more attention is paid to analyze every individual word and the semantic components associated with it [2]. In this way, the attention mechanism, together with its integration into deep learning methods, is most pronounced.

Aiming to establish aspect-orientated sentiment representation, [19] and [22] apply the attention mechanisms to the bidirectional long short-term memory (Bi-LSTM) network. [3] devise a multi-head attention-based method, which works on capturing sentiment features to address the long-distance relation constructing. [21] integrate both the attention mechanism and the gate mechanism into CNN to precisely capture the aspect-related information from the context. More recently, advances in pre-training models, such as transformer and Bert, significantly improve the capabilities of ABSA methods [16, 20]. State-of-arts results are obtained via extensive training.

Syntactic-Based Methods: The analysis of sentence syntax efficiently establish a bridge between the aspect and its opinion words, which resolve the

long-distance dependency fundamentally. The employment of syntactic information in ABSA paves a way for accurately setting the connection of the aspect with its sentiment [4, 8, 9, 18]. Basically, the syntax dependency tree is constructed for syntactic information performing and conveying [4]. More recently, GCN-based models receive growing attention in NLP, with the integrating with syntax dependency tree as a primary choice of syntax analysis [26]. Seeing that GCN is capable of dealing with graph data, the syntax dependency graphs, containing rich relation information, are constructed [13, 23]. [17] exploit GAT to re-constructed the syntax dependency tree and thus revise the relation between aspect and opinion words. [15] devise a dependency graph enhanced dual-transformer network, which combines the flat representations learnt from Transformer and graph-based representations learnt from the corresponding dependency graph. In addition, [24] clarify various types of dependency relations and lexical word pairs by convoluting over hierarchical syntactic and lexical graphs.

3 Methods

The architecture of SDGCN is shown in Fig. 2. There are three main components of our model, i.e. a sentence encoder, a graph encoder, a hierarchical aspect-based attention module and a sentiment classifier. We start with describing the working principle of GCN. Then each component of the SDGCN will be presented in detail.

3.1 Graph Convolutional Network

Fundamentally, GCN is proposed to tackle graph-structured data [7]. Let $G = (V, E, A)$ be a target graph with V , E and A as the collections of node, edges and adjacent matrices of the graph, respectively. For $V = [v_i]_{i=1}^n$, the encoding of graph is delivered as:

$$H^{(l+1)} = \sigma(\tilde{A}H^{(l)}W^{(l+1)}) \quad (1)$$

where \tilde{A} is the symmetric normalized adjacency matrix of $A + I$, \tilde{D} is the degree matrix of \tilde{A} , $H^{(l)}$ refers to the node collection, $W^{(l+1)} \in \mathbb{R}^{d_l \times d_{l+1}}$ is the weight matrix of the l -th layer, and σ is a nonlinear activation function, such as ReLU.

3.2 Sentence Encoder

Given an n -word sentence $s = [w_1, w_2, \dots, w_{\tau+1}, \dots, w_{\tau+m}, \dots, w_n]$ with aspect $[w_{\tau+1}, w_{\tau+2}, \dots, w_{\tau+m}]$ in it, we map each word w_i into a low-dimensional vector by looking up in a pretrained word embedding matrix $E \in \mathbb{R}^{|V| \times d_e}$ where $|V|$ is the lexicon size and d_e is the dimension of word vector [1]. The hidden states of sentence s are extracted via Bi-LSTM.

Besides, we present the encoding process by using Bert as well. The sequence “[CLS] s [SEP] a [SEP]” is sent to the encoder to obtain the representation of a specific aspect. Based on the token-level encoding mechanism, the first token embedding of each word is taken as the corresponding word-level representation.

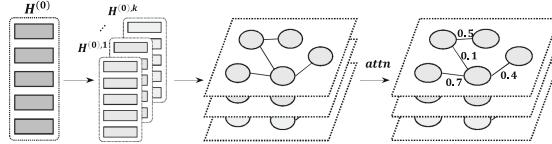


Fig. 3. Constructing of the semantic graph

In such a manner, the output of sentence encoder is $H^{(0)} = [h_1^{(0)}, h_2^{(0)}, \dots, h_n^{(0)}]$ with aspect $[h_{\tau+1}^{(0)}, \dots, h_{\tau+m}^{(0)}]$.

3.3 Graph Encoder

According to Fig. 2, the graph encoder consists of syntax-based GCNs, semantic-based GCNs and graph communication modules. Both the syntactic graph and the semantic graph are constructed and sent to the syntax-based GCNs and the semantic-based GCNs, respectively.

Syntax-Based GCN: In line with [13], each word in the given sentence is a node while each syntax dependency is an edge. The syntax dependency tree is thus transformed into a syntax graph, i.e. $G_{syn} = (A_{syn}, H^{(0)})$. The adjacent matrix $A_{syn} \in \mathbb{R}^{n \times n}$ can be obtained, which is

$$A_{syn}(i, j) = \begin{cases} 1 & \text{if } i \leftrightarrow j \\ 0 & \text{others} \end{cases} \quad (2)$$

where $i \leftrightarrow j$ indicates the mutual dependency between node i and node j .

The syntactic information of G_{syn} is captured via the Syntax-based GCN:

$$H_{syn}^{(l+1)} = \sigma \left(\tilde{A}_{syn} H_{syn}^{(l)} W_{syn}^{(l+1)} \right) \quad (3)$$

where $H_{syn}^{(l+1)} \in \mathbb{R}^{n \times d}$, the initialized input is $H_{syn}^{(0)} = H^{(0)}$, $W_{syn}^{(l+1)} \in \mathbb{R}^{n \times d}$ represents the weight matrix of the l -th layer of GCN and d is the output dimension of GCN.

Semantic-Based GCN: As pointed out in the Introduction, for sentences of irregular syntax structure, the syntactic analysis can fail to a sentiment classification result. That is, more information has to be exploited rather than merely studying the syntax structure. To this end, an attention-based K-Head cosine similarity method is proposed (Fig. 3), which aims to capture the semantic relation within sentence to supplement syntactic information. To start with, $H^{(0)}$ is mapped into K different d -dimensional semantic vectors to obtain the semantic features, which is:

$$H^{(0),k} = \sigma \left(H^{(0)} W_H^k + b_H^k \right) \quad (4)$$

where $H^{(0),k} = [h_1^{(0),k}, h_2^{(0),k}, \dots, h_n^{(0),k}]$, $k \in [1, K]$, $W_H^k \in \mathbb{R}^{d \times d}$ and $b_H^k \in \mathbb{R}^{1 \times d}$ stand for the mapping matrix and the bias vector, respectively.

Subsequently, since the semantic relation among distinguishing words are different, the self-attention mechanism is employed to automatically learn the semantic connection weights between words. Hence, the semantic graph G_{sem} is generated.

Specifically, the semantic connecting relation between node w_i and node w_j is delivered as:

$$A_{sem}[i, j] = \Gamma(i, j) \quad (5)$$

where $A_{sem} \in \mathbb{R}^{n \times n}$ and $\Gamma(i, j)$ is computed by:

$$\Gamma(i, j) = \frac{1}{K} \sum_{k=1}^K attn_{i,j}^k \cdot a_{i,j}^k \quad (6)$$

$$a_{i,j}^k = \begin{cases} 1 & \text{if } \cos(h_i^{(0),k}, h_j^{(0),k}) > \rho \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

$$attn_{i,j}^k = \frac{(h_i^{(0),k} W_1^k) (h_j^{(0),k} W_2^k)^T}{\sqrt{d}} \quad (8)$$

with $W_1^k \in \mathbb{R}^{d \times d}$ and $W_2^k \in \mathbb{R}^{d \times d}$ standing for the self-attention weight matrices.

Based on the G_{sem} , the semantic information is captured via GCN, which is:

$$H_{sem}^{(l+1)} = \sigma(\tilde{A}_{sem} H_{sem}^{(l)} W_{sem}^{(l+1)}) \quad (9)$$

where $H_{sem}^{(l+1)} \in \mathbb{R}^{n \times d}$, $H_{sem}^{(0)} = H^{(0)}$ and $W_{sem}^{(l+1)} \in \mathbb{R}^{n \times d}$ represents the weight matrix of the l -th layer of GCN.

Graph Communication: The graph communication unit is developed to perform the interaction between the syntax and the semantics. As such, both the syntactic information and the semantic information are learned and integrated for further processing. The communication between the inputs $H_{syn}^{(l+1)}$ and $H_{sem}^{(l+1)}$ is carried out by the characteristic interaction as follows:

$$H'_{syn}, H'_{sem} = COMM(H_{sem}^{(l+1)}, H_{syn}^{(l+1)}) \quad (10)$$

$$H'_{syn} = U_{syn} H_{sem}^{(l+1)} \quad (11)$$

$$H'_{sem} = U_{sem} H_{syn}^{(l+1)} \quad (12)$$

$$U_{syn} = softmax(H_{syn}^{(l+1)} W_3 H_{sem}^{(l+1)T} + b_3) \quad (13)$$

$$U_{sem} = softmax(H_{sem}^{(l+1)} W_4 H_{syn}^{(l+1)T} + b_4) \quad (14)$$

where $W_3 \in \mathbb{R}^{d \times d}$ and $W_4 \in \mathbb{R}^{d \times d}$ are weight matrices while b_3 and b_4 are biases.

Furthermore, we have $H_{syn}^{(l+1)}$ and $H_{sem}^{(l+1)}$ updated by using H'_{syn} and H'_{sem} , which can be written as:

$$H_{syn}^{(l+1)} = H_{syn}^{(l+1)} + H'_{syn} \quad (15)$$

$$H_{sem}^{(l+1)} = H_{sem}^{(l+1)} + H'_{sem} \quad (16)$$

3.4 Hierarchical Aspect-Based Attention

Based on the aforementioned processing, each layer within the GCN obtains a distinctive representation. On this stage, we shall apply a hierarchical aspect-based attention module to establish more accurate representations of both the aspect and the contexts.

Let $H_{syn}^{(1)}, H_{syn}^{(2)}, \dots, H_{syn}^{(L)}$ be the syntactic feature sequence. Considering $h_{syn,i}^{(l)}$ as the i -th node feature in l -th layer and $h_{syn,i}^{(L)}$ as that in L -th layer, the attention weight $\alpha_{l,i}$ between these nodes is computed as:

$$\alpha'_{l,i} = h_{syn,i}^{(l)} \left(h_{syn,i}^{(L)} \right)^T \quad (17)$$

$$\alpha_{l,i} = \frac{\exp(\alpha'_{l,i})}{\sum_{j=1}^{L-1} \exp(\alpha'_{j,i})} \quad (18)$$

Notably, a larger α_l refers to a more important l -th layer.

The syntactic feature $H_{syn} \in \mathbb{R}^{n \times d}$ is obtained via the weighted sum of the syntactic representation from each layer, i.e.

$$H_{syn} = [h_{syn,1}, h_{syn,2}, \dots, h_{syn,n}] \quad (19)$$

$$h_{syn,i} = h_{syn,i}^{(L)} + \sum_{l=1}^{L-1} \alpha_{l,i} h_{syn,i}^{(l)} \quad (20)$$

The specific syntactic representation of aspect and context can be extracted from H_{syn} in Eq. (19). The attention mechanism is computationally efficient to capture the aspect-related words from C_{syn} . Then we have that:

$$a_{syn} = \text{AvgPool}([h_{syn,i}] \mid i \in (\tau, \tau + m)] \quad (21)$$

$$C_{syn} = [h_{syn,j}] \mid j \in [1, \tau] \cup (\tau + m, n] \quad (22)$$

$$U'_{syn} = \text{softmax}(a_{syn} W_{ac} C_{syn}^T) \quad (23)$$

$$c_{syn} = \text{AvgPool}(U'_{syn} \cdot C_{syn}) \quad (24)$$

where $a_{syn} \in \mathbb{R}^{1 \times d}$, $C_{syn} \in \mathbb{R}^{(n-m) \times d}$, $c_{syn} \in \mathbb{R}^{1 \times d}$, $U'_{syn} \in \mathbb{R}^{1 \times (m-n)}$, $\text{AvgPool}(\cdot)$ stands for the average pooling, $W_{ac} \in \mathbb{R}^{d \times d}$ is the trainable weight matrix.

Likewise, the semantic feature sequence $H_{sem}^{(1)}, H_{sem}^{(2)}, \dots, H_{sem}^{(L)}$ is processed in a same manner. As presented in Eq. (21)–Eq. (24), the semantic representations of the aspect and the context, i.e. a_{sem} and c_{sem} , are therefore computed.

Lastly, a trainable parameter α and a linear layer are proposed for the fusion of syntactic and semantic features. The final representation $h \in \mathbb{R}^{1 \times d}$ is given by

$$h' = \alpha [a_{syn}, c_{syn}] + (1 - \alpha) [a_{sem}, c_{sem}] \quad (25)$$

$$h = \sigma(h' W_5 + b_5) \quad (26)$$

with $h' \in \mathbb{R}^{1 \times 2d}$. W_5 and b_5 represent the trainable weight matrix and the bias, respectively.

3.5 Model Training

The final representation is sent to a fully connected softmax layer for sentiment classification. The sentiment distribution of the given aspect is identified as:

$$\hat{y} = \text{softmax}(hW^T + b) \quad (27)$$

where both W and b are trainable weight matrices.

The model training is carried out by using the standard gradient descent algorithm with cross entropy and L_2 regularization. The loss function is given by:

$$L = - \sum_{i \in D} \sum_{j \in P} y_j^i \log \hat{y}_j^i + \lambda \|\theta\|_2 \quad (28)$$

where D represents the training dataset, P refers to the sentiment classes, y is the ground truth and \hat{y} is the predicted one. Besides, λ is the regularization coefficient while θ indicates the collection of all trainable parameters.

4 Experiments

4.1 Datasets

Experiments are conducted on three public datasets, which are Rest14 and lap14 from SemEval 2014 Task4 [11] and Twitter [4]. All the samples in our experiments are labeled as three different polarities, i.e. positive, neutral and negative. Details of each dataset are exhibited in Table 1.

4.2 Experimental Settings

In this experiment, the syntax dependency trees of each sentence is constructed using Stanford parser (<https://stanfordnlp.github.io/CoreNLP/>). For the basic SDGCN, word embeddings from all datasets are initialized using 300-dimensional word vectors pretrained by Glove [10]. Furthermore, the pre-training using Bert is also conducted, based on which the last hidden state is taken as $H^{(0)}$. According to [13], a 30-dimensional part-of-speech (POS) embedding and a 30-dimensional position embedding are integrated into our word embeddings to enrich the word representations. Besides, We take both sentiment classification accuracy and Macro-F1 as the evaluation indices. More details of the parameter setting are available in our code¹.

4.3 Baselines

Aiming to verify the effectiveness of the proposed model for the ABSA, 12 comparative models are adopted. Notably, the baselines can be subdivided into two categories as follows:

¹ Data and code can be found at <https://github.com/xiaodou12046/SDGCN>.

Table 1. Statistics of datasets.

Dataset	Positive		Negative		Neutral	
	Train	Test	Train	Test	Train	test
Rest14	2164	728	807	196	637	196
Laptop14	994	341	870	128	464	169
Twitter	1561	173	1560	173	3127	346

Table 2. Sentiment classification results.

Category	Method	Twitter		Lap14		Rest14	
		ACC	F1	ACC	F1	ACC	F1
Sem.	ATAE-LSTM	–	–	68.70	–	77.20	–
	RAM	69.36	67.3	74.49	71.35	80.23	70.80
	MGAN	72.54	70.81	75.39	72.47	81.25	71.94
	GCAE	–	–	69.14	–	77.28	–
Syn.	LSTM+SynATT	–	–	72.57	69.13	80.45	71.26
	TD-GAT	–	–	74.00	–	80.35	–
	ASGCN	72.15	70.40	75.55	71.05	80.77	72.02
	CDT	74.66	73.66	77.19	72.99	82.30	74.02
	BiGCN	74.16	73.35	74.59	71.84	81.97	73.48
	R-GAT	75.57	73.82	77.42	73.76	83.30	76.08
	RepWalk	74.4	72.6	78.2	74.3	83.8	76.9
	DGEDT	74.8	73.4	76.8	72.3	83.9	75.1
Ours	SDGCN	75.79	74.32	78.95	75.76	83.98	76.74
	-w/o H_{syn}	74.94	73.72	77.37	74.00	82.37	73.15
	-w/o H_{sem}	73.38	72.26	77.05	73.54	83.27	75.11
	-w/o COMM	74.37	71.71	77.21	73.31	81.93	72.1
	-w/o HAA	73.52	72.75	77.53	74.02	82.19	73.22
With BERT	R-GAT+BERT	76.15	74.88	78.21	74.07	86.60	81.35
	DGEDT+BERT	77.9	75.4	79.5	75.6	86.3	80.0
Ours	SDGCN+BERT	78.49	77.18	80.38	77.22	87.22	81.51

* ‘–’ indicates the result is not available;

‘-w/o H_{syn} ’: removing syntax-based GCN;‘-w/o H_{sem} ’: removing semantic-based GCN;

‘-w/o COMM’: removing graph communication unit;

‘-w/o HAA’: removing hierarchical aspect-based attention module.

Semantic-Based Methods: ATAE-LSTM [19], RAM [3], MGAN [5], GCAE [21].

Syntax-Based Methods: LSTM+SynATT [6], ASGCN [23], CDT [13], TD-GAT [21], BiGCN [24], R-GAT [17], RepWalk [27], DGEDT [15].

4.4 Results

Table 2 shows the results of the ABSA tasks carried out using all datasets. Generally, one can easily see that the syntax-based methods have a more satisfying performance than the semantic-based methods. Comparing to the syntax-based methods, our model is more competitive in all evaluation settings. In terms of the classification accuracy, the performance gaps between SDGCN and the second best model are 0.75% (Lap14), 0.08% (Rest 14) and 0.22% (Twitter), respectively. Sentences from Lap14 contain abundant syntactic and semantic information, based on which SDGCN outperforms other models by constructing and fusing the representations for sentiment classification. By contrast, the dataset Rest 14 is far less informative than Lap14. Clearly, since current encoders are capable of capturing the syntax structure from Rest 14, the improvement of our model on this dataset is limited. Besides, seeing that sentences from Twitter are typically absent of syntactic features, SDGCN effectively exploits the semantic information to supplement syntactic features. In this way, a higher accuracy is therefore obtained.

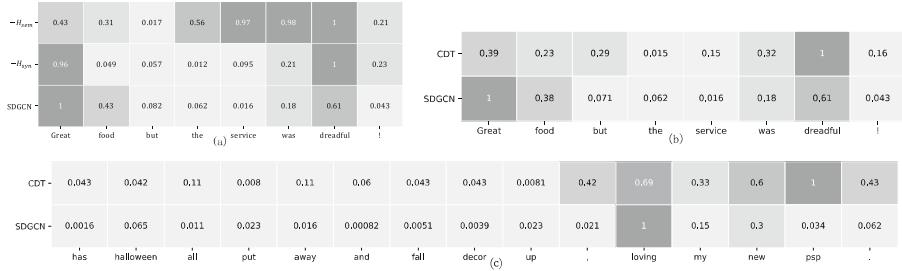


Fig. 4. (a)Word relevance scores with the aspect word ‘food’. (b) (c)Word relevance scores for CDT and SDGCN with the aspect word ‘food’ and ‘psp’, respectively.

With the pre-training of Bert, our model is still the best-performing method in this group. There exist a range of increase from 0.59% to 0.88% between SDGCN and other models. The employment of Bert further gives an improvement in the sentiment classification results. Since our model shows the superiority in integrating the syntax and the semantics of the sentence, it is reasonable to expect better working performance in ABSA, as it is the case.

4.5 Ablation Study

For the purpose of determining the significance of the different components in SDGCN, an ablation study is carried out. The basic SDGCN is taken as a baseline model.

The results in Table 2 show that the removal of graph encoders, i.e. syntax-based GCNs and semantic-based GCNs, results in the drop of working performance in all datasets. For Rest14, the ablating of syntax-based GCN has a

greater impact than semantic-based GCNs while otherwise for Lap14 and Twitter. It establishes a strong evidence that the semantic information can be taken to supplement syntactic features for a better performance. Moreover, the worst experimental result comes from the removal of graph communication unit, which indicates the significance of integrating the semantics and the syntax. Similarly, the ablating of hierarchical aspect-based attention module also causes a considerable accuracy decrease. That is, each component in SDGCN makes a distinctive contribution to the model.

4.6 Case Study

The mask experiment Eq. (29) [13] is performed to compute the contribution of each word in the sentence, targeting at investigating the significance of semantics and syntax processing.

$$\gamma(w, s) = \frac{1}{m} \sum_{i=1}^d \left| h_{(s)}^i - h_{(s/w)}^i \right| \quad (29)$$

where $h_{(s/w)}$ represents the final representation of sentence s with the masking of word w while $h_{(s)}$ stands for the basic representation of s . For $m = \max_{w \in s} \gamma(w, s)$, supposing that $\gamma(w, s) = 0$, we can conclude that w makes no contribution to $h_{(s)}$.

In Fig. 4(a), the ground-truth sentiment for the aspect word ‘*food*’ is positive. Based solely on the syntactic processing module, focus is given to the word ‘*dreadful*’ instead of its opinion word ‘*great*’. Furthermore, the application of semantic-based GCN identifies the word ‘*great*’ as a marginal contribution to the aspect. By contrast, SDGCN assigns more attention to the opinion word ‘*great*’ than ‘*dreadful*’ by integrating the syntactic information and the semantic information. Thus, the sentiment polarity of the aspect ‘*food*’ is classified as positive. The comparison on relevance scores between SDGCN and CDT is presented in Fig. 4(b), which demonstrates the effectiveness of our model.

Likewise, according to Fig. 4(c), the relevance scores of ‘*loving*’ and ‘*new*’ are comparable to the aspect ‘*psp*’. In contrast, SDGCN can precisely recognize the opinion word ‘*loving*’ and remove the noise from unrelated words. With the integration of syntactic information and semantic information, the proposed model significantly improves the sentiment analysis results comparing to the state-of-arts.

5 Conclusion

In this work, a Semantic Enhanced Dual-channel Graph Communication Network (SDGCN) is established on the task of ABSA. Based on GCN, our model is capable of capturing and integrating the syntax and the semantics of the sentence. The communication mechanism and the hierarchical attention network are also applied to obtain the word representation for sentiment classification.

Experiments are conducted on a variety of datasets to validate that our model obtains the best and most consistent results comparing to state-of-art methods. In this way, an even higher sentiment classification accuracy is achieved.

References

1. Bengio, Y., Ducharme, R., Vincent, P., Janvin, C.: A neural probabilistic language model. *J. Mach. Learn. Res.* **3**, 1137–1155 (2003)
2. Brooke, J.: A semantic approach to automated text sentiment analysis. Ph.D. thesis, Department of Linguistics-Simon Fraser University (2009)
3. Chen, P., Sun, Z., Bing, L., Yang, W.: Recurrent attention network on memory for aspect sentiment analysis. In: EMNLP, pp. 452–461 (2017)
4. Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., Xu, K.: Adaptive recursive neural network for target-dependent Twitter sentiment classification. In: ACL, pp. 49–54 (2014)
5. Fan, F., Feng, Y., Zhao, D.: Multi-grained attention network for aspect-level sentiment classification. In: EMNLP, pp. 3433–3442 (2018)
6. He, R., Lee, W.S., Ng, H.T., Dahlmeier, D.: Effective attention modeling for aspect-level sentiment classification. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 1121–1131 (2018)
7. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint [arXiv:1609.02907](https://arxiv.org/abs/1609.02907) (2016)
8. Liu, K., Xu, H.L., Liu, Y., Zhao, J.: Opinion target extraction using partially-supervised word alignment model. In: IJCAI, vol. 13, pp. 2134–2140 (2013)
9. Nguyen, T.H., Shirai, K.: PhraseRNN: phrase recursive neural network for aspect-based sentiment analysis. In: EMNLP, pp. 2509–2514 (2015)
10. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: EMNLP, pp. 1532–1543 (2014)
11. Pontiki, M., Papageorgiou, H., Galanis, D., Androutsopoulos, I., Pavlopoulos, J., Manandhar, S.: SemeVal-2014 task 4: aspect based sentiment analysis, p. 27 (2014)
12. Pylkkänen, L.: The neural basis of combinatory syntax and semantics. *Science* **366**(6461), 62–66 (2019)
13. Sun, K., Zhang, R., Mensah, S., Mao, Y., Liu, X.: Aspect-level sentiment analysis via convolution over dependency tree. In: EMNLP-IJCNLP, pp. 5683–5692 (2019)
14. Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., Qin, B.: Learning sentiment-specific word embedding for Twitter sentiment classification. In: ACL, pp. 1555–1565 (2014)
15. Tang, H., Ji, D., Li, C., Zhou, Q.: Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification. In: ACL, pp. 6578–6588 (2020)
16. Wan, H., Yang, Y., Du, J., Liu, Y., Qi, K., Pan, J.Z.: Target-aspect-sentiment joint detection for aspect-based sentiment analysis. In: AAAI, vol. 34, pp. 9122–9129 (2020)
17. Wang, K., Shen, W., Yang, Y., Quan, X., Wang, R.: Relational graph attention network for aspect-based sentiment analysis. arXiv preprint [arXiv:2004.12362](https://arxiv.org/abs/2004.12362) (2020)
18. Wang, W., Pan, S.J., Dahlmeier, D., Xiao, X.: Recursive neural conditional random fields for aspect-based sentiment analysis. arXiv preprint [arXiv:1603.06679](https://arxiv.org/abs/1603.06679) (2016)
19. Wang, Y., Huang, M., Zhu, X., Zhao, L.: Attention-based LSTM for aspect-level sentiment classification. In: EMNLP, pp. 606–615 (2016)

20. Xu, H., Liu, B., Shu, L., Yu, P.S.: Bert post-training for review reading comprehension and aspect-based sentiment analysis. arXiv preprint [arXiv:1904.02232](https://arxiv.org/abs/1904.02232) (2019)
21. Xue, W., Li, T.: Aspect based sentiment analysis with gated convolutional networks. arXiv preprint [arXiv:1805.07043](https://arxiv.org/abs/1805.07043) (2018)
22. Yang, M., Tu, W., Wang, J., Xu, F., Chen, X.: Attention based LSTM for target dependent sentiment classification. In: AAAI (2017)
23. Zhang, C., Li, Q., Song, D.: Aspect-based sentiment classification with aspect-specific graph convolutional networks. arXiv preprint [arXiv:1909.03477](https://arxiv.org/abs/1909.03477) (2019)
24. Zhang, M., Qian, T.: Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis. In: EMNLP, pp. 3540–3549 (2020)
25. Zhao, J., Wang, X., Shi, C., Hu, B., Song, G., Ye, Y.: Heterogeneous graph structure learning for graph neural networks. In: AAAI (2021)
26. Zhao, P., Hou, L., Wu, O.: Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification. Knowl. Based Syst. **193**, 105443 (2020)
27. Zheng, Y., Zhang, R., Mensah, S., Mao, Y.: Replicate, walk, and stop on syntax: an effective neural network model for aspect-level sentiment classification. In: AAAI, vol. 34, pp. 9685–9692 (2020)



Highway-Based Local Graph Convolution Network for Aspect Based Sentiment Analysis

Shiguan Pang, Zehao Yan, Weihao Huang, Bixia Tang, Anan Dai,
and Yun Xue^(✉)

School of Physics and Telecommunication Engineering, South China Normal University, Guangzhou, China

{psg-nlp, yzh_scnu, huangweihao, 2020022259, daianan, xueyun}@m.scnu.edu.cn

Abstract. Aspect-level sentiment analysis is a fine-grained task in sentiment analysis, whose target is to identify the sentiment polarity of a specific aspect in a sentence. Due to the complexity of the human language, the widely-applied syntactic-based neural network methods have deficiencies in precisely capturing the relation between aspects and opinion words, and thus results in the misunderstanding of the sentiment. To address such issue, we focus on optimizing the encoding of syntactic information. To start with, the sub-dependency trees, from the basic dependency tree, are constructed in line with the syntactic distance. Further, we propose a novel Highway-Based Local Graph Convolution Network (HL-GCN) to capture the more-related information and thus facilitate the sentiment classification. Substantial experiments on a variety of datasets are performed. Comparing to the state-of-arts, the proposed model shows the effectiveness in eliminating the noise from the dependency tree, which results in an even higher classification accuracy.

Keywords: Aspect-level sentiment analysis · Highway network · Graph convolution network · Local dependency tree

1 Introduction

Aspect-based sentiment analysis (ABSA) aims to precisely identify the polarity of a given aspect within a certain context, which leads to its widely-application in multiple fields due to its significance in understanding the sentiment of specific words. For instance, in the sentence ‘*It has bad memory but a good battery life.*’, the sentiment polarity is negative for the aspect ‘*memory*’ but positive for ‘*battery life*’. That is, the aspect-based sentiment analysis, is more and more recognized as a resolution because it paves a way for greater depth of analysis.

One of the key points of aspect-based sentiment analysis is to establish the connections between the aspect and opinion words [1–4]. Notably, recent research highlights the superiority of attention mechanisms in tackling such issue.

S. Pang and Z. Yan—Equal contribution.

Whereas, the use of attention mechanisms is still limited primarily because of the complexity of human languages. Current attention-based methods may fail to capture the relation between the aspect and the opinion words in certain cases. We take a comment '*The staff should be a bit more friendly*' as an example. The word '*friendly*' is commonly seen in relation to the word '*staff*' in consumer reviews. For this reason, more attention weight may be assigned to '*friendly*' rather than '*should be*', which can result in the opposite interpretation of the aspect sentiment. On the other hand, efforts are made to establish relation between aspect and opinion words based on syntactic information. Previous work of manually-defined syntactic principles largely depend on the hand-craft effectiveness. More recently, the establishing of dependency tree is capable of providing more comprehensive understanding of the syntax structure, based on which research of applying RNN to encode the syntactic information is still ongoing [5–8]. Specifically, RNN-based models have distinctive defects in encoding the syntax structure of the sentence.

Encouragingly, Graph Neural Network is one such approach, with recent research exploring the potential of integrating syntactic information into context and aspect for ABSA [9–13]. However, in terms of the complexity sentence structure, the aspect node and its sentiment node can be connected to each other through multiple generations, during which process irrelevant information can be introduced. Besides, the issue of consistent node representation, caused by the over smoothing of multi-layer architecture, is also pronounced [14].

There is a considerable gap between the accurately classifying aspect sentiment and the state-of-arts outcomes. In order to tackle those issues, we mainly focus on establishing the relation between the aspect and its most relevant words as well as eliminating the noise from unrelated words. To this end, we compute the syntactic distance of the aspect and its context from the dependency tree, based on which the basic dependency tree is divided reshape into sub-trees. Such a sub-tree not only concentrates more on the aspect and its opinions words with reduced distance, but also removes the noise from unrelated words. Then we devise a Highway-Based Local Graph Convolution Network (HL-GCN) to deal with the original dependency trees and proposed dependency sub-trees. Our model integrates the Highway Network into GCN for effective-information retaining, and thus reduces the noise within the modeling of syntactic tree. Finally, the highway mechanism is utilized to integrate the global and local information. On the task of sentiment classification, we compare our models against baseline methods on publicly available datasets to investigate its working performance.

To sum up, our contributions are three-folds:

1. On purpose to focus on the aspect and its related words, we propose the sub-dependency trees by pruning the original dependency tress in line with the syntactic distances.
2. The Highway Network is embedded within the layers of GCN, which reduces the over-smoothing impact of multi-layers. In such manner, the noise caused by unrelated words are removed.

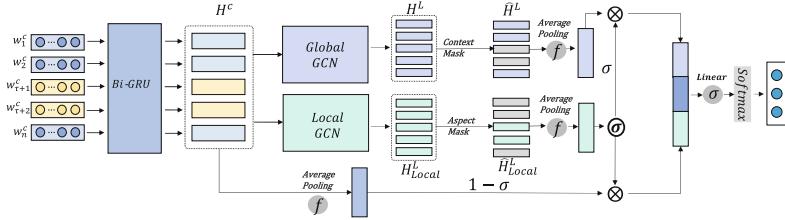


Fig. 1. Architecture of HL-GCN

3. Experiments correspond to extracting syntactic information and reducing noise are devised to validate the model effectiveness. Our method is a competitive alternative in aspect-level sentiment analysis based on the experimental results.

2 Related Work

Previous works combining recurrent neural networks (RNNs) and classical attention mechanism (i.e. co-attention mechanism, self-attention mechanism, hierarchical attention mechanism) have already achieve decent results in sentiment analysis [4, 15, 16]. In such works attention mechanisms have identified the significance in extracting the information from both the aspect and the context [17]. Besides, the pre-trained model Bert has made great success in some NLP tasks. For example, [18]utilized an additional corpus to fine-tune and showed its improvement in aspect extraction and ABSA.

Some other strategies try incorporate syntactic information into representation forming process. Dong et al. devise the Adaptive Recursive Neural Network (AdaRNN) that can adaptively propagate the sentiments of words towards the target based on context and syntactic structure [6]. Following this research, Nguyen and Shirai propose an extension model of RNN and AdaRNN to obtain a an aspect representation from ‘a target dependent binary phrase dependency tree’ [7]. By employing the proximity weight, Zhang et al. build a framework that leverages n-gram information and syntactic dependency between aspect and contextual terms into an applicable model [10].

More recently, advances in GCN provide more opportunities for aspect-based sentiment analysis. [9, 11] transformed syntactic dependency tree into graph structure and utilized GCN to extract syntactic information. [13] proposed Bidirectional-GCN to encode the directional information of syntactic dependency tree. However, those model ignore the complexity of the sentence structure when encoding the base syntactic dependency tree. Our work differs from these because it focuses on eliminating the noise from unrelated words during syntactic tree establishing. Furthermore, the increasing number of GCN layers leads to the over smoothing, and thus results in the similar representation of each node in the graph. To this end, a better working performance, on aspect-based sentiment analysis, is expected.

3 Methodology

Figure 1 presents the architecture of HL-GCN. The proposed model firstly maps the context and target words into word vectors. Then Bi-GRU is carried out to preserve sequential information on each word vector. In Global GCN module, the global syntactic information is extracted and integrated into the sentence representation. Nevertheless, the representation more concentrated on aspect is obtained in the local GCN module. Lastly, following the average pooling, the highway mechanism is utilized to fused the representation learned before and sent to the softmax layer for sentiment polarity prediction.

The components of HL-GCN are introduced separately in the rest of the section.

3.1 Bi-GRU Layer

The Bidirectional GRU (abbreviated as Bi-GRU) layer is established for sequence encoding, based on which the hidden state can be extracted. The hidden state representation of sentence c is conveyed as $H^c = \{h_1^c, h_2^c, \dots, h_{\tau+1}^c, \dots, h_{\tau+m}^c, h_{n-1}^c, h_n^c\}$, together with $h_t^c \in \mathbb{R}^{2d_h}$ referring to the hidden state of Bi-GRU at time step t and d_h to the dimensionality of a hidden state vector in an unidirectional GRU.

3.2 Global Graph Convolution Module

Highway Network. In the highway network, by using the gating units, some inputs are regulated through the network whilst others can flow across the layers unimpededly. Let T be the transform gate and C be the carry gate, to facilitate computing, we set $C = 1 - T$, thereby the highway network is expressed as:

$$y = H(x, W_H) \circ T(x, W_T) + x \circ (1 - T)(x, W_C) \quad (1)$$

where W_H , W_T and W_C are parametric matrices optimized via training. We integrated the Highway unit into the GCN to improve the working performance.

Highway Graph convolution on Dependency Trees. At this stage, we take the dependency tree, corresponding to the sentence, as a graph. Within this graph, each representation of aspect/ context refers to a node while the dependency between each two aspect/ context indicates the node connection. Accordingly, the GCN model is applied to capture the syntactic information between each node and its adjacent nodes. The information propagation during this process can be delivered as:

$$\tilde{h}_i^l = \sum_{j=1}^n A_{ij} W^l h_j^{l-1} \quad (2)$$

together with

$$h_i^l = \text{ReLU}\left(\frac{\tilde{h}_i^l}{(d_i + 1)} + b^l\right) \quad (3)$$

where $h_j^{l-1} \in \mathbb{R}^{2d_h}$ stands for j^{th} node representation evolved from $(l-1)^{th}$ layer while $h_i^l \in \mathbb{R}^{2d_h}$ is the output of l^{th} layer. We also have $d_i = \sum_{j=1}^n A_{ij}$ denoting the degree of i^{th} node, W^l as the weight and the b^l as bias.

On this occasion, the Highway Network [19] is introduced to tackle the GCN layer output. Specifically, the highway network is embedded between two adjacent layers whose main purpose is to preserve the node information and reduce over-fitting of GCN. For the node representation in l^{th} layer defined as $h^l = \{h_1^l, h_2^l, \dots, h_n^l\}$, we concatenate h^l and the hidden state H^c . The concatenating result is sent to the Highway Network, which is:

$$\text{Highway}_{in}^l = [h^l \oplus H^c] \quad (4)$$

and its output can be:

$$\text{Highway}_{out}^l = \text{Highway}(\text{Highway}_{in}^l) \quad (5)$$

$$\tilde{h}_i^{l+1} = \sum_{j=1}^n A_{ij} W^{l+1} \text{Highway}_{out}^l \quad (6)$$

We take $H^L = \{h_1^L, h_2^L, \dots, h_{\tau+1}^L, \dots, h_{\tau+m}^L, \dots, h_n^L\}$ as the outcome of $L - layer$ Global Graph Convolutional Module. In this way, the overfitting of GCN, caused by the increasing number of layers, can be effectively suppressed. That is, via the multiplying of GCN layers, more attention is assigned to more important representation, ignoring the longer distances. Moreover, more syntactic information is taken into consideration as well.

3.3 Local Graph Convolutional Module

Notably, aiming at concentrating on the semantic-related words to the aspect, our Local Dependency Graph Convolutional Module gives the first priority to the sentiment modifiers, and then other informative words (i.e. conjunctions, auxiliary verbs and etc.)

As pointed out in the Introduction, within a complex sentence structure, there can be a long distance between an aspect and its semantic-related words. For this reason, noise from unrelated words is generated while extracting the syntactic information. Thus, the sub-tree of the basic dependency tree, with the direct connection between the aspect and the related words, is established.

Obtaining Sub-dependency Trees. Specifically, the distance between two connected nodes can be defined as 1. By traversing the dependency tree, the syntactic distance of all remaining nodes to the current aspect is obtained. Figure 2(a) illustrates the distance to aspect ‘memory’ and ‘battery life’ in sentence ‘It has bad memory but a good battery life’. We observe that the distance of ‘memory’ to itself is zero and that of ‘has’, ‘bad’, ‘but’ and ‘life’ is 1, etc. The distance of ‘battery life’ is determined in the same manner. As pointed out in the Introduction, in classical attention mechanism, less attention weight will be given to some word like ‘but’ in

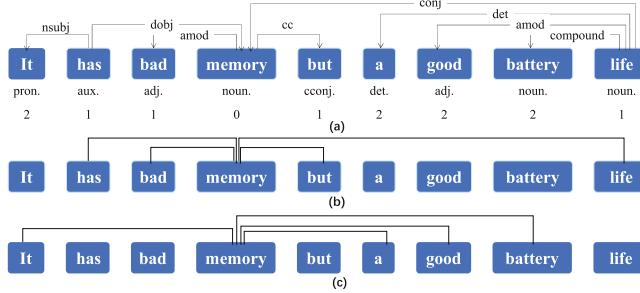


Fig. 2. (a) basic dependency tree with aspect ‘memory’;(b) ‘1-Local’ sub-dependency tree;(c) ‘2-Local’ sub-dependency tree

this sentence, which is in fact not the case. In comparison, our approach is more capable in incorporating the syntactic information.

Furthermore, all the nodes with distance 1 are kept to establish a local dependency tree. Figure 2(b) and (c) exhibit the local dependency tree of aspects ‘memory’ with distance 1 and 2 respectively.

Highway Graph convolution on Sub-Dependency Trees. Notwithstanding, there are still some words in the local dependency tree having little contribution to the aspect sentiment. As a result, the proposed Highway GCN is integrated to further remove the unrelated word representation and update the aspect. In such a way, we have:

$$\tilde{h}_{n-Local}^{l+1} = \sum_{j=1}^k A_{ij}^{n-Local} W_{n-Local}^{l+1} h_j^l \quad (7)$$

$$h_{n-Local}^{l+1} = \text{ReLU}\left(\frac{\tilde{h}_{n-Local}^{l+1}}{(d_\tau + 1)} + b_{n-Local}^l\right) \quad (8)$$

$$\text{Highway}_{in}^l = [h_{n-Local}^l \oplus H^c] \quad (9)$$

$$\text{Highway}_{out}^l = \text{Highway}(\text{Highway}_{in}^l) \quad (10)$$

The final:

$$\tilde{h}_{n-Local}^L = \sum_{j=1}^k A_{ij}^{n-Local} W_{Aspect}^L \text{Highway}_{out}^{L-1} \quad (11)$$

It is worth noticing that each node within each sub-graph fully connects to aspect. We thus adopt a 2-layer-Highway GCN instead of a large number.

We employ the attention mechanism to learn their corresponding importance a_1 , a_2 and a_3 as follow:

$$a_1, a_2, a_3 = \text{Attention}(H_{1-Local}^L, H_{2-Local}^L, H_{3-Local}^L) \quad (12)$$

where a_1 , a_2 and a_3 indicates the attention values with embedding $H_{1-Local}^L$, $H_{2-Local}^L$ and $H_{3-Local}^L$ respectively. The output of this layer is computed as follows:

$$H_{Local}^L = a_1 \circ H_{1-Local}^L + a_2 \circ H_{2-Local}^L + a_3 \circ H_{3-Local}^L \quad (13)$$

where $H_{1-Local}^L$, $H_{2-Local}^L$ and $H_{3-Local}^L$ are the representation via Highway GCN.

Accordingly, the sentence representation H_{Local}^L , as the output of this module is obtained with more concerning about the aspect.

3.4 Aspect-Specific Mask and Context-Specific Mask

The sentence representation from the Local Dependency Graph Convolutional Module is given as H_{Local}^L , which concerns more about the aspect. In contrast, H^L from the Global Graph Convolutional Module contains more information about the context. At this stage, we respectively mask out the information of the context from H^L and keep the aspect information within H_{Local}^L unchanged:

$$mask_{aspect} = \begin{cases} 0, & 1 \leq t < \tau + 1, \tau + m < t < n \\ 1, & \tau + 1 \leq t \leq \tau + m \end{cases} \quad (14)$$

$$mask_{context} = \begin{cases} 1, & 1 \leq t < \tau + 1, \tau + m < t < n \\ 0, & \tau + 1 \leq t \leq \tau + m \end{cases} \quad (15)$$

$$\hat{H}^L = H^L * mask_{context} \quad (16)$$

$$\hat{H}_{Local}^L = H^L * mask_{aspect} \quad (17)$$

The outputs of the masking layer, as the most prominent representation of the aspect and the context, are presented as $\hat{H}_{Local}^L = \{0, \dots, \hat{h}_{Local-(\tau+1)}^L, \dots, \hat{h}_{Local(\tau+m)}^L, \dots, 0\}$ and $\hat{H}^L = \{\hat{h}_1^L, \hat{h}_2^L, \dots, 0, \dots, 0, \dots, \hat{h}_n^L\}$.

3.5 Sentiment Classification

An average pool is performed on aspect, context and sentence to retain the information within the representations.

$$r_a = AveragePooling(\hat{H}_{Local}^L) \quad (18)$$

$$r_c = AveragePooling(\hat{H}^L) \quad (19)$$

$$r_s = AveragePooling(H^c) \quad (20)$$

We use the highway mechanism to obtain the final representation o , r_a , r_c and r_s are computed as:

$$o = r_s \otimes [1 - \sigma(r_a)] + r_c \otimes \sigma(r_a) \quad (21)$$

Compared with concatenation, the information is enabled to transmit across multiple channels and we can observe the information flow more clearly in this way. The final representation o is fed to a fully connected layer, followed by a softmax classifier to obtain the probability distribution over the different sentiment polarities.

$$p = \text{softmax}(W_p o + b_p) \quad (22)$$

where $p \in \mathbb{R}^{d_p}$ and d_p stands for the number of sentiment classes. We also have $W_p \in \mathbb{R}^{d_p \times 3d_h}$ and $b_p \in \mathbb{R}^{d_p}$ as the trainable parameter matrix and the learnt weight.

3.6 Model Training

The training of our model is performed with the loss being the cross entropy loss with L_2 regularization, as shown in Eq. (24), whose parameters and weights are updated via backpropagation.

$$L = -\sum_i \sum_{j=1}^P y_i^j \log \hat{y}_i^j + \lambda \|\theta\|^2 \quad (23)$$

where i is the index of i^{th} sample, j is the index of j^{th} sentiment class, P is the number of sentiment classes, y is the real distribution of sentiment and \hat{y} is the predicted one. Besides, λ_r is the weight of L_2 regularization.

4 Experiment

4.1 Experimental Setting

Our experiments are carried out on three public datasets, i.e. Twitter reported in [6] as well as lap14 and rest14 from SemEval 2014 Task4 [20]. The detail of each dataset is presented in Table 1. All the samples in the experiment are labeled with three polarities: positive, neutral and negative. The dependency tree is obtained from <https://spacy.io/api/dependencyparser>. The word embeddings in all datasets are initialized using 300-dimensional word vectors pretrained by Glove [21]. To prevent overfitting, the hidden states of Bi-GRU are set to 300 with the learning rate of 0.001. Besides, the Adam optimizer is adopted [22]. The L_2 -regularization weight is set as 0.0001.

In this experiment, we adopt accuracy and Macro-F1 as evaluation metrics to denote the working performance. The reported outcomes are obtained as the average value over 3 runs with random initializations ¹.

4.2 Results

We take the model with the best performance in the past two years as the baseline. The classification results of our model compared to the baseline methods

¹ Data and code can be found at <https://github.com/pangsg/HLGCN>.

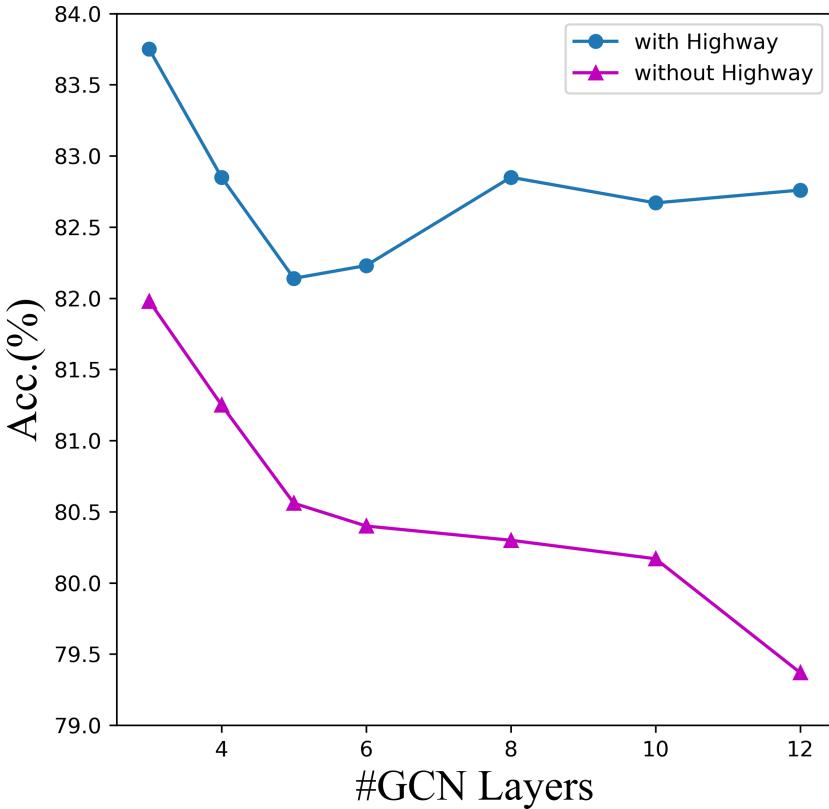


Fig. 3. Number of GCN layers in rest14 test set

are shown in Table 2, from which several phenomena can be observed. Firstly, among all these models, the accuracy and F1-score of HL-GCN outperforms most of baseline methods. Secondly, compared to AS-GCN, CDT, and Bi-GCN, the performance of GCN has been improved prominently when highway network integrated into GCN and utilizing the sub-dependency tree as input. It also outperforms R-GAT which encodes the syntactic information in a different way. This demonstrates the syntactic information can be better encoded by our HL-GCN. Finally, as the powerful pre-trained model, the basic Bert can improve the existing ABSA models (R-GAT+Bert, DGEDT+Bert), after incorporating our HL-GCN(HL-GCN+Bert), a new state of the art has been achieved. These results have proved that our HL-GCN performs effectively in capturing significant syntactic information in ABSA.

4.3 Number of GCN Layers

We propose an investigation on the number of GCN layers. to verify the effectiveness of Highway Network in our model. We set 3, 4, 5, 6, 8, 10 and 12 for numbers of GCN layer on rest14 dataset and report the average result.

According to Fig. 3, there is a considerable performance gap between employing and removing the Highway Network. The application of Highway Network does bring an increment to the working performance. On the other hand, the classification accuracy fluctuates within a small range of $\pm 0.5\%$ in spite of the increasing number of GCN layers. In contrast to the model without Highway network, the implementation of Highway network explicitly prevents the impact of overfitting caused by GCN layers.

4.4 Ablation Study

In order to determine the significance of the different components, an ablation study is carried out.

Table 1. Statistics of datasets.

Dataset	Positive		Negative		Neutral	
	Train	Test	Train	Test	Train	test
Rest14	2164	728	807	196	637	196
Laptop14	994	341	870	128	464	169
Twitter	1561	173	1560	173	3127	346

Table 2. Sentiment classification results (including ablation Study)

Category	Method	Twitter		Lap14		Rest14	
		Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
GLoVe models	AS-GCN	72.15	70.40	75.55	71.05	80.96	72.21
	CDT	74.66	73.66	77.19	72.99	82.30	74.02
	BiGCN	74.16	73.35	74.59	71.84	81.97	73.48
	R-GAT	75.57	73.82	77.42	73.76	83.30	76.08
	DGEDT	74.8	73.82	76.8	72.3	83.90	75.1
Ours	HL-GCN	76.01	74.77	78.05	74.14	83.75	76.30
	HL-GCN w/o highway	74.84	73.00	75.92	72.00	81.95	74.74
	HL-GCN w/o Local GCN	73.68	71.10	75.39	70.69	80.98	70.83
	HL-GCN w/o Global GCN	72.95	71.53	75.39	70.79	80.26	70.66
Bert models	R-GAT+Bert	76.15	74.88	78.21	74.07	86.60	81.35
	DGEDT+Bert	77.9	75.4	79.5	75.6	86.3	80.00
Ours	HL-GCN+Bert	78.01	77.28	80.22	77.28	87.66	82.79

* The symbol ‘-’ indicates this result is not available in their work. The results of baseline are taken from the original papers.

* ‘w/o highway’ : removing Highway Networks between GCN layers;

‘w/o local GCN’ : removing Local Graph Convolution Module;‘w/o Global GCN’: removing Global

Graph Convolution Module.

First, removal of Highway Networks between GCN layers leads to accuracy drop of 1.17%, 2.13% and 1.8% on Twitter, lap14 and rest14, respectively. Since the gate mechanism in Highway Network effectively filters the noise, the integration of highway units into GCN layers can facilitate the retaining of the meaningful information. Second, the withdrawal of Local GCN results in an even larger decrease in working performance. Without exploiting the closer contact word dependencies on the dependency trees, the model becomes less competitive in capturing the relation among aspects and opinion words. In this paper, the maximal distance of sub dependency trees is ‘3-Local’ according to extra test. As for ‘w/o Global GCN’ model, the accuracy on three datasets is approximate to the ‘w/o local GCN’ model. We conclude that the global syntactic information and the local syntactic information are complementary in sentences of datasets.

4.5 Visualized Analysis

In this section, we employ the method ‘Mask Experiment’ proposed by CDT [23] to estimate the sentiment contribution of word w in the sentence a . The formula is as follows:

$$\gamma(w, s) = \frac{1}{m} \sum_{i=1}^d |\hat{h}_a^i - \hat{h}_{(a//w)}^i| \quad (24)$$

where d is the dimension of the final representation \hat{h}_a learned by our model, w is the masked word, which indicates w becomes zero vector, $\hat{h}_{(a//w)}$ is the final representation of sentence a generated by our HL-GCN with the word w masked. If $\hat{h}_a^i = \hat{h}_{(a//w)}^i$, which demonstrates the word w has no sentiment contribution to sentence a .

CDT	0.34	0.35	1	0.16	0.039	0.26	0.15	0.019	0.014	0.0056	0.056	0.01	0.029	0.06	0.012	0.074	0.016	0.077	0.0056	0.09	0.0047	0.0074
HLGCN	1	0.37	0.3	0.26	0.33	0.37	0.3	0.2	0.38	0.33	0.25	0.4	0.24	0.22	0.2	0.29	0.39	0.13	0.33	0.19	0.11	0.096

CDT	0.00017	0.053	0.01	0.019	0.0082	0.011	0.024	0.033	0.0019	0.071	0.38	0.032	0.043	0.078	0.0209	1	0.016	0.004	0.11	0.3	0.066
HLGCN	0.3	0.46	0.51	0.51	0.48	0.55	0.64	0.6	0.65	1	0.87	0.46	0.3	0.63	0.39	0.52	0.32	0.48	0.49	0.3	0.22

Fig. 4. Word relevance scores for CDT and HL-GCN with the aspect word ‘waiter’ (Fig. 4(a)) and ‘soft shell crab’ (Fig. 4(b)), respectively. CDT predicts neutral and positive. HL-GCN predicts negative and neutral (ground truth is negative and neutral)

Figure 4 is the visualization of the attention placed on words, from which can be observed that our model assigns the attention score more legitimately than CDT in some long sentences cases. Figure 4(a) shows that CDT has assigned the highest score on ‘to’ mistakenly. However, our proposed HL-GCN is able to reduce the attention on irrelevant word ‘to’ and assign more attention on ‘complained’ correctly. The situation in Fig. 4(b) is the same as in Fig. 4(a), CDT regards ‘well’ as the opinion word for aspect word ‘soft shell crab’. Inversely,

HL-GCN performs better. Implying that our HL-GCN extracts local and global information reasonably, which verifies the validity of our strategy mentioned before.

5 Conclusion

In this paper, a highway-based local graph convolution network is proposed for aspect-based sentiment analysis task. In line with the working principle of GCN, the Highway Network and the local dependency tree are integrated into the model. By exploiting the syntactic structure and the word dependencies, the relation between aspects and their contexts are precisely captured. As such, the syntactic and semantic information is preserved and conveyed by the representations for sentiment classification. Experimental results show that the proposed model stably outperforms the current baselines and achieves the new state-of-the-art results on various datasets.

References

1. Jakob, N., Gurevych, I.: Extracting opinion targets in a single and cross-domain setting with conditional random fields. In: EMNLP, pp. 1035–1045 (2010)
2. Wang, Y., Huang, M., Zhu, X., Zhao, L.: Attention-based LSTM for aspect-level sentiment classification. In: EMNLP, pp. 606–615 (2016)
3. Chen, P., Sun, Z., Bing, L., Yang, W.: Recurrent attention network on memory for aspect sentiment analysis. In: EMNLP, pp. 452–461 (2017)
4. Li, H., Xue, Y., Zhao, H., Hu, X., Peng, S.: Co-attention networks for aspect-level sentiment analysis. In: Tang, J., Kan, M.-Y., Zhao, D., Li, S., Zan, H. (eds.) NLPCC 2019. LNCS (LNAI), vol. 11839, pp. 200–209. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32236-6_17
5. Lakkaraju, H., Socher, R., Manning, C.: Aspect specific sentiment analysis using hierarchical deep learning. In: NIPS Workshop on deep learning and representation learning, pp. 1–9 (2014)
6. Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., Xu, K.: Adaptive recursive neural network for target-dependent twitter sentiment classification. In: ACL, pp. 49–54 (2014)
7. Nguyen, T.H., Shirai, K.: PhraseRNN: phrase recursive neural network for aspect-based sentiment analysis. In: EMNLP, pp. 2509–2514 (2015)
8. Wang, W., Pan, S.J., Dahlmeier, D., Xiao, X.: Recursive neural conditional random fields for aspect-based sentiment analysis. In: EMNLP, pp. 616–626 (2016)
9. Zhang, C., Li, Q., Song, D.: Aspect-based sentiment classification with aspect-specific graph convolutional networks. arXiv preprint [arXiv:1909.03477](https://arxiv.org/abs/1909.03477) (2019)
10. Zhang, C., Li, Q., Song, D.: Syntax-aware aspect-level sentiment classification with proximity-weighted convolution network. In: ACM SIGIR, pp. 1145–1148 (2019)
11. Sun, K., Zhang, R., Mensah, S., Mao, Y., Liu, X.: Aspect-level sentiment analysis via convolution over dependency tree. In: (EMNLP-IJCNLP), pp. 5683–5692 (2019)
12. Wang, K., Shen, W., Yang, Y., Quan, X., Wang, R.: Relational graph attention network for aspect-based sentiment analysis. In: ACL-IJCNLP (2020)

13. Tang, H., Ji, D., Li, C., Zhou, Q.: Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification. In: ACL, pp. 6578–6588 (2020)
14. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint [arXiv:1609.02907](https://arxiv.org/abs/1609.02907) (2016)
15. Wang, S., Mazumder, S., Liu, B., Zhou, M., Chang, Y.: Target-sensitive memory networks for aspect sentiment classification. In: ACL, pp. 957–967 (2018)
16. Li, L., Liu, Y., Zhou, A.: Hierarchical attention based position-aware network for aspect-level sentiment analysis. In: CoNLL, pp. 181–189 (2018)
17. Zeng, J., Ma, X., Zhou, K.: Enhancing attention-based LSTM with position context for aspect-level sentiment classification. IEEE Access **7**, 20462–20471 (2019)
18. Xu, H., Liu, B., Shu, L., Yu, P.S.: Bert post-training for review reading comprehension and aspect-based sentiment analysis. arXiv preprint [arXiv:1904.02232](https://arxiv.org/abs/1904.02232) (2019)
19. Srivastava, R.K., Greff, K., Schmidhuber, J.: Highway networks. arXiv preprint [arXiv:1505.00387](https://arxiv.org/abs/1505.00387) (2015)
20. Pontiki, M., Papageorgiou, H., Galanis, D., Androutsopoulos, I., Pavlopoulos, J., Manandhar, S.: SemEval-2014 task 4: Aspect based sentiment analysis. In: SemEval, vol. 2014, p. 27 (2014)
21. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: EMNLP, pp. 1532–1543 (2014)
22. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
23. Sun, K., Zhang, R., Mensah, S., Mao, Y., Liu, X.: Aspect-level sentiment analysis via convolution over dependency tree. In: EMNLP-IJCNLP, pp. 5683–5692 (2019)



Dual Adversarial Network Based on BERT for Cross-domain Sentiment Classification

Shaokang Zhang^{1,2}, Xu Bai^{1,2(✉)}, Lei Jiang^{1,2}, and Huailiang Peng^{1,2}

¹ Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{zhangshaokang, baixu, jianglei, penghuailiang}@iie.ac.cn

² School of Cyber Security, University of Chinese Academy of Sciences,
Beijing, China

Abstract. Cross-domain sentiment classification uses useful information of the source domain to promote the classification accuracy of the target domain. Although previous approaches consider the effects of aspect information of the sentences, they lack the mechanism of syntactic constraints which may mistakenly assign irrelevant words to aspects. In this paper, we propose Dual Adversarial Network based on BERT (DAN-BERT), which can better transfer sentiment across domains by jointly learning the representation of sentences and aspect-based syntax. Specifically, DAN-BERT extracts the common features at the sentence level and aspect-based syntax level by adversarial training. We learn the features of aspect-based syntax by building Graph Convolutional Network over the dependency tree of a sentence. Experiments on the four datasets show that Dual Adversarial Network based on BERT outperforms state-of-the-art methods.

Keywords: Sentiment classification · Dual adversarial network · Domain adaptation · Graph convolution network

1 Introduction

Sentiment classification is an important task in natural language processing and the purpose is to identify the sentiment polarity (e.g., positive or negative) of a sentence. General sentiment classification methods can achieve outstanding effect under a large amount of labeled training data [16]. However, we usually face some obstacles in actual situations: many scenes (e.g., politic and violence) lack sufficient labeled training data and labeling data is expensive and time-consuming. Cross-domain sentiment classification is proposed to solve this problem. It transfers the knowledge of the source domain to the target domain.

The main challenge of cross-domain sentiment classification is domain discrepancy because different ways of expressing emotions across domains. Typically, we obtain the domain-invariant features by adversarial learning methods [3]. It utilizes the domain classifier to minimize the difference of the source

domain and target domain by the gradient reversal layer (GRL). Recently, pre-trained language models show that they can effectively improve performance in many language tasks. BERT is the language model with multi-layer transformer and is trained by mask language model and next sentence prediction tasks [2]. Although BERT has achieved good performance, some problems are still not resolved when fine-tuning BERT in cross-domain sentiment classification. Firstly, the data of target domain is unlabeled which brings many difficulties to the fine-tuning stage. If we fine-tune BERT only by the labeled source domain data, the knowledge shift between source and target domains will reduce the performance of BERT. Secondly, the attention-based methods cannot sufficiently capture the syntactical dependencies between aspects and context words. Zhang et al. [20] uses an interactive attention transfer network to model aspects which express the sentiment directly. However, the attention mechanism may mistakenly assign unrelated context words to aspects. For example, “a good book for beginners but mediocre for more advanced readers”. Attention mechanism often uses “mediocre” to describe the aspect “beginners”, which is incorrect in the current context.

In this paper, we propose Dual Adversarial Network based on BERT (DAN-BERT) for Cross-domain Sentiment Classification. DAN-BERT model separately extracts domain-invariant features at the sentence level and aspect-based syntax level by adversarial training. Firstly, we further pre-train the BERT model by in-domain data to make the data distribution more inclined to specific task. In terms of the sentence level, we utilize BERT to learn the source and target domain features and obtain domain-invariant features by DANN. In terms of the aspect-based syntax level, BERT output is followed by Graph Convolutional Network [7] which captures the potential syntactic structure of the sentence by referring to syntactical dependency trees. Then, we only keep aspect features by a masking mechanism that covers up non-aspect words. We obtain the aspect-based syntax features on source and target domains by attention mechanism. Similarly, we get domain-invariant features through DANN. Because the target domain data is unlabeled, we select data which is similar to the distribution of target domain from source domain as candidates. Finally, we connect sentence and aspect-based syntax features and train the classifier under candidates. In summary, the main contributions of our work are summarized as follows:

- We propose the dual adversarial network based on BERT which associates with aspect-based syntax. It uses the dual adversarial mechanism to obtain the sentence and aspect-based syntax features for the sentiment classification task.
- We built the comparative experiments on the four datasets. The experimental results demonstrate that our method outperforms other state-of-the-art methods.

2 Related Work

The related works can be classified into two categories: domain adaptation, pre-trained language model.

Domain Adaptation: There are many methods to solve the domain adaptation problem. Domain adaptation such as cross-domain sentiment classification has attracted more and more research attention in natural language processing over the past decades. Among them, The Structural Correspondence Learning (SCL) [1] is proposed to produce shared features between source and target domains. Using domain-independent words as a bridge, the Spectral Feature Alignment (SFA) [11] solves the feature mismatch problem by aligning domain-specific words. Sharma et al. [14] extract the transferable information by the χ^2 test and cosine-similarity of the word vectors.

In recent years, many methods based on neural networks can extract better features. Yu et al. [19] used two auxiliary tasks to induce the sentence embeddings by convolutional neural network. The Domain-Adversarial training of Neural Networks (DANN) [3] leverages the adversarial learning method to extract common features of the source domain and target domain. Hierarchical Attention Transfer Network (HATN) [9] transfers the word-level and sentence-level attentions across domains. Interactive Attention Transfer Network (ITAN) [20] combines the information of sentences and aspects by interactive learning. However, these models do not consider the aspect-based syntax. We use Graph Convolutional Network to obtain the potential structure of the sentence by introducing syntactical dependency trees. We obtain the aspect-based syntax features through attention mechanism.

Pre-trained Language Model: Pre-trained language model has achieved significant improvements on multiple NLP tasks such as text classification [4], reading comprehension [13], machine translation [18]. Previous methods can be divided into two categories: feature-based methods and fine-tuning methods. The first focuses on learning contextualized word representations such as ELMo [12], which are applied to downstream tasks. Fine-tuning methods mainly further pre-train the language model on unlabeled corpus and fine-tune model with the labeled data. Universal language model fine-tuning (ULMFiT) [6] uses different learning rates and gradual unfreezing. BERT [2] consists of multi-layer transformer and learns bidirectional representations. Besides, many studies show that the improvement is obvious by further pre-training BERT because of decreasing domain discrepancy [4,15]. Because target domain data is unlabeled, it is unrealistic to fine-tune model. To address this problem, we automatically select the training data from the labeled source domain data.

3 Model

In this section, we first present the problem definition, followed by future pre-training and the details of dual adversarial network based on BERT. Finally, the adversarial training process is introduced.

3.1 Problem Definition

We assume that D_s and D_t are the source and target domains. In source domain, $\mathbf{X}_s^l = \{x_s^i, y_s^i\}_{i=1}^{N_s^l}$ are labeled source domain data, where N_s^l is the number of

labeled data. Besides, there are also some unlabeled data $\mathbf{X}_s^u = \{x_s^i\}_{i=1+N_s^l}^{N_s^l+N_s^u}$ in source domain, where N_s^u is the number of unlabeled data. In target domain, $\mathbf{X}_t = \{x_t^j\}_{j=1}^{N_t}$ are unlabeled data, where N_t is the number of unlabeled data. Cross-domain sentiment classification aims to train a classifier on the source data to predict the label of target domain data. The overall framework of the model is shown in Fig. 1.

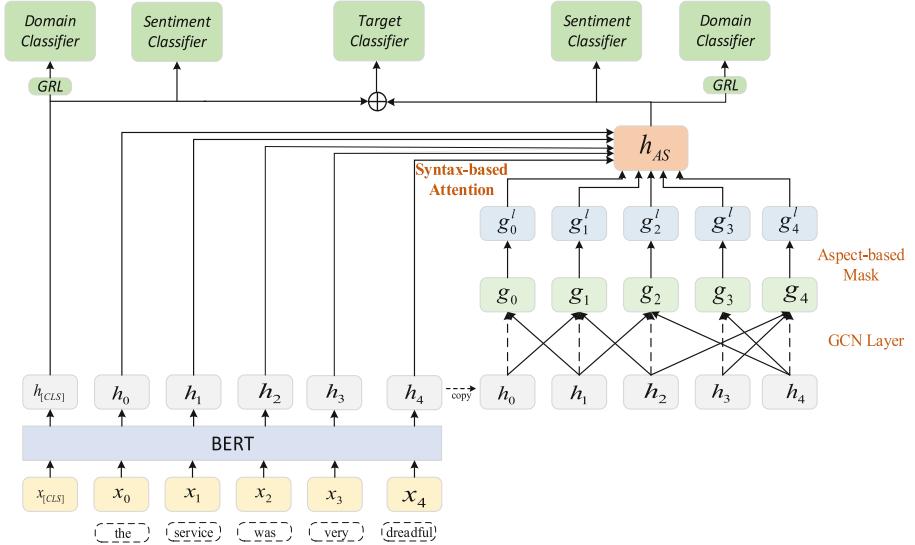


Fig. 1. The architecture of the DAN-BERT. The dotted line of GCN Layer denotes self-looping.

3.2 Further Pre-training

Although BERT achieves great success, there are still challenges in domain adaptation. BERT is pre-trained by Wikipedia which is not related to the sentiment classification task. Especially in the cross-domain scenario, the domain discrepancy will degrade the performance of BERT. Therefore, we continue pre-training the BERT on in-domain data. We focus on four datasets: Amazon, IMDB, Yelp and Airline. Because the Amazon dataset includes four fields: Books, Dvd, Electronics and Kitchen, we pre-train BERT on these data. On the other hand, we further pre-train BERT with the rest of the datasets.

3.3 Dual Adversarial Network Based on BERT

The further pre-trained BERT is suitable for specific task, but the aspect-based syntax features are not joined. In this study, we first obtain sentence features by BERT. Then, we utilize the multi-layer graph convolution networks over the syntactical dependency tree of a sentence to get syntactic features. Using

the syntactic features as a bridge, we obtain aspect-based syntax features by a attention mechanism which assigns weights to aspects. Moreover, we propose a data selection method that better fine-tunes BERT to adapt target domain. Finally, the dual adversarial training is introduced.

Extracting Sentence Features: BERT model consists of 12 transformer layers and the input sequence length does not exceed 512. The first token of sequence is the classification embeddings $[CLS]$. Given a sentence $x = [x_{[CLS]}, x_0, x_1, \dots, x_k]$, where k is the sequence length. The sentence features are obtained through BERT encoder:

$$h = BERT(x) \quad (1)$$

where $h = [h_{[CLS]}, h_0, h_1, \dots, h_k]$. We choose $[CLS]$ of last layer as final sentence features $h_{[CLS]}$.

Extracting Aspect-Based Syntax Features: Because the structure of syntactic dependency tree is graph, we leverage the GCN to encode it. The dependency tree of a sentence is built by the spaCy toolkit¹. GCN codes and updates the representations of nodes in the graph by immediate nodes. For a sentence, the number of nodes is k and the adjacency matrix $A \in \mathbb{R}^{k*k}$ is obtained on a dependency tree. Following the idea of self-looping [7], each word connects itself so that the diagonal of A is 1. The representation of each node is updated with GCN as below:

$$g_i^l = \text{ReLU}\left(\left(\sum_{j=1}^k A_{ij} W^l g_j^{l-1}\right)/(D_i + 1) + b^l\right) \quad (2)$$

where $g_j^{l-1} \in \mathbb{R}^d$ is the j -th token's representation in $l-1$ layer. Degree matrix $D_i = \sum_{j=1}^k A_{ij}$ is the j -th token in the tree. W^l and b^l are weights and biases.

We use two-layers GCN to learn the syntax features $g = [g_0, g_1, \dots, g_k]$. The last layer embedding of tokens $\bar{h} = [h_0, h_1, \dots, h_k]$ are fed into GCN.

We extract all aspects of sentence $AspectData$ by applying existing work [8]. We mask out words that are not belong to $AspectData$. The mask mechanism is as follows:

$$M_i = \begin{cases} M_i = 1, x_i \in AspectData \\ M_i = 0, x_i \notin AspectData \end{cases} \quad (3)$$

where M is mask vector. The masked syntax features $g^l = [g_0^l, g_1^l, \dots, g_k^l] = gM$ can capture long-distance words relations at aspect level. Each aspect expresses different semantics under syntax analysis. We construct an attention mechanism based on syntax features. The aspect-based syntax features are computed as below:

¹ <https://spacy.io/>.

$$\alpha_t = \frac{\exp(h_t(g_t^l)^T)}{\sum_{i=1}^k \exp(h_i(g_i^l)^T)} \quad (4)$$

$$h_{AS} = \sum_{i=1}^k \alpha_i h_i \quad (5)$$

Dual Adversarial Training: We separately obtain the domain-invariant features at the sentence level and aspect-based syntax level by adversarial training.

Domain Classifier: The goal of domain classifier is to predict the domain labels which come from the source or target domains. We use the gradient reversal layer (GRL) [3] to learn the common features which is difficult to distinguish by the domain classifier. The forward propagation and backpropagation process are as follows:

$$F(x) = x, \frac{\partial F(x)}{\partial x} = -\lambda I \quad (6)$$

where λ is the trade-off parameter. Before feeding to the domain classifier, the sentence and aspect-based syntax features separately go through the GRL as $F(h_{[CLS]}) = \hat{h}_{[CLS]}$, $F(h_{AS}) = \hat{h}_{AS}$. Then we feed it to the corresponding domain classifiers:

$$y_{[CLS]}^d = \text{softmax}(W_{[CLS]}^d \hat{h}_{[CLS]} + b_{[CLS]}^d) \quad (7)$$

$$y_{AS}^d = \text{softmax}(W_{AS}^d \hat{h}_{AS} + b_{AS}^d) \quad (8)$$

The cross-entropy loss is L_{dc} for data of the source and target domains:

$$L_{dc} = -\frac{1}{N_s^l + N_t} \sum_{i=1}^{N_s^l + N_t} (L(y_{[CLS]}^d, d_i) + L(y_{AS}^d, d_i)) \quad (9)$$

where d_i is the true domain label (0 indicates the source domain and 1 indicates the target domain). The parameters of domain classifier are not shared.

Sentiment Classifier: The sentence and aspect-based syntax features are sent to the sentiment classifier. The formula is as follows:

$$y_{[CLS]}^s = \text{softmax}(W_{[CLS]}^s h_{[CLS]} + b_{[CLS]}^s) \quad (10)$$

$$y_{AS}^s = \text{softmax}(W_{AS}^s h_{AS} + b_{AS}^s) \quad (11)$$

The classifier is trained on the labeled data from source domain:

$$L_{sc} = \frac{1}{N_s^l} \sum_{i=1}^{N_s^l} (L(y_{[CLS]}^s, y_s^i) + L(y_{AS}^s, y_s^i)) \quad (12)$$

where y_s^i is the ground truth of source domain. The parameters of sentiment classifier are not shared (Fig. 2).

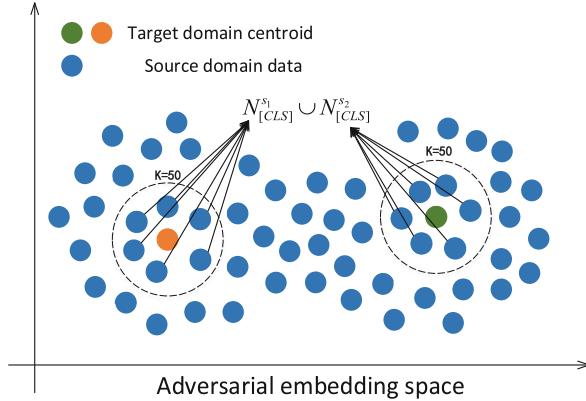


Fig. 2. The details of data selection. We map source and target domain data to a shared space by adversarial training. For each centroid, we select k nearest neighbors from source domain.

Data Selection: until now, we obtain the domain-invariant features and the data distribution of the source and target domains is similar. Intuitively, we can select training data from source domain. We propose a data selection method that automatically selects training data. We first obtain the sentence features on source and target domains $h_{[CLS]}^s, h_{[CLS]}^t$. Since the target domain data are unlabeled, we cluster the features of target domain [10] and get two centroids $c_{[CLS]}^{t_1}, c_{[CLS]}^{t_2}$. We select k candidates of each centroid from source domain. The candidates $N_{[CLS]}^{s_1}, N_{[CLS]}^{s_2}$ are selected via nearest neighbors selection (k NN) as shown in Fig. 3. The candidates are $N_{[CLS]} = N_{[CLS]}^{s_1} \cup N_{[CLS]}^{s_2}$ at sentence level. Similarly, we get candidates $N_{[AS]} = N_{[AS]}^{s_1} \cup N_{[AS]}^{s_2}$ at aspect-based syntax level. The final training data are $N_{ds} = N_{[CLS]} \cap N_{[AS]}$. We combine sentence and aspect-based syntax features for target classification:

$$y_t = \text{softmax}(W_t[h_{[CLS]} \oplus h_{[AS]}] + b_t) \quad (13)$$

The target classifier is trained on N_{ds} as follows:

$$L_{ds} = \frac{1}{N_{ds}} \sum_{i=1}^{N_{ds}} L(y_t, y_s^i) \quad (14)$$

Training Strategy: The training process is divided into two stages. We first get the domain-invariant features of sentence and aspect-based syntax using the following loss:

$$L = L_{dc} + L_{sc} \quad (15)$$

The best model is saved on the validation set. Then we use data selection method to obtain training data through the saved model. The loss of target classifier is as follows:

$$L_t = L_{ds} + \rho L_{reg} \quad (16)$$

where L_{reg} is the regularization and prevents the overfitting, ρ is the hyperparameter.

4 Experiment

4.1 Dataset

We utilize Amazon reviews dataset [9] to evaluate the effectiveness of our method. It includes four domains: Books (B), Dvd (D), Electronics (E) and Kitchen (K). We built 12 cross-domain sentiment classification tasks. For example, the $A \rightarrow B$ is the task which transfers from the source domain A to the target domain B. There are a lot of unlabeled data which are used to further pre-train the BERT. Moreover, we also select the three datasets IMDB (I), Yelp (Y) [17] and Airline (A) datasets ² with the different origins. The BERT also continues to be pre-trained on the unlabeled data and we construct 6 cross-domain sentiment classification tasks. In order to make the label space consistent, we uniformly changed the label to binary. Table 1 summarizes the all datasets.

Table 1. Statistics of the experimental datasets

Domain	Books	Dvd	Electronics	Kitchen	IMDB	Yelp	Airline
Train	5600	5600	5600	5600	5600	5600	5600
Test	400	400	400	400	400	400	400
Unlabel	9750	11843	17009	13856	78919	72966	35396

4.2 Implementation Details

We utilize the WordPiece to split the sentences to tokens. In our case, we use the BERT_{base} (uncased) to extract the sentence embeddings. We separately continue pre-training the BERT on Amazon and other datasets. During the pre-train, the maximum sequence length, batch size, learning rate and step is 128, 32, 2e-5 and 100000 respectively. During the dual adversarial training, the maximum sequence length, batch size, learning rate and dropout is 128, 20, 2e-5 and 0.1 respectively. The adaptation rate is $\lambda = \frac{2}{1+exp(-10q)-1}$, where $q = \frac{e}{E}$. The e and E are current epoch and the maximum epoch, respectively.

4.3 Baselines

Naive: it is a non-domain-adaptive method based on LSTM [5].

DANN: it is based on the adversarial training. DANN performs domain adaptation with the representation encoded in a 5000-dimension feature vector [3].

² <https://github.com/quankiquanki/skytrax-reviews-dataset>.

BERT: it fine-tunes vanilla BERT by labeled source domain data.

FPT-BERT: the further pre-training BERT model on unlabeled data. It is trained on labeled source domain data.

IATN-BERT: IATN [20] model based on BERT.

DAN-BERT: it is the proposed models.

We compare our method with other state-of-the-art methods on four datasets and the experimental results are shown in Table 2. The mean and standard error of the accuracy are calculated over 5 runs with different random seeds. It is obvious that DAN-BERT method achieves the best performances on most tasks. Since the Naive only use the source domain samples, the classification accuracy is low at every task. The vanilla BERT only uses the source domain data and the classification accuracy is 85.8% on average. It proves that BERT model generates high-quality sentence embeddings. Compared with BERT, FPT-BERT exceeds 3% on average. BERT is continuously pre-trained on in-domain data and the data distribution is closer to cross-domain sentiment classification task. The performance is significantly improved due to reduce domain discrepancy. For

Table 2. Classification accuracy (%) on the Amazon, IMDB, Yelp and Airline datasets.

S T	Naive	DANN	BERT	IATN-BERT	FPT-BERT	DAN-BERT
B D	80.1 ± 0.2	81.2 ± 0.4	88.6 ± 0.3	88.8 ± 0.5	90.8 ± 0.5	91.4 ± 0.4
B E	72.2 ± 0.4	76.5 ± 0.7	89.4 ± 1.3	89.9 ± 1.4	93.8 ± 0.3	94.1 ± 0.2
B K	74.6 ± 0.3	80.3 ± 0.2	90.5 ± 0.4	91.4 ± 0.6	94.0 ± 0.5	94.8 ± 0.3
D B	80.1 ± 0.5	81.6 ± 0.6	90.9 ± 0.5	90.8 ± 0.6	92.2 ± 0.4	93.0 ± 0.3
D E	72.3 ± 0.3	76.9 ± 0.4	88.5 ± 1.1	89.3 ± 0.2	93.0 ± 0.2	93.5 ± 0.4
D K	75.8 ± 0.6	77.6 ± 0.6	90.9 ± 0.2	91.7 ± 0.6	93.1 ± 0.5	94.0 ± 0.4
E B	70.9 ± 0.3	77.7 ± 0.2	88.7 ± 0.2	88.6 ± 0.7	91.4 ± 0.3	91.8 ± 0.2
E D	74.7 ± 0.4	75.5 ± 0.3	86.4 ± 0.6	86.8 ± 0.6	90.7 ± 0.7	91.0 ± 0.3
E K	80.9 ± 0.6	85.0 ± 0.6	92.8 ± 0.6	93.3 ± 0.2	95.0 ± 0.2	94.8 ± 0.3
K B	72.5 ± 0.3	79.0 ± 0.5	89.2 ± 0.3	89.2 ± 0.6	91.7 ± 0.4	92.5 ± 0.3
K D	73.6 ± 0.2	78.3 ± 0.4	87.9 ± 0.4	87.6 ± 1.0	90.7 ± 0.5	90.9 ± 0.1
K E	79.8 ± 0.4	84.6 ± 0.2	92.5 ± 0.2	92.3 ± 0.7	94.0 ± 0.4	94.7 ± 0.3
I Y	68.2 ± 0.4	72.4 ± 2.1	80.8 ± 1.7	83.1 ± 0.3	82.4 ± 0.6	87.1 ± 0.5
I A	69.5 ± 0.8	74.6 ± 1.8	80.1 ± 2.1	80.4 ± 0.7	83.2 ± 1.2	85.2 ± 0.4
Y I	65.3 ± 1.1	69.5 ± 1.5	72.1 ± 1.3	74.6 ± 0.6	79.9 ± 0.9	80.3 ± 0.7
Y A	70.2 ± 0.9	74.2 ± 2.2	83.8 ± 1.6	84.8 ± 0.4	86.2 ± 0.8	87.3 ± 0.5
A I	60.9 ± 0.6	63.7 ± 1.9	70.6 ± 1.6	70.3 ± 0.6	70.1 ± 1.3	77.4 ± 0.4
A Y	71.1 ± 0.7	73.9 ± 1.3	81.2 ± 1.1	80.4 ± 0.4	86.0 ± 0.7	86.9 ± 0.6
Avg	72.9	76.8	85.8	86.3	88.8	90.0

simple transfer tasks ($E \rightarrow K$), the classification accuracy of FPT-BERT exceeds DAN-BERT because the difference of source and target domains is small. For hard transfer tasks ($Y \rightarrow I$, $A \rightarrow I$), FPT-BERT can improve more performance compared with BERT. The DAN-BERT improves the classification accuracy by 3.7% and 1.2% than IATN-BERT and FPT-BERT respectively. It proves that our method can extract better features.

4.4 Feature Visualization

To intuitively understand the effect of further per-training and dual adversarial, we visualize the features of the variants of BERT on source and target domains as shown in Fig. 3. We perform the visualization on $B \rightarrow D$ task by t-SNE. In the BERT (3a), the samples of different categories of the source domain are well distinguished. However, there are still some samples of the target domain are confused. Because the BERT is continuously trained, the classification accuracy is improved (3b). For DAN-BERT model (3c), the domain discrepancy is decreased and the domain-invariant features are distilled. The data from different domains are fully mixed by adversarial training.

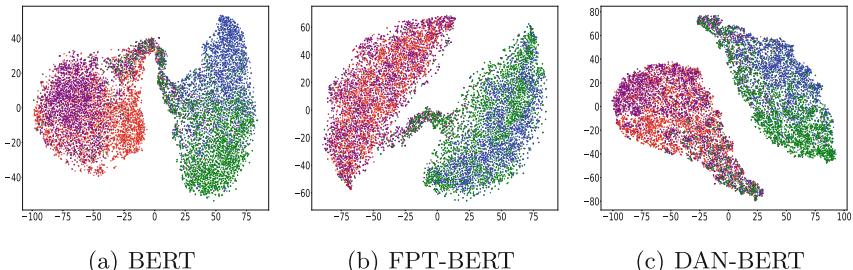


Fig. 3. The t-SNE visualization of BERT (a), FPT-BERT (b) and DAN-BERT (c) on $B \rightarrow D$ task. The red, blue, purple and green points denote the source positive, source negative, target positive and target negative examples correspondingly.

4.5 Ablation Studies

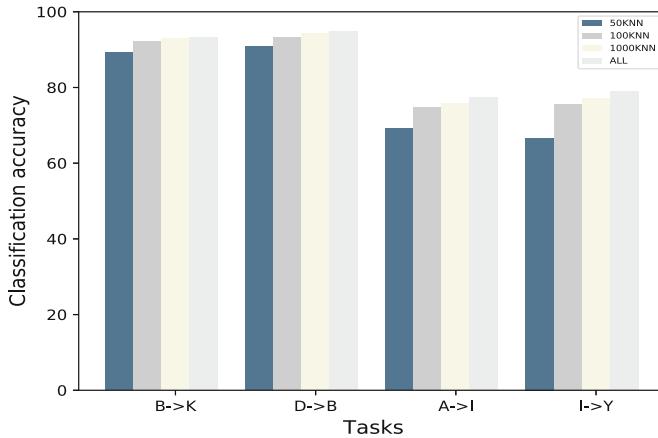
To analyze the effect of domain-invariant features at sentence (S) and aspect-based syntax (AS) level. we conduct the ablation experiments on task $B \rightarrow D$, $E \rightarrow V$, $V \rightarrow D$, $I \rightarrow A$ and $Y \rightarrow I$ in Table 3. The experimental results show that sentence and aspect-based syntax features can improve the classification accuracy.

Table 3. Results of ablation study. -w/o means without.

Model	BERT	-w/o S	-w/o AS	DAN-BERT
B → K	90.5 ± 0.4	94.5 ± 0.3	94.2 ± 0.2	94.8 ± 0.3
D → B	90.3 ± 0.5	92.8 ± 0.4	92.4 ± 0.3	93.0 ± 0.3
I → Y	80.8 ± 1.7	84.1 ± 0.6	83.8 ± 0.7	87.1 ± 0.5
Y → A	83.8 ± 1.6	85.2 ± 0.6	84.9 ± 0.7	87.3 ± 0.5
A → I	70.6 ± 1.6	74.1 ± 0.3	73.3 ± 0.8	77.4 ± 0.4

4.6 Effects of K

To verify the effect of k NN, we select different k on task $B \rightarrow K$, $D \rightarrow B$, $A \rightarrow I$ and $I \rightarrow Y$ in Fig. 4. For simple transfer tasks ($B \rightarrow K$, $D \rightarrow B$), as k increases, the performance is improved by 4.1% on average. For difficult transfer tasks ($A \rightarrow I$ and $I \rightarrow Y$), the improvement is 10.2% on average. Because the distribution of source and target domains is close on ($B \rightarrow K$, $D \rightarrow B$), we effectively train the classifier by small source domain data. We need more training data from source domain on ($A \rightarrow I$ and $I \rightarrow Y$) and the improvement is obvious.

**Fig. 4.** Classification accuracy (%) with different numbers of neighbors. ALL means all source domain data.

5 Conclusion

In this paper, we propose a novel framework to enhance the performance for Cross-domain Sentiment Classification. We further pre-train the BERT on in-domain data to reduce domain discrepancy. Then, it uses the GCN to obtain

the aspect-based syntax features over syntactical dependency trees. The domain-invariant sentence and aspect-based syntax features are get through dual adversarial mechanism. Besides, we also propose a data selection method which automatically selects training data from the source domain. Experiments on the four datasets demonstrate that DAN-BERT outperforms the state-of-the-art methods.

Acknowledgments. This paper is Supported by National Key Research and Development Program of China under Grant No.2017YFB0803003 and National Science Foundation for Young Scientists of China (Grant No. 61702507).

References

- Blitzer, J., McDonald, R., Pereira, F.: Domain adaptation with structural correspondence learning. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp. 120–128 (2006)
- Devlin, J., Chang, M.W., Lee, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) (2018)
- Ganin, Y., et al.: Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* **17**(1), 2030–2096 (2016)
- Gururangan, S., et al.: Don’t stop pretraining: adapt language models to domains and tasks. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 8342–8360 (2020)
- Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
- Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, pp. 328–339 (2018)
- Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, ICLR. OpenReview.net (2017)
- Li, X., Bing, L., Li, P., Lam, W., Yang, Z.: Aspect term extraction with history attention and selective transformation. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI, pp. 4194–4200. ijcai.org (2018)
- Li, Z., Wei, Y., Zhang, Y., Yang, Q.: Hierarchical attention transfer network for cross-domain sentiment classification. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
- von Luxburg, U.: A tutorial on spectral clustering. *Stat. Comput.* **17**(4), 395–416 (2007)
- Pan, S.J., Ni, X., Sun, J.T., Yang, Q., Chen, Z.: Cross-domain sentiment classification via spectral feature alignment. In: Proceedings of the 19th international conference on World wide web, pp. 751–760. ACM (2010)
- Peters, M.E., et al.: Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, pp. 2227–2237 (2018)

13. Ramnath, S., Nema, P., Sahni, D., Khapra, M.M.: Towards interpreting BERT for reading comprehension based QA. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pp. 3236–3242 (2020)
14. Sharma, R., Bhattacharyya, P., Dandapat, S., Bhatt, H.S.: Identifying transferable information across domains for cross-domain sentiment classification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL, pp. 968–978 (2018)
15. Sun, C., Qiu, X., Xu, Y., Huang, X.: How to fine-tune BERT for text classification? In: Sun, M., Huang, X., Ji, H., Liu, Z., Liu, Y. (eds.) CCL 2019. LNCS (LNAI), vol. 11856, pp. 194–206. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32381-3_16
16. Tang, D., Qin, B., Feng, X., Liu, T.: Target-dependent sentiment classification with long short term memory. arXiv preprint [arXiv:1512.01100](https://arxiv.org/abs/1512.01100) (2015)
17. Tang, D., Qin, B., Liu, T.: Learning semantic representations of users and products for document level sentiment classification. 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), pp. 1014–1023 (2015)
18. Yang, J., et al.: Towards making the most of BERT in neural machine translation. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, pp. 9378–9385. AAAI Press (2020)
19. Yu, J., Jiang, J.: Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 236–246 (2016)
20. Zhang, K., Zhang, H., Liu, Q., Zhao, H., Zhu, H., Chen, E.: Interactive attention transfer network for cross-domain sentiment classification (2019)



Syntax and Sentiment Enhanced BERT for Earliest Rumor Detection

Xin Miao^{1(✉)}, Dongning Rao¹, and Zhihua Jiang²

¹ School of Computer, Guangdong University of Technology, Guangzhou, China
miaox@mail2.gdut.edu.cn, raodn@gdut.edu.cn

² College of Information Science and Technology, Jinan University,
Guangzhou, China
tjiangzhh@jnu.edu.cn

Abstract. With the rapid development of social media, rumor is becoming an increasingly significant problem. Although quite a few researches have been proposed recently, most of methods rely on contextual information or propagation pattern of reply posts. For some threatening rumors, we need to interrupt their transmission in the beginning. To solve this problem, we propose Syntax and Sentiment Enhanced BERT (SSE-BERT), which can achieve superior performance only based on source post. SSE-BERT can learn extra syntax and sentiment features by additional linguistic knowledge. Experimental results on two real-word datasets show that our method outperforms some state-of-the-art methods on earliest rumor detection. Furthermore, to alleviate the shortage of Chinese dataset, we collect a new rumor detection dataset Weibo20 (The experimental resource is available <https://github.com/SeanMiao95/SSE-BERT>).

Keywords: Earliest rumor detection · BERT promotion · Linguistic knowledge integration

1 Introduction

With the increasing proportion of people who acquires information from social media [12], the risk of online rumor is being more obvious. Rumor is typically defined as a statement whose truth value is unverified or deliberately false [11]. Rumor not only causes economic damages, but also could be life-threatening. For instance, it was reported that at least 800 people died and 5800 were admitted to hospital as a result of false information related to the COVID-19 pandemic [13]. Accordingly, it is significant to identify misinformaiton without delay, especially for the life-threatening speech, which may cause injury to any receiving user.

Rumor detection can be divided into three stages corresponding to the life-cycle of rumor, viz. earliest rumor detection, early rumor detection and general rumor detection. Figure 1 shows the scope of each stage in rumor dissemination.

The design philosophy of existing rumor detection methods can be aligned with aforementioned three stages. The majority of existing methods focuses on

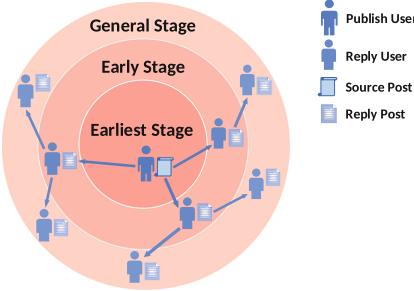


Fig. 1. Scope illustration of rumor dissemination on each stage.

general rumor detection. For example, Bian et al. [1] utilize bi-directional graph convolutional network (Bi-GCN) to learn propagation features, which achieves state-of-the-art results. Khoo et al. [5] extract interaction features from source post and reply posts by transformer model. Although they can achieve impressive results, the performance will decline significantly in absence of reply posts.

With the development of rumor detection, few methods designed for early rumor detection have been proposed. For example, Zhou et al. [22] leverage deep reinforcement learning to enable classifier to make judgement with less reply posts. Xia et al. [16] employ Kleinberg algorithm to segment reply posts into several sub-groups based on state transition, which enables encoder to capture fine-grained state features on early stage. Generally, these methods can achieve better performance on early detection, but they still depend on early reply posts.

Earliest rumor detection is challenging, because of only source post is available. The method proposed by Xu et al. [17] is the only work focuses on earliest rumor detection. It applies textual and topic features of source post to detect rumor. Although it needn't reply posts, the performance remains mediocre. It not considers the further linguistic features of rumor, and not leverags transformer-based model, which has been proven to possess strong feature-extraction ability.

Devlin et al. [2] propose Bidirectional Encoder Representations from Transformers (BERT), which is pre-trained based on a large corpus, possessing abundant linguistic knowledge. But BERT lacks advanced knowledge for rumor detection. Inspired by rumor psychology, we propose Syntax and Sentiment Enhanced BERT (SSE-BERT) for earliest rumor detection. Knapp [6] indicates that successful rumor is short, simple, and salient, which means that rumor contains specialized syntax features compared with non-rumor. To introduce syntax knowledge into BERT, we propose Dependency Encoder (DE) module, which employs off-the-shelf dependency parser to generate dependency tree of source post, then converts the dependency tree to dependency sequence by preorder traversal. BERT can learn syntax features from the dependency sequence, which solves the problem that BERT cannot process tree-structured data. Additionally, DiFonzo et al. [3] reveal that anxiety predicts rumor activity, evoking emotion is the nature of rumor. And previous works [4,14] have demonstrated the effectiveness of sentiment features on general rumor detection, but they not consider to

introduce fine-grained sentiment knowledge into BERT. To solve the problem, We propose Sentiment Recognizer (SR) module, which introduces fine-grained sentiment embeddings into BERT based on off-the-shelf sentiment lexicon. SR module can improve the perception of sentiment for BERT. Compared with some state-of-the-art methods, SSE-BERT achieves better performance on two real-world datasets, reaching 94.7% and 94.3% accuracy respectively only depend on source post. The main contributions of this work are as follows:

- To alleviate the shortage problem of Chinese dataset, we collect a new rumor detection dataset Weibo20, which is crawled from social media Sina Weibo¹.
- We propose SSE-BERT, which integrates syntax and sentiment knowledge for earliest rumor detection. To the best of our knowledge, this is the first study of introducing syntax or fine-grained sentiment knowledge into BERT.
- Our proposed SSE-BERT outperforms all compared methods on earliest rumor detection, which fills the vacancy of current methods on earliest stage.

2 Related Work

General Rumor Detection. Most of existing methods not consider the limitation of using reply posts. Early methods [10, 15, 19] employ Support Vector Machine (SVM) as classifier. The feature engineering-based methods are biased and time-consuming. Recently, deep learning methods achieve better performance. The most common methods [4, 9, 14] input source and reply posts into Recurrent Neural Network (RNN) to learn temporal features. Most recently, the emerging methods [1, 7] attempt to identify the propagation pattern by Graph Convolutional Network (GCN). In addition, transformer-based method [5] also obtains competitive performance. Although these general detection methods can reach impressive results, the performance will decline when reply posts are unavailable.

Early Rumor Detection. Few methods consider the limitation of early stage, only early reply posts are allowed to be used. The explicit method [22] leverages reinforcement learning to lead the rumor detection module in making earlier judgments. The implicit methods [16, 20] improve the early performance by incorporating additional features, viz. user credibility features and fine-grained state features respectively. The additional features can make up for the insufficient of features on early stage. Although these early detection methods rely on less reply posts, the performance will decline when the early reply posts are not available.

Earliest Rumor Detection. Almost no method consider the limitation of earliest stage, no reply post is available. Majority of researchers ignores this situation, the only method [17] combines textual and topic features of source post to improve the performance. This method only considers topic features as supplement, ignoring the further linguistic features of rumor, e.g. syntax and sentiment.

¹ <http://weibo.com>.

Moreover, it not leverages transform-based model, which has been proven to possess strong feature-extraction ability. Therefore, it performs poorly. To address this issue, we enhance BERT with syntax and sentiment knowledge. Compared with above methods, our method achieves better performance on earliest stage.

3 Problem Formulation

Let $D = \{(s_1, y_1), (s_2, y_2), \dots, (s_{|D|}, y_{|D|})\}$ be the dataset of rumor detection, where s_i is the i -th source post, y_i is the corresponding label and $|D|$ denotes the total number of the source posts. On social media, source post s_i is usually a short text, which contains the incident description released by publish user. Rumor detection is regarded as binary classification task viz. $y_i \in \{0, 1\}$, where $y_i = 0$ represents source post s_i is non-rumor, then $y_i = 1$ represents s_i is rumor.

In short, earliest rumor detection can be described as follow: given a set of source posts $S = \{s_1, s_2, \dots, s_{|D|}\}$, we need to predict the corresponding labels $Y = \{y_1, y_2, \dots, y_{|D|}\}$. These notations express the same meaning in the following.

4 The Proposed SSE-BERT Model

4.1 Overall Architecture

The overall architecture of SSE-BERT is shown in Fig. 2. SSE-BERT is composed of three components, DE module, SR module and BERT. DE module employs dependency parser to generate the dependency tree of source post, then converts the dependency tree to dependency sequence by preorder traversal. SR

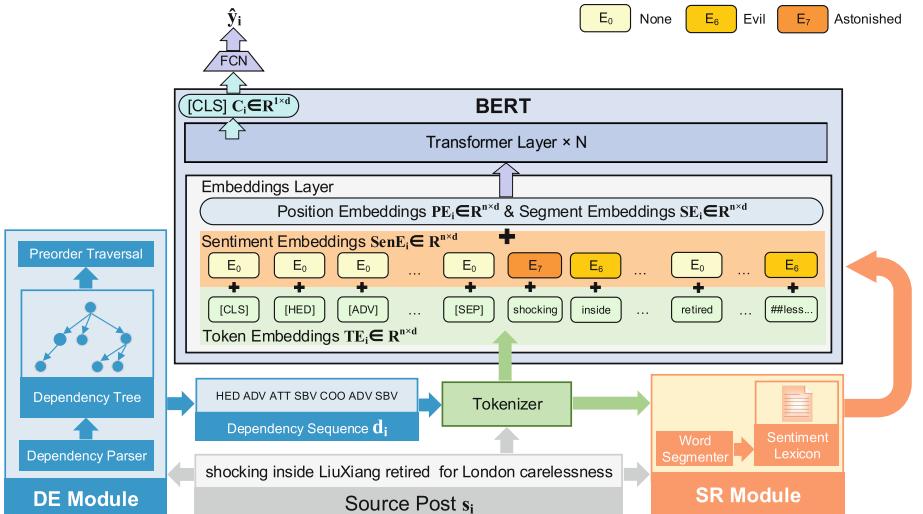


Fig. 2. The overall architecture of SSE-BERT, the source post is a real-world example.

module introduces fine-grained sentiment embeddings into BERT, then assigns specific embeddings for each token according to sentiment lexicon. After syntax and sentiment knowledge is introduced, BERT model learns contextual representation from dependency sequence and source post. The final representation vector of [CLS] token is fed into Fully Connected Network (FCN) for prediction.

4.2 Dependency Tree Encoding

Dependency parsing aims to annotate sentences into a dependency tree, which is designed to be easy for humans and computers alike to understand. Although dependency tree contains syntax knowledge, BERT cannot process tree-structured data. Our proposed DE module solves this problem, the details of DE module is shown in Fig. 3. We first employ dependency parser DDParse [21] to annotate source post s_i into dependency tree, each node represents a word or entity, each edge represents the relationship between two nodes, annotated with a specific relation tag. Then we traverse the dependency tree by preorder traversal, and return the relation tags in turn, the ordered relation tags is the dependency sequence d_i . BERT learns syntax features from d_i , The entire process can be regarded as encoding dependency tree into sequence, it can be formulated as:

$$d_i = \text{Pre}(\text{Parser}(s_i)) \quad (1)$$

where Parser represents DDParse, Pre represents preorder traversal algorithm.

After acquiring dependency sequence, we put dependency sequence d_i in front of source post s_i before executing WordPiece tokenization. The default tokenizer of BERT is employed to project the inputs to token embeddings $\mathbf{TE}_i \in R^{n \times d}$:

$$\mathbf{TE}_i = \text{Tokenizer}([\text{CLS}]; d_i; [\text{SEP}]; s_i) \quad (2)$$

where n is the number of tokens, d is the hidden size of representation vector. Classifier token [CLS] is placed at the forefront, sentence separator token [SEP]

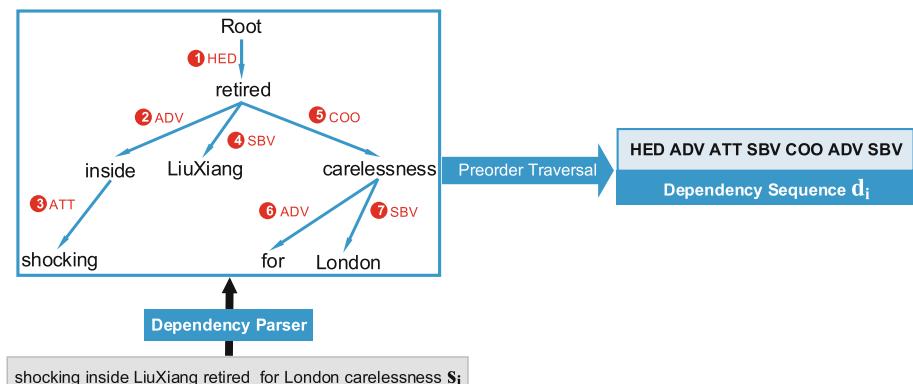


Fig. 3. The detail illustration of DE module.

is inserted between d_i and s_i . To enable BERT to recognize the relation tags, we add relation tokens to the original vocabulary of BERT, e.g. [ADV] token represents the ADV relation tag, this is benefit from the reserved position in the vocabulary. The detailed explanation of the relation tags are explained in [21].

4.3 Sentiment Words Recognition

To enable BERT to recognize sentiment words, we introduce external sentiment knowledge viz. fine-grained sentiment lexicon [18], which divides emotion words into seven categories by specialists, including “fun”, “good”, “angry”, “sad”, “fearful”, “evil” and “astonished”. To introduce the sentiment knowledge into BERT conveniently, we define a new type of embeddings viz. sentiment embeddings for BERT. Except above mentioned seven emotion types, we also define None type in sentiment embeddings, which corresponds to non-sentiment words.

SR module is used to assign corresponding sentiment embeddings for each input token. First, we need to figure out the sentiment words in source post s_i :

$$\text{SenWords}_i = \text{Lookup}(\text{Segmenter}(s_i), \text{lexicon}) \quad (3)$$

where *Segmenter* represents the word segmenter [8], it is essential for Chinese datasets, *lexicon* represents sentiment lexicon. SenWords_i is the sentiment words in s_i , which is found by *Lookup* function. Before getting the corresponding sentiment embeddings $\text{SenE}_i \in R^{n \times d}$, we need to align SenWords_i with TE_i :

$$\text{SenE}_i = \text{Align}(\text{SenWords}_i, \text{TE}_i) \quad (4)$$

where SenE_i has the same dimension size as TE_i . *Align* function assigns corresponding sentiment embeddings for each token in TE_i , e.g. Astonished embeddings for the `shocking` token, Evil embeddings for the `inside` token, None embeddings for the non-sentiment word and relation tokens, shown in Fig. 2.

4.4 Results Prediction

Except for token embeddings, BERT also contains position embeddings and segment embeddings. Position embeddings $\text{PE}_i \in R^{n \times d}$ follows the default settings, which measures space by absolute distance. For segment embeddings $\text{SE}_i \in R^{n \times d}$, we apply different segment embeddings between relation tokens and word tokens, which allows BERT to distinguish different parts of the input.

After aforesaid embeddings are determined, BERT conducts element-wise addition on these embeddings, and receives the final embeddings $\text{E}_i \in R^{n \times d}$:

$$\text{E}_i = \text{Add}(\text{TE}_i, \text{SenE}_i, \text{PE}_i, \text{SE}_i) \quad (5)$$

where *Add* represents element-wise addition. E_i is sent to transformer layer of BERT, the high-level representation vectors is computed by multi-head self-attention mechanism. After computing, the representation vector of [CLS] token $\text{C}_i \in R^{1 \times d}$ is sent to FCN to predict the label of s_i , which can be formulated as:

$$\text{C}_i = \text{Transformer}(\text{E}_i)[0] \quad (6)$$

$$\hat{y}_i = \text{Softmax}(\text{FCN}(C_i)) \quad (7)$$

where $\hat{y}_i \in R^{1 \times c}$ is a vector of probabilities for all classes of label, c is the number of classes. All embeddings viz. \mathbf{TE}_i , \mathbf{SenE}_i , \mathbf{PE}_i and \mathbf{SE}_i are learnable parameters of SSE-BERT. During the training process, we train all the parameters by minimizing the cross-entropy of the predictions \hat{Y} and ground truth Y .

5 Experiments

5.1 Datasets and Settings

Datasets. We evaluate our proposed method on two real-world datasets viz. Ma-Weibo [9] and Weibo20. Ma-Weibo is a widely used dataset for rumor detection, but it is collected before 2016, which causes the data missing of latest rumors. To fill the gap and verify the generalization performance of SSE-BERT and compared methods, we collect a new dataset Weibo20 from Sina community management center², which reports verified rumors in public. We collect 3034 rumors and 3034 non-rumors in the last five years (2016–2020), which is complementary to Ma-Weibo. The statistics of datasets are shown in Table 1.

Table 1. Statistics of the datasets.

Statistic	Ma-Weibo	Weibo20
# of source posts	4664	6068
# of non-rumors	2351	3034
# of rumors	2313	3034
Avg. words per source	105	88
Time span	2010–2015	2016–2020

Experimental Setup. We compare our method with some state-of-the-art and classical baselines, which includes the methods of different design philosophy:

- **SVM-TS** [10]: A linear SVM classifier that utilizes handcrafted features to construct time-series model. Including content, user, and diffusion features.
- **CGRU** [14]: A two-layer Cascaded Gated Recurrent Unit (CGRU) model that incorporates fine-grained sentiment features based on sentiment lexicon.
- **Bi-GCN** [1]: A rumor detection model based on Bi-directional Graph Convolutional Network (Bi-GCN), learning propagation features by reply posts.
- **PLAN** [5]: A Post-Level AttentioN (PLAN) model that learns interaction features between source post and reply posts by a transformer-based unit.
- **ERD** [22]: An Early Rumor Detection(ERD) system based on GRU units, leveraging deep reinforcement learning to detect rumor with less reply posts.

² <http://service.account.weibo.com>.

- **STN** [16]: A State-independent and Time-evolving Network (STN) model that learns state features based on Convolutional Neural Network (CNN).
- **TDRD** [17]: An rumor detection model that leverages two CNN units to extract the textual features and topic features of source post respectively.
- **SSE-BERT**: our proposed Syntax and Sentiment Enhanced BERT (SSE-BERT) that integrates syntax and sentiment knowledge for rumor detection.

We implement SVM-TS by Scikit-learn³, reproducing Bi-GCN by its open-source code, other neural network models are implemented by PyTorch⁴. SSE-BERT is implemented based on BERT-base, which contains 12 transformer layers, and the hidden size is 768. In training process, all parameters initialized with the pre-trained model released by HuggingFace⁵, except for sentiment embeddings, which is initialized by random number. The max sequence length is set to 256, learning rate is set to 2e-5, and warmup proportion is set to 0.1. To make a fair comparison, we use the same settings presented in the original papers for comparison methods, and randomly split the datasets into five parts, conducting 5-fold cross-validation to obtain robust results. We adopt Accuracy (Acc.), Precision (Pre.), Recall (Rec.), and F1 score (F_1) as the evaluation metrics.

5.2 Results of Earliest Detection

Table 2. Primary results of compared methods on two experimental datasets.

Design Philosophy	Method	Ma-Weibo				Weibo-20			
		F_1	Rec.	Pre.	Acc.	F_1	Rec.	Pre.	Acc.
General detection	SVM-TS	0.748	0.754	0.778	0.753	0.731	0.734	0.743	0.734
	CGRU	0.855	0.857	0.865	0.856	0.860	0.861	0.867	0.861
	Bi-GCN	0.892	0.892	0.896	0.892	0.881	0.882	0.889	0.882
	PLAN	0.864	0.864	0.865	0.864	0.852	0.852	0.853	0.852
Early detection	ERD	0.872	0.872	0.877	0.873	0.864	0.865	0.871	0.865
	STN	0.895	0.895	0.899	0.895	0.886	0.886	0.888	0.886
Earliest detection	TDRD	0.867	0.867	0.867	0.867	0.862	0.862	0.864	0.862
	SSE-BERT	0.947	0.947	0.948	0.947	0.943	0.942	0.943	0.943

Table 2 shows the performance of all experimental methods on Ma-Weibo and Weibo20 datasets. All methods are divide into three groups by design philosophy.

First, SSE-BERT outperforms all general detection methods. Bi-GCN is the state-of-the-art method, which can achieve impressive performance with reply posts, but the performance decline significantly when reply posts are unavailable. It indicates that methods designed for general detection cannot give full play to

³ <http://scikit-learn.org>.

⁴ <http://pytorch.org>.

⁵ <http://huggingface.co>.

its performance on earliest stage, as a result of lacking additional features of reply posts. SSE-BERT can achieve at least 5% improvement than these methods.

Second, SSE-BERT gains at least 4% improvement than early detection methods. Although ERD and STN are designed for detecting rumor rely on fewer reply posts, they still encounter performance decline when no reply post is available, which indicates that early reply posts are essential for these methods.

Finally, SSE-BERT is significantly superior than the earliest detection method. Although TDRD is designed for earliest stage, it performs poorly. In our analysis, there are two main reasons to explain this problem. The first, the applied topic features is not salient at least for earliest rumor detection. The second, the applied CNN-based model's ability of feature extraction seems somehow limited. In any case, SSE-BERT acquires about 8 % improvement than TDRD.

5.3 Ablation Study

To analyze the effect of each module, we report how each of SSE-BERT component contributes by removing each one from the entire model. Below DE-BERT and SR-BERT represent that only the DE module or SR module is equipped respectively. BERT denotes the original state. The results are presented in Fig. 4, which applies Accuracy (Acc.) as metric. We can find that every module indeed plays significant contribution equally, SR module only performs little better than DE module. The results of DE-BERT and SR-BERT confirm that syntax or sentiment features can facilitate the performance of rumor detection, which is in line with the conclusion of psychology researches [3, 6]. Moreover, BERT can achieve considerable results, which shows the strong extraction ability of BERT.

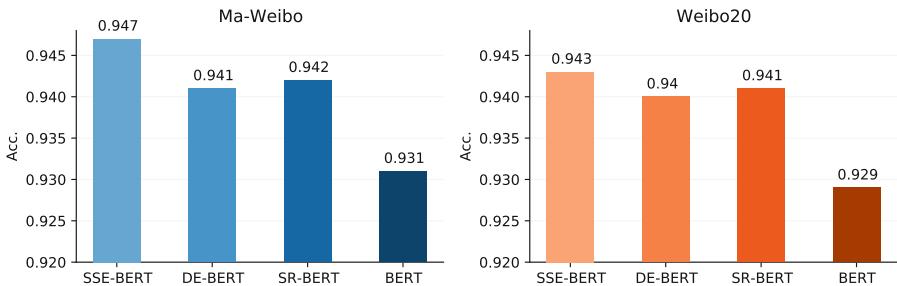


Fig. 4. The ablation results.

5.4 Analysis Study

To further investigate the reasons for the differences of syntax and sentiment between rumors and non-rumors respectively, we propose two quantitative metrics to explore the saliency, which explains why DE module and SR module work.

The **Relative Distance to HED** (hereinafter abbr. Dis) is to calculate the “hop distance” between relation token and HED token. To find out the differences of dependency sequences between rumors and non-rumors, we first split dataset into two subsets by labels, then calculate average Dis (AvgDis) for each type of relation token on the two subsets respectively, the process can be formulated as:

$$AvgDis(r, l) = \frac{\sum_{d_i}^{D_l} \sum_{t=r \cap t \in d_i} Dist_t}{|t|} \quad (8)$$

where r is a certain type of relation token, D_l represents l -label subset, d_i is the dependency sequence of i -th source post, t is a relation token in d_i . $t = r$ represents t is r type token. $|t|$ is the total number of r type token in subset D_l .

The distribution results of all type of relation tokens are shown in Fig. 5. We observed a significant differences in the distribution curves between rumors and non-rumors on each dataset, which indicates that there is indeed syntax differences between rumors and non-rumors, visually confirming the conclusion of psychology research [6]. Furthermore, we also notice obvious differences between two datasets, we assume it is due to the emergence of different events over time.

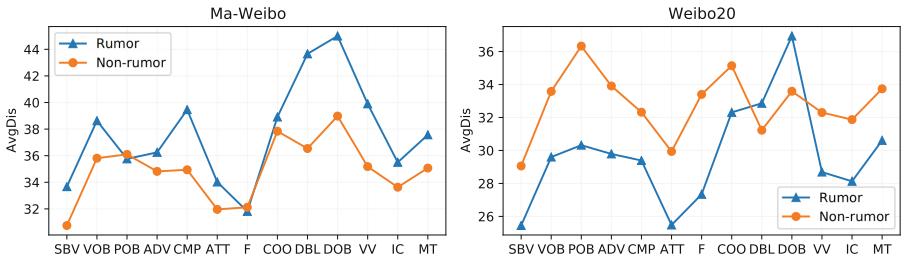


Fig. 5. The distribution of relation tokens.

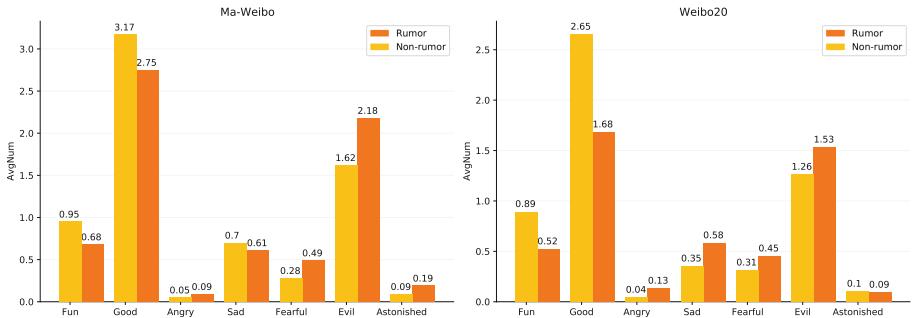


Fig. 6. The distribution of sentiment words.

The **Average Sentiment Number** (hereinafter abbr. AvgNum) represents the average number of sentiment words in each source post. To figure out the

sentiment differences between rumors and non-rumors, we also need to split datasets into two subsets by labels at first, and then calculate AvgNum for each type of emotion on two subsets respectively. The process can be formulated as:

$$AvgNum(e, l) = \frac{\sum_{s_i}^{D_l} \sum_{w=e \cap w \in s_i} count(w)}{|D_l|} \quad (9)$$

where e represents a certain type of emotion, D_l represents l -label subset, s_i represents the i -th source post in D_l , w represents a word in s_i . $w = e$ represents w is a sentiment word of e type, $count(w)$ represents accumulating the number of e type sentiment word, $|D_l|$ represents the total number of source posts in D_l .

Figure 6 shows the sentiment distribution of all emotions on the two datasets. The emotion of “fun” and “good” are positive, the emotion of “angry”, “sad”, “fearful”, “evil” and “astonished” are negative. We observed that non-rumors contain more positive words than rumors, rumors contain more negative words than non-rumors, which indicates that rumor are usually related to negative incidents, which confirms the conclusion of rumor tends to spread anxiety [3].

6 Conclusion

In this study, we propose a Syntax and Sentiment Enhanced BERT (SSE-BERT) model for earliest rumor detection and collect a new rumor detection dataset Weibo20. SSE-BERT is able to predict whether a source post is rumor on earliest period, which introduces external syntax and sentiment knowledge into BERT. The problem scenario is more realistic and challenging than most existing studies, earliest rumor detection is significant for interrupting the dissemination of threatening rumor without delay. Evaluation results show the powerful effectiveness and the reasonable explainability of SSE-BERT. Besides, our proposed new dataset Weibo20 contains the latest rumors, which makes up for the problem of lacking timely Chinese dataset. Furthermore, we believe our proposed syntax and sentiment enhanced methods can be used for not only rumor detection, but also other NLP tasks, such as aggressive detection, deception detection, and sentiment analysis. In future work, We will test our method with more datasets in different language and explore other useful features for earliest rumor detection.

Acknowledgement. We would like to thank the anonymous reviewers for their valuable comments. This work was supported by Guangdong Basic and Applied Basic Research Foundation [grant number 2021A1515012556].

References

1. Bian, T., et al.: Rumor detection on social media with bi-directional graph convolutional networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 549–556 (2020)

2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) (2018)
3. DiFonzo, N., Bordia, P.: Corporate rumor activity, belief and accuracy. *Public Relat. Rev.* **28**(1), 1–19 (2002)
4. Guo, C., Cao, J., Zhang, X., Shu, K., Liu, H.: DEAN: learning dual emotion for fake news detection on social media. arXiv preprint [arXiv:1903.01728](https://arxiv.org/abs/1903.01728) (2019)
5. Khoo, L.M.S., Chieu, H.L., Qian, Z., Jiang, J.: Interpretable rumor detection in microblogs by attending to user interactions. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 8783–8790 (2020)
6. Knapp, R.H.: A psychology of rumor. *Public Opin. Q.* **8**(1), 22–37 (1944)
7. Lu, Y.J., Li, C.T.: GCAN: graph-aware co-attention networks for explainable fake news detection on social media. arXiv preprint [arXiv:2004.11648](https://arxiv.org/abs/2004.11648) (2020)
8. Luo, R., Xu, J., Zhang, Y., Ren, X., Sun, X.: PKUSEG: a toolkit for multi-domain Chinese word segmentation. CoRR abs/1906.11455 (2019). <https://arxiv.org/abs/1906.11455>
9. Ma, J., et al.: Detecting rumors from microblogs with recurrent neural networks (2016)
10. Ma, J., Gao, W., Wei, Z., Lu, Y., Wong, K.F.: Detect rumors using time series of social context information on microblogging websites. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 1751–1754 (2015)
11. Qazvinian, V., Rosengren, E., Radev, D., Mei, Q.: Rumor has it: identifying misinformation in microblogs. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pp. 1589–1599 (2011)
12. Shu, K., Wang, S., Liu, H.: Understanding user profiles on social media for fake news detection. In: 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), pp. 430–435. IEEE (2018)
13. Silva, A., Luo, L., Karunasekera, S., Leckie, C.: Embracing domain differences in fake news: cross-domain fake news detection using multi-modal data. arXiv preprint [arXiv:2102.06314](https://arxiv.org/abs/2102.06314) (2021)
14. Wang, Z., Guo, Y.: Rumor events detection enhanced by encoding sentimental information into time series division and word representations. *Neurocomputing* **397**, 224–243 (2020)
15. Wu, K., Yang, S., Zhu, K.Q.: False rumors detection on sina weibo by propagation structures. In: 2015 IEEE 31st International Conference on Data Engineering, pp. 651–662. IEEE (2015)
16. Xia, R., Xuan, K., Yu, J.: A state-independent and time-evolving network with applications to early rumor detection. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 9042–9051 (2020)
17. Xu, F., Sheng, V.S., Wang, M.: Near real-time topic-driven rumor detection in source microblogs. *Knowl. Based Syst.* **207**, 106391 (2020)
18. Xu, L., Lin, H., Pan, Y., Ren, H., Chen, J.: Constructing the affective lexicon ontology. *J. Chin. Soc. Sci. Tech. inf.* **27**(2), 180–185 (2008)
19. Yang, F., Liu, Y., Yu, X., Yang, M.: Automatic detection of rumor on sina weibo. In: Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics, pp. 1–7 (2012)
20. Yuan, C., Ma, Q., Zhou, W., Han, J., Hu, S.: Early detection of fake news by utilizing the credibility of news, publishers, and users based on weakly supervised learning. arXiv preprint [arXiv:2012.04233](https://arxiv.org/abs/2012.04233) (2020)

21. Zhang, S., Wang, L., Sun, K., Xiao, X.: A practical Chinese dependency parser based on a large-scale dataset. arXiv preprint [arXiv:2009.00901](https://arxiv.org/abs/2009.00901) (2020)
22. Zhou, K., Shu, C., Li, B., Lau, J.H.: Early rumour detection. 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), pp. 1614–1623 (2019)



Aspect-Sentiment-Multiple-Opinion Triplet Extraction

Fang Wang², Yuncong Li¹, Sheng-hua Zhong^{2(✉)}, Cunxiang Yin¹,
and Yancheng He¹

¹ Tencent Inc., Shenzhen, China

{liyuncong,jasonyin,collinhe}@tencent.com

² College of Computer Science and Software Engineering, Shenzhen University,
Shenzhen, China

wangfang20161@email.szu.edu.cn, csshzhong@szu.edu.cn

Abstract. Aspect Sentiment Triplet Extraction (ASTE) aims to extract aspect term (aspect), sentiment and opinion term (opinion) triplets from sentences and can tell a complete story, i.e., the discussed aspect, the sentiment toward the aspect, and the cause of the sentiment. ASTE is a charming task, however, one triplet extracted by ASTE only includes one opinion of the aspect, but an aspect in a sentence may have multiple corresponding opinions and one opinion only provides part of the reason why the aspect has this sentiment, as a consequence, some triplets extracted by ASTE are hard to understand, and provide erroneous information for downstream tasks. In this paper, we introduce a new task, named Aspect Sentiment Multiple Opinions Triplet Extraction (ASMOTE). ASMOTE aims to extract aspect, sentiment and multiple opinions triplets. Specifically, one triplet extracted by ASMOTE contains all opinions about the aspect and can tell the exact reason that the aspect has the sentiment. We propose an Aspect-Guided Framework (AGF) to address this task. AGF first extracts aspects, then predicts their opinions and sentiments. Moreover, with the help of the proposed Sequence Labeling Attention (SLA), AGF improves the performance of the sentiment classification using the extracted opinions. Experimental results on multiple datasets demonstrate the effectiveness of our approach (Data and code can be found at <https://github.com/l294265421/ASMOTE>).

Keywords: Aspect sentiment multiple opinions triplet extraction · Sequence labeling attention · Aspect sentiment triplet extraction

1 Introduction

Sentiment analysis [13, 14] is an important task in natural language understanding and receives much attention in recent years. Aspect-based sentiment analysis (ABSA) [19–21] is a branch of sentiment analysis. ABSA includes several sub-tasks, such as Aspect Term Extraction (ATE), Aspect Term Sentiment Analysis

F. Wang and Y. Li—Equal contribution.

(ATSA) and Target-oriented Opinion Words Extraction (TOWE) [7]. **Aspect terms** (or simply **aspects**) are the linguistic expressions used in sentences to refer to the reviewed entities. **Opinion terms** (or simply **opinions**) are the expressions that carry subjective attitudes in sentences. Given a sentence, ATE extracts the aspects in the sentence. Given a sentence and an aspect in the sentence, ATSA and TOWE predict the corresponding sentiment and opinions respectively. For example, given the sentence in Fig. 1, ATE extracts “lobster knuckles” and “sashimi”. ATSA predicts the negative sentiments toward “lobster knuckles” and “sashimi”. TOWE extracts “ok” and “tasteless” for “lobster knuckles” and “wasn’t fresh” for “sashimi”.

Sentence: The lobster knuckles were <u>ok</u> , but pretty <u>tasteless</u> and <u>sashimi</u> <u>wasn't</u> fresh.		
Task	Input	Output
ATE	Sentence	“lobster knuckles”, “sashimi”
	Sentence+“lobster knuckles”	negative
ATSA	Sentence + “sashimi”	negative
	Sentence + “lobster knuckles”	“ok”, “tasteless”
TOWE	Sentence + “sashimi”	“wasn’t fresh”
	Sentence	(“lobster knuckles”, negative, “ok”), (“lobster knuckles”, negative, “tasteless”), (“sashimi”, negative, “wasn’t fresh”)
ASTE	Sentence	(“lobster knuckles”, negative, (“ok”, “tasteless”)), (“sashimi”, negative, (“wasn’t fresh”))
ASMOTE(ours)	Sentence	(“lobster knuckles”, negative, (“ok”, “tasteless”)), (“sashimi”, negative, (“wasn’t fresh”))

Fig. 1. An example of our ASMOTE and several ABSA subtasks. In the sentence, the bold words are aspects and the underlined words are opinions. The red triplet extracted by ASTE is confusing, because the sentiment of “ok” is neutral rather than negative.

The individual subtasks mentioned above or a combination of two subtasks can only answer one question or two questions, but can not tell a complete story, i.e. the discussed aspect, the sentiment toward the aspect, and the cause of the sentiment. To address this limitation, Peng et al. [16] introduced the Aspect Sentiment Triplet Extraction (ASTE) task. A triplet extracted from a sentence by ASTE includes an aspect, the sentiment that the sentence expresses toward the aspect, and one opinion about the aspect in the sentence. Given the sentence in Fig. 1, ASTE extracts three triplets, (“lobster knuckles”, negative, “ok”), (“lobster knuckles”, negative, “tasteless”) and (“sashimi”, negative, “wasn’t fresh”).

However, one triplet extracted by ASTE only includes one opinion of the aspect, but an aspect in a sentence may have multiple corresponding opinions. One opinion of the aspect with multiple opinions only provides part of the reason why the aspect has the sentiment, resulting in some triplets extracted by ASTE are hard to understand and provide erroneous information for downstream tasks. For example, when seeing the triplet, (“lobster knuckles”, negative, “ok”), extracted by ASTE, we will be confused, because the sentiment of “ok” is neutral.

One direct solution to the problem of ASTE is to add a post-processing step after ASTE models, which combines the multiple triplets with the same aspect into one triplet. For example, the post-processing step merges the triplets extracted by ASTE from the sentence in Fig. 1 and obtains two triplets: (“lobster knuckles”, negative, (“ok”, “tasteless”)) and (“sashimi”, negative, (“wasn’t

fresh’’)). The obtained triplets are correct and understandable, since they contains all the opinions in the sentence about the aspect and the opinions in these triplets can tell the exact reason that the aspect has the sentiment. However, this solution of patching is not elegant. And, ASTE models need to extract erroneous triplets (e.g. (“lobster knuckles”, negative, “ok”)), which is unreasonable.

In this paper, we introduce a new task, Aspect Sentiment Multiple Opinions Triplet Extraction (ASMOTE). ASMOTE has the same goal as the combination of ASTE and the post-processing step. That is, ASMOTE extracts aspect, sentiment and multiple opinions triplets. One triplet extracted by ASMOTE contains all the opinions in the sentence about the aspect. The example illustrated in Fig. 1 shows the inputs and outputs of the tasks mentioned above.

We propose an Aspect-Guided Framework (AGF) for ASMOTE. AGF includes two stages. The first stage extracts aspects, and the second stage predicts the sentiments and opinions of the aspects extracted in the first stage. The ASMOTE triplets can be obtained by merging the results of the two stages. Specifically, given a sentence, the first stage uses a neural sequence labeling model to extract aspects. For each aspect extracted in the first stage, AGF generates aspect-specific representations with the guidance of the aspect. The obtained representations are used to predict the corresponding sentiment and opinions of the aspect. AGF also uses a neural sequence labeling model to extract opinions associated with the aspect. Moreover, it is intuitive that the opinions of an aspect can help models predict the sentiment of the aspect. For example, given the sentence in Fig. 1 and the aspect “sashimi”, if AGF knows that the opinion associated with “sashimi” is the phrase “wasn’t fresh”, AGF will predict the negative sentiment more easily. Based on the intuition, we propose a Sequence Labeling Attention(SLA). Specifically, SLA converts the prediction results of the neural sequence labeling model for opinion extraction into attention weights. The attention weights are used to generate an opinion representation. The opinion representation is used to predict the sentiment of the aspect. SLA is a kind of attention mechanism with supervision and sequence labeling tasks can be seen as attention supervision tasks.

Our contributions are summarized as follows:

- We introduce a new task, Aspect Sentiment Multiple Opinions Triplet Extraction (ASMOTE).
- We propose an Aspect-Guided Framework (AGF) for ASMOTE and a Sequence Labeling Attention (SLA). AGF improves the performance of the sentiment classification using extracted opinions with the help of SLA.
- Experimental results on four public datasets demonstrate the effectiveness of AGF and SLA.

2 Related Work

Aspect-based sentiment analysis (ABSA) [19–21] aims to address various sentiment analysis tasks at a fine-grained level. ABSA includes several subtasks, such as Aspect Term Extraction (ATE), Opinion Term Extraction (OTE) extracting

opinions from sentences and Aspect Term Sentiment Analysis (ATSA). Many methods have been proposed for these subtasks, such as [11, 26, 31] for ATE, [4, 24, 25] for OTE and [6, 22, 23, 35] for ATSA. Since the three subtasks are correlated in pairs, some studies improved the performances of the three subtasks by jointly modelling two or three of them. Wang et al. [24, 25] and Dai and Song [4] jointly modelled ATE and OTE. Li et al. [10] and Phan and Ogunbona [18] jointly modelled ATE and ATSA. He et al. [8] and Chen and Qian [3] jointly modelled ATE, OTE and ATSA.

Although extracting aspects and opinions as pairs is significant, the aspects and opinions extracted by the methods mentioned above are not in pairs [7]. Fan et al. [7] put forward a new subtask of ABSA: Target-oriented Opinion Words Extraction (TOWE). TOWE aims to extract the corresponding opinions with respect to the given aspect. A few methods [7, 29] have been proposed for TOWE. While TOWE assumes the golden aspect was given, Zhao et al. [34] and Chen et al. [2] explored Aspect-Opinion Pair extraction task, which aims at extracting aspects and opinions in pairs without given golden aspects.

The above tasks are still not enough to get a complete picture regarding sentiment [16]. Peng et al. [16] proposed a new subtask: Aspect Sentiment Triplet Extraction (ASTE). ASTE extracts aspect, sentiment and opinion triplets and can tell a complete story, i.e. the discussed aspect, the sentiment toward the aspect, and the cause of the sentiment. Some methods [1, 16, 28, 32, 33] have been proposed for ASTE. However, ASTE has the problem mentioned in Introduction.

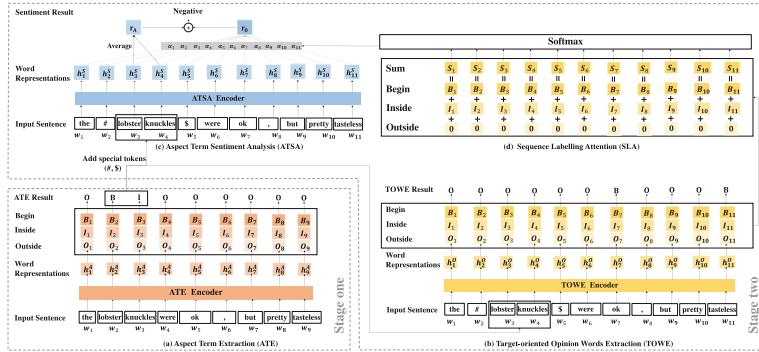


Fig. 2. Our proposed Aspect-Guided Framework (AGF) for ASMOTE.

3 Aspect-Guided Framework (AGF)

3.1 Framework

The overall architecture of our Aspect-Guided Framework (AGF) for ASMOTE is shown in Fig. 2. AGF decomposes ASMOTE into three subtasks: Aspect Term Extraction (ATE), Aspect Term Sentiment Analysis (ATSA) and Target-oriented

Opinion Words Extraction (TOWE). Given a sentence $S = \{w_1, \dots, w_i, \dots, w_n\}$, ATE extracts a set of aspects $A = \{a_1, \dots, a_j, \dots, a_m\}$. For each aspect extracted by ATE, a_j , ATSA predicts its sentiment $s_j \in \{\text{positive}, \text{neutral}, \text{negative}\}$, and TOWE extracts its opinions $O = \{o_j^1, \dots, o_j^k, \dots, o_j^{l_j}\}$. An aspect may have more than one opinion and l_j is the number of opinions with respect to the j -th aspect. ASMOTE obtains the triplets by merging the results of the three subtasks: $T = \{(a_1, s_1, (o_1^1, \dots, o_1^{l_1})), \dots, (a_m, s_m, (o_m^1, \dots, o_m^{l_m}))\}$.

AGF can be divided into two stages. The first stage performs ATE and the second stage performs ATSA and TOWE jointly. Moreover, the aspects extracted in the first stage are used to guide the sentence encoders of ATSA and TOWE to generate aspect-specific sentence representations.

3.2 Encoders

Note that AGF is a general framework, we can use any network as the encoder to learn sentence representations for ATE or aspect-specific sentence representations for ATSA and TOWE. In this paper, we implement three different encoders. The first one is the BiLSTM with pre-trained word embeddings, which has been widely used in neural-based models for NLP tasks. The second is BERT [5], a pre-trained bidirectional transformer encoder, which has achieved state-of-the-art performances across a variety of NLP tasks. The third is the BiLSTM with BERT, which has been widely used in neural sequence labeling models. The three encoders are written as $\text{Encoder}_{\text{BiLSTM_EMB}}$, $\text{Encoder}_{\text{BERT}}$, and $\text{Encoder}_{\text{BiLSTM_BERT}}$, respectively. All the three encoders take a sentence, $S = \{w_1, \dots, w_i, \dots, w_n\}$, as input, and output corresponding sentence representations, $H = \{h_1, \dots, h_i, \dots, h_n\}$.

3.3 Stage One: Aspect Term Extraction (ATE)

We formulate ATE as a sequence labeling problem. Given a sentence $S = \{w_1, \dots, w_i, \dots, w_n\}$, an encoder takes the sentence as input and outputs the corresponding sentence representation, $H^A = \{h_1^A, \dots, h_i^A, \dots, h_n^A\}$. ATE uses h_i^A to predict the tag $y_i^A \in \{B, I, O\}$ of the word w_i . It can be regarded as a three-class classification problem at each position of the sentence S . We use a linear layer and a softmax layer to compute prediction probability \hat{y}_i^A :

$$\hat{y}_i^A = \text{softmax}(W_1^A h_i^A + b_1^A) \quad (1)$$

where W_1^A and b_1^A are learnable parameters.

The cross-entropy loss of ATE task can be defined as follows:

$$L_{ATE} = - \sum_{i=1}^n \sum_{t=0}^2 \mathbb{I}(y_i^A = t) \log(\hat{y}_i^A) \quad (2)$$

where the tags $\{B, I, O\}$ are correspondingly converted into labels $\{0, 1, 2\}$ and y_i^A denotes the ground truth label. \mathbb{I} is an indicator function. If $y_i^A == t$, $\mathbb{I} = 1$, otherwise 0. We minimize the loss L_{ATE} to optimize the ATE model.

3.4 Stage Two

Since, in the same sentence, for different aspects, ATSA and TOWE models need to output different results. Therefore, it is crucial for both ATSA models and TOWE models to learn aspect-specific sentence representations. Moreover, Liang et al. [12] and Xing et al. [30] showed that utilizing the given aspect to guide the sentence encoding can obtain better aspect-specific representations. Therefore, in AGF, the aspects extracted in the first stage are used to guide the sentence encoders of ATSA and TOWE to generate aspect-specific sentence representations.

Target-Oriented Opinion Words Extraction (TOWE). We also formulate TOWE as a sequence labeling problem. Given a sentence $S = \{w_1, \dots, w_i, \dots, w_n\}$ and an aspect in the sentence, we first modify the sentence by inserting the special token # at the beginning of the aspect and the special token \$ at the end of the aspect. We then get a new sentence $S_{new} = \{w_1, \dots, \#, w_{a_s}, \dots, w_{a_e}, \$, \dots, w_{n+2}\}$, where $\{w_{a_s}, \dots, w_{a_e}\}$ are the corresponding words with respect to the given aspect. An encoder takes the new sentence as input. The special tokens explicitly tell the sentence encoder the corresponding words of the aspect in the sentence. Special tokens were first used by Wu and He [27] to incorporate target entities information into BERT on the relation classification task. We explore this method in not only BERT-based models but also LSTM-based models. The sentence encoder outputs the aspect-specific sentence representation, $H^O = \{h_1^O, \dots, h_i^O, \dots, h_{n+2}^O\}$. TOWE uses h_i^O to predict the tag $y_i^O \in \{B, I, O\}$ of the word w_i in the new sentence. It can be regarded as a three-class classification problem at each position of S_{new} . We use a linear layer and a softmax layer to compute logit \hat{l}_i^O and its probability \hat{y}_i^O :

$$\hat{l}_i^O = W_1^O h_i^O + b_1^O, \hat{y}_i^O = \text{softmax}(\hat{l}_i^O) \quad (3)$$

where W_1^O and b_1^O are learnable parameters.

The cross-entropy loss of TOWE task can be defined as follows:

$$L_{TOWE} = - \sum_{i=1}^n \sum_{t=0}^2 \mathbb{I}(y_i^O = t) \log(\hat{y}_{i_t}^O) \quad (4)$$

where the tags $\{B, I, O\}$ are correspondingly converted into labels $\{0, 1, 2\}$ and y_i^O denotes the ground truth label.

Sequence Labeling Attention (SLA). SLA converts the prediction results of TOWE into an attention vector. Specifically, the input of SLA is logits $\{\hat{l}_1^O, \dots, \hat{l}_i^O, \dots, \hat{l}_{n+2}^O\}$. SLA then obtains a vector:

$$\beta = [\beta_1, \dots, \beta_i, \dots, \beta_{n+2}] \quad (5)$$

where β_i is computed by summing the predicted logits on TOWE related to labels B and I in \hat{l}_i^O . SLA uses the softmax function on β to get the attention weight vector:

$$\alpha = [\alpha_1, \dots, \alpha_i, \dots, \alpha_{n+2}] \quad (6)$$

SLA can take probabilities $\{\hat{y}_1^O, \dots, \hat{y}_i^O, \dots, \hat{y}_{n+2}^O\}$ as input. When taking probabilities as input, SLA will behave differently.

Aspect Term Sentiment Analysis (ATSA). We formulate ATSA as a text span-based classification problem. In this paper, we use the given aspect to guide the sentence encoding in a similar manner as the TOWE task. Given a sentence $S = \{w_1, \dots, w_i, \dots, w_n\}$ and an aspect in the sentence, we first modify the sentence by inserting the special token # at the beginning of the aspect and the special token \$ at the end of the aspect. We then get a new sentence $S_{new} = \{w_1, \dots, \#, w_{a_s}, \dots, w_{a_e}, \$, \dots, w_{n+2}\}$, where $\{w_{a_s}, \dots, w_{a_e}\}$ are the corresponding words with respect to the given aspect. An encoder takes the new sentence as input and outputs the corresponding sentence representation, $H^S = \{h_1^S, \dots, h_{a_s}^S, \dots, h_{a_e}^S, \dots, h_{n+2}^S\}$. We then obtain the aspect representation by averaging the corresponding hidden states:

$$r_A = \frac{1}{(a_e - a_s + 1)} \sum_{i=a_s}^{i=a_e} h_i^S \quad (7)$$

We use the attention vector α generated by SLA to obtain the opinion representation:

$$r_O = H^S \alpha^T \quad (8)$$

AGF concatenates r_A with r_O to get the aspect-specific sentence representation for ATSA:

$$r = [r_A; r_O] \quad (9)$$

The aspect-specific representation is then used to predict the sentiment polarity of the aspect. Formally, its sentiment distribution is calculated by:

$$p = \text{softmax}(W_2^S(\text{ReLU}(W_1^S r + b_1^S)) + b_2^S) \quad (10)$$

where W_1^S , b_1^S , W_2^S and b_2^S are parameters.

We use cross entropy as the loss function:

$$L_{ATSA} = - \sum_{t=0}^2 \mathbb{I}(y^S = t) \log p_t \quad (11)$$

where y^S denotes the ground truth label and the sentiments $\{positive, neutral, negative\}$ are correspondingly converted into labels $\{0, 1, 2\}$.

Loss. The loss of the second stage is defined as follows:

$$L_{second} = L_{TOWE} + L_{ATSA} \quad (12)$$

We minimize the loss L_{second} to optimize the ATSA and TOWE model.

4 Experiments

4.1 Datasets and Metrics

We construct four datasets (i.e., 14res, 14lap, 15res, 16res) to evaluate the performance of methods on the ASMOTE task. Similar to Peng et al. [16] who constructed the Aspect Sentiment Triplet Extraction (ASTE) datasets, we obtain the four ASMOTE datasets by aligning the four Target-oriented Opinion Words Extraction (TOWE) datasets [7] and the corresponding SemEval Challenge datasets [19–21]. We do not use the ASTE datasets constructed by previous studies [16, 32] to build ASMOTE datasets (i.e., combine the multiple triplets with the same aspect in the ASTE datasets into one triplet to get ASMOTE triplets), because these datasets do not include the sentences which only contain aspects without corresponding opinion terms. We think datasets including these sentences can better evaluate the performance of methods, since methods can encounter this kind of sentences in real-world scenarios. Statistics of the ASMOTE datasets are given in Table 1. Since the number of triplets with conflict sentiment is small, these triplets are removed in our experiments.

Table 1. Dataset statistics. The tc indicates triplet with conflict sentiment.

Dataset	14res			14lap			15res			16res		
	train	dev	test									
#sentence	1615	404	606	1183	296	422	666	167	401	987	247	419
#aspect	2943	751	1134	1883	482	656	961	238	542	1391	352	612
#triplet	2116	522	864	1295	315	481	870	206	436	1206	301	456
#tc	74	12	13	31	5	14	7	2	6	17	1	8

To evaluate the performance of methods on ASMOTE, we use precision, recall, and F1-score as the metrics. A extracted triplet is regarded as correct only if predicted aspect spans, sentiment, multiple opinions spans and ground truth aspect spans, sentiment, multiple opinions spans are exactly matched.

4.2 Our Methods

AGF uses the encoder $Encoder_{BiLSTM_EMB}$ for ATE, TOWE and ATSA.

AGF-p is the pipeline version of AGF. AGF-p doesn't contain SLA and performs ATSA and TOWE separately.

AGF-t is a variant of AGF. AGF-t replaces the TOWE model jointly trained in AGF with the TOWE model trained separately. That is, AGF-t only uses the results of ATSA jointly trained in AGF.

_S are variants of AGF. *_S indicate that SLA takes probabilites rather than logits as input.

B use encoder $Encoder_{BERT}$ for ATSA, and encoder $Encoder_{BiLSTM_BERT}$ for both ATE and TOWE. The parameters of BERT are fixed during training.

BF are variants of B . BF finetune BERT during training.

4.3 Implementation Details

We implement our models in PyTorch [15]. We use 300-dimensional word vectors pretrained by GloVe [17] to initialize the word embedding vectors. $Encoder_{BERT}$ and $Encoder_{BiLSTM_BERT}$ use the uncased basic pre-trained BERT. The batch size is set to 32 for all models. All models are optimized by the Adam optimizer [9]. The learning rates are set to 0.001 and 0.00002 for non-BERT models and BERT-based models, respectively. Since the TOWE model is harder to converge than the ATSA model, for AGF, TOWE is trained first then both of TOWE and ATSA are trained together. We apply early stopping in training and the patience is 10. We run all models for 5 times and report the average results on the test datasets.

Table 2. Results of the ASMOTE task.

Method	14res			14lap			15res			16res		
	P	R	F1									
MTL	56.7	38.6	45.9	41.4	24.2	30.5	57.2	33.5	42.1	54.1	44.8	49
JET ^t ($M = 6$)	47.1	47.8	47.4	42.4	33.8	37.6	55.7	40.7	47.0	56.2	49.6	52.7
JET ^o ($M = 6$)	58.6	48.6	53.2	39.9	31.2	35.0	52.7	42.6	47.1	62.3	56.5	59.3
GTS-CNN	59.8	51.3	55.2	46.3	34.7	39.7	51.6	45.7	48.4	54.9	57.0	55.9
GTS-BiLSTM	60.4	45.8	52.1	46.0	30.3	36.5	62.2	40.9	49.3	61.8	48.9	54.6
AGF-t (ours)	62.8	56.7	59.6	46.1	35.0	39.8	55.9	45.3	50.0	62.1	58.8	60.4
JET ^t _{+bert} ($M = 6$)	50.3	53.0	51.6	44.8	35.3	39.5	55.4	44.2	49.2	51.4	56.9	54.0
JET ^o _{+bert} ($M = 6$)	57.0	47.6	51.9	43.0	33.5	37.7	58.0	47.0	51.9	66.7	54.5	60.0
GTS-BERT	63.9	61.6	62.7	51.7	44.6	47.9	57.9	53.3	55.5	56.4	64.5	60.2
AGF-t ^{BF} (ours)	63.5	64.6	64.0	46.7	49.3	48.0	56.8	54.6	55.7	57.7	69.5	63.0

4.4 Comparison Methods

We compare our methods with several methods proposed for the ASTE task, including i) five non-BERT models: MTL [33], JET^t [32], JET^o [32], GTS-CNN [28], and GTS-BiLSTM [28], ii) three BERT-based models: JET^t_{+bert} [32], JET^o_{+bert} [32] and GTS-BERT [28]. We add a post-processing step described in Introduction section after these models to obtain ASMOTE triplets.

4.5 Results

Experimental results of our methods and baselines on the ASMOTE task are reported in Table 2. From Table 2 we draw the following conclusions. First, although all the baselines are joint models, which jointly extract ASTE triplets, our pipeline models, AGF-t and AGF-t^{BF}, achieve better F1 score than their counterparts, indicating that, to achieve ASMOTE, well-designed models for ASMOTE is more effective than a combination of an ASTE model and the post-processing step. Second, AGF-t^{BF} outperforms AGF-t on all datasets in terms of F1 score, which shows that BERT can boost the performance of AGF-t.

Table 3. Results of the variants of AGF on ASMOTE, ATSA and TOWE.

Method	ASMOTE (F1)				ATSA (accuracy)				TOWE (F1)			
	14res	14lap	15res	16res	14res	14lap	15res	16res	14res	14lap	15res	16res
AGF-p	57.5	39.7	48.0	58.7	77.2	69.8	73.9	86.3	77.1	68.0	69.9	79.6
AGF	59.5	38.4	48.8	<u>58.0</u>	80.5	72.0	76.7	87.7	76.8	66.2	69.8	77.2
AGF _S	57.9	<u>38.9</u>	<u>50.0</u>	57.4	78.3	68.9	75.7	86.3	<u>77.4</u>	<u>67.2</u>	71.1	<u>78.2</u>
AGF-t	59.6	39.8	50.0	60.4	—	—	—	—	—	—	—	—
AGF-p ^B	55.4	40.6	50.2	56.2	73.8	64.3	71.0	82.4	79.7	70.4	76.5	81.6
AGF ^B	59.3	45.1	53.9	<u>58.5</u>	80.8	76.1	81.7	88.8	78.2	70.3	75.1	<u>80.5</u>
AGF _S ^B	58.4	<u>45.8</u>	53.5	56.8	79.7	74.0	77.9	88.2	79.9	71.6	<u>76.3</u>	80.3
AGF-t ^B	60.4	43.8	55.9	60.3	—	—	—	—	—	—	—	—
AGF-p ^{BF}	63.4	48.6	55.5	62.9	83.7	76.0	81.3	90.7	79.6	73.8	77.1	81.2
AGF ^{BF}	63.2	<u>48.0</u>	<u>54.5</u>	<u>62.6</u>	83.8	77.0	81.5	91.2	79.1	73.2	<u>76.2</u>	81.9
AGF _S ^{BF}	<u>63.8</u>	47.8	54.1	61.6	83.1	75.5	80.0	89.8	80.2	<u>73.7</u>	75.8	81.6
AGF-t ^{BF}	64.0	48.0	55.7	63.0	—	—	—	—	—	—	—	—

4.6 Ablation Study

Experimental Results of the variants of AGF are presented in Table 3. The underlined scores are the better scores between AGF* (i.e., AGF, AGF^B, AGF^{BF}) and AGF_S* (i.e., AGF_S, AGF_S^B, AGF_S^{BF}). From the results we draw the following conclusions. First, AGF-t* (i.e., AGF-t, AGF-t^B, AGF-t^{BF}) outperform AGF-p* (i.e., AGF-p, AGF-p^B, AGF-p^{BF}) in 11 of 12 results on the ASMOTE task and AGF* surpass AGF-p* on the ATSA task, indicating the effectiveness of SLA. Second, on ASMOTE, AGF* outperform AGF_S* in 8 of 12 results, which shows that SLA taking logits as input is a little more effective than SLA taking probabilities as input. Third, AGF* obtain better performances than AGF_S* on the ATSA task, while AGF_S* obtain better performances than AGF* on TOWE.

AGF	0.00	0.00	0.00	0.00	0.00	0.44	0.52	0.00	0.04	0.00		
AGF _S	0.08	0.08	0.08	0.08	0.08	0.19	0.21	0.08	0.08	0.08		
AGF ^B	0.00	0.00	0.00	0.00	0.00	0.16	0.84	0.00	0.00	0.00		
AGF _S ^B	0.08	0.08	0.08	0.08	0.08	0.17	0.21	0.08	0.08	0.08		
AGF ^{BF}	0.00	0.00	0.00	0.00	0.00	0.94	0.07	0.00	0.00	0.00		
AGF _S ^{BF}	0.07	0.07	0.07	0.07	0.07	0.20	0.20	0.07	0.07	0.07		
	0-The	1-#	2-bread	3-\$	4-is	5-top	6-notch	7-as	8-well	9-		

Fig. 3. Visualization of attentions.

4.7 Visualization of Attentions

Figure 3 shows the attention weights of AGF* and AGF_S* on the sentence “The bread is top notch as well”. AGF* assign more accurate weights to the opinion words, which can explain why AGF* obtain better performance on the ATSA task. This also can explain why AGF* obtain worse performance on the TOWE task. More accurate weights mean that AGF* are more confident of their prediction on the TOWE task and unfortunately obtain poor generalization ability.

5 Conclusion

In this paper, we introduce a new task, Aspect Sentiment Multiple Opinions Triplet Extraction (ASMOTE). One triplet extracted by ASMOTE includes an aspect term, the sentiment toward the aspect term, and all opinion terms associated with the aspect term in the sentence. We build four ASMOTE datasets for the ASMOTE task based on previous ATSA datasets and TOWE datasets. We propose an Aspect-Guided Framework (AGF) with a Sequence Labeling Attention (SLA) for ASMOTE. Moreover, experiments validate the effectiveness of AGF and SLA. These results provide a benchmark performance for ASMOTE.

References

- Chen, P., Chen, S., Liu, J.: Hierarchical sequence labeling model for aspect sentiment triplet extraction. In: Zhu, X., Zhang, M., Hong, Yu., He, R. (eds.) NLPCC 2020. LNCS (LNAI), vol. 12430, pp. 654–666. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60450-9_52
- Chen, S., Liu, J., Wang, Y., Zhang, W., Chi, Z.: Synchronous double-channel recurrent network for aspect-opinion pair extraction. In: ACL, pp. 6515–6524 (2020)
- Chen, Z., Qian, T.: Relation-aware collaborative learning for unified aspect-based sentiment analysis. In: ACL, pp. 3685–3694 (2020)
- Dai, H., Song, Y.: Neural aspect and opinion term extraction with mined rules as weak supervision. In: ACL, pp. 5268–5277 (2019)
- Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT, Volume 1 (Long and Short Papers), pp. 4171–4186 (2019)
- Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., Xu, K.: Adaptive recursive neural network for target-dependent Twitter sentiment classification. In: ACL (volume 2: Short papers), pp. 49–54 (2014)
- Fan, Z., Wu, Z., Dai, X., Huang, S., Chen, J.: Target-oriented opinion words extraction with target-fused neural sequence labeling. In: NAACL-HLT (2019)
- He, R., Lee, W.S., Ng, H.T., Dahlmeier, D.: An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In: ACL (2019)
- Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: ICLR (2015)
- Li, X., Bing, L., Li, P., Lam, W.: A unified model for opinion target extraction and target sentiment prediction. In: AAAI, vol. 33, pp. 6714–6721 (2019)
- Li, X., Bing, L., Li, P., Lam, W., Yang, Z.: Aspect term extraction with history attention and selective transformation. arXiv preprint [arXiv:1805.00760](https://arxiv.org/abs/1805.00760) (2018)
- Liang, Y., Meng, F., Zhang, J., Xu, J., Chen, Y., Zhou, J.: A novel aspect-guided deep transition model for aspect based sentiment analysis. In: EMNLP-IJCNLP, pp. 5572–5584 (2019)
- Liu, B.: Sentiment Analysis and Opinion Mining. Synthesis Lectures on Human Language Technologies, vol. 5, no. 1, pp. 1–167 (2012)
- Pang, B., Lee, L.: Opinion mining and sentiment analysis. Found. Trends® Inf. Retrieval **2**(1–2), 1–135 (2008)
- Paszke, A., et al.: Automatic differentiation in pytorch (2017)
- Peng, H., Xu, L., Bing, L., Huang, F., Lu, W., Si, L.: Knowing what, how and why: a near complete solution for aspect-based sentiment analysis. In: AAAI (2020)

17. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: EMNLP, pp. 1532–1543 (2014)
18. Phan, M.H., Ogunbona, P.O.: Modelling context and syntactical features for aspect-based sentiment analysis. In: ACL, pp. 3211–3220 (2020)
19. Pontiki, M., et al. SemEval-2016 task 5: aspect based sentiment analysis. In: SemEval-2016, San Diego, California, pp. 19–30. ACL, June 2016
20. Pontiki, M., Galanis, D., Papageorgiou, H., Manandhar, S., Androutsopoulos, I.: SemEval-2015 task 12: aspect based sentiment analysis. In: SemEval 2015, Denver, Colorado, pp. 486–495. ACL, June 2015
21. Pontiki, M., et al.: SemEval-2014 task 4: aspect based sentiment analysis. In: SemEval 2014, Dublin, Ireland, pp. 27–35. ACL, August 2014
22. Tang, H., Ji, D., Li, C., Zhou, Q.: Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification. In: ACL, pp. 6578–6588 (2020)
23. Wang, K., Shen, W., Yang, Y., Quan, X., Wang, R.: Relational graph attention network for aspect-based sentiment analysis. arXiv preprint [arXiv:2004.12362](https://arxiv.org/abs/2004.12362) (2020)
24. Wang, W., Pan, S.J., Dahlmeier, D., Xiao, X.: Recursive neural conditional random fields for aspect-based sentiment analysis. arXiv preprint [arXiv:1603.06679](https://arxiv.org/abs/1603.06679) (2016)
25. Wang, W., Pan, S.J., Dahlmeier, D., Xiao, X.: Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In: AAAI (2017)
26. Wei, Z., Hong, Y., Zou, B., Cheng, M., Jianmin, Y.: Don't eclipse your arts due to small discrepancies: boundary repositioning with a pointer network for aspect extraction. In: ACL, pp. 3678–3684 (2020)
27. Wu, S., He, Y.: Enriching pre-trained language model with entity information for relation classification. In: CIKM, pp. 2361–2364 (2019)
28. Wu, Z., Ying, C., Zhao, F., Fan, Z., Dai, X., Xia, R.: Grid tagging scheme for aspect-oriented fine-grained opinion extraction. In: Findings of EMNLP, pp. 2576–2585. ACL, Online, November 2020
29. Wu, Z., Zhao, F., Dai, X.Y., Huang, S., Chen, J.: Latent opinions transfer network for target-oriented opinion words extraction. In: AAAI (2020)
30. Xing, B., Liao, L., Song, D., Wang, J., Zhang, F., Wang, Z., Huang, H.: Earlier attention? Aspect-aware LSTM for aspect sentiment analysis. In: IJCAI (2019)
31. Xu, H., Liu, B., Shu, L., Philip, S.Y.: Double embeddings and CNN-based sequence labeling for aspect extraction. In: ACL, pp. 592–598 (2018)
32. Xu, L., Li, H., Lu, W., Bing, L.: Position-aware tagging for aspect sentiment triplet extraction. In: EMNLP, pp. 2339–2349. ACL, Online, November 2020
33. Zhang, C., Li, Q., Song, D., Wang, B.: A multi-task learning framework for opinion triplet extraction. In: Findings of EMNLP, pp. 819–828. ACL, Online, November 2020
34. Zhao, H., Huang, L., Zhang, R., Lu, Q., et al.: SpanMlt: a span-based multi-task learning framework for pair-wise aspect and opinion terms extraction. In: ACL, pp. 3239–3248 (2020)
35. Zhao, P., Hou, L., Wu, O.: Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification. Knowl. Based Syst. **193**, 105443 (2020)



Locate and Combine: A Two-Stage Framework for Aspect-Category Sentiment Analysis

Yang Wu, Zhenyu Zhang, Yanyan Zhao, and Bing Qin^(✉)

Harbin Institute of Technology, Harbin, China
{ywu,zyzhang,yyzhao,qinb}@ir.hit.edu.cn

Abstract. Aspect category sentiment classification aims at predicting the sentiment polarity of the given aspect category. Since the aspect category may not occur in the sentence, it is hard for the model to directly find the appropriate sentiment words for the aspect category and disregard unrelated ones. To address it, previous works have explored leveraging implicitly the information of the aspect term in the sentence and demonstrated the effectiveness of such information. Inspired by this conclusion, we propose a two-stage strategy named Locate-Combine(LC) to utilize the aspect term in a more straightforward way, which first locates the aspect term and then takes it as the bridge to find the related sentiment words. Specifically, in the “Locate” stage, we locate the aspect term corresponding to the given aspect category in the sentence, which can crystallize the target and further enable our model to focus on the target-related words. In the “Combine” stage, we first apply the graph convolutional network (GCN) over the dependency tree of the sentence to combine the information of the aspect term and related sentiment words and then take the output representation corresponding to the located aspect term to predict the sentiment polarity. The experimental results on the public datasets show that the proposed two-stage strategy is effective, which achieves state-of-the-art performance. Furthermore, our model can output explainable intermediate results for model analysis. (Code can be found at <https://github.com/SCIR-MSA-Team/LC-ACSA>)

Keywords: Aspect category sentiment classification · Aspect based sentiment analysis · Graph convolutional network.

1 Introduction

Aspect term sentiment classification (ATSC) and aspect category sentiment classification (ACSC) are two main tasks in aspect-level sentiment analysis. ATSC aims to predict the sentiment polarity toward a given aspect term occurring in the sentence, while ACSC aims at detecting the sentiment polarity of a predefined aspect category. The core problem of both two tasks is how to connect the given aspect term/category to the related sentiment words in the sentence. By comparison, it's more difficult for ACSC, since the given aspect category may

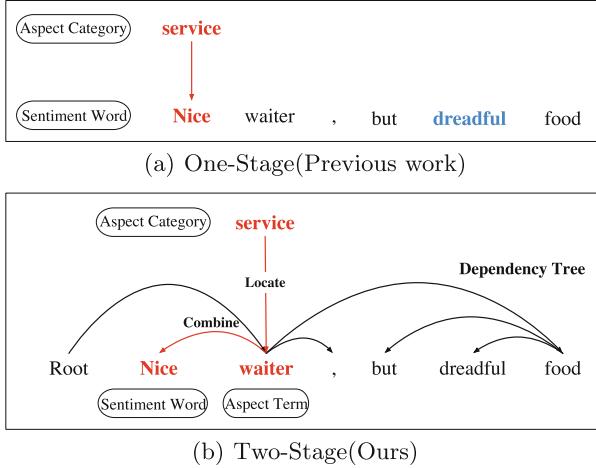


Fig. 1. Illustration of one-stage and two-stage methods.

not occur in the sentence. Previous works [8, 10] usually directly capture the relationship between the aspect category and sentiment words in the sentence via attention or gating mechanism. However, the limited information provided by the aspect category may result in mismatching between the given aspect category and the sentiment words.

To address it, some works [1, 5] take the aspect term information into consideration and propose models to implicitly leverage such information. Wang et al. [8] consider that aspect term extraction (ATE) can help the model to capture aspect-related information, which is useful for finding aspect-specific sentiment words of the sentence, and then propose the HiErarchical ATTention (HEAT) network, which consists of two main part, aspect attention and sentiment attention. The aspect attention can extract the aspect-related information to guide the sentiment attention to better allocate aspect-specific sentiment words of the sentence. However, training HEAT needs to annotate manually the aspect terms indicating the given aspect category, which is time-consuming and expensive. Li et al. [5] consider that using aspect category detection (ACD) can help the model to focus on the words related to the aspect category such as the aspect term and ignore other words, resulting in more accurate sentiment prediction. Motivated by it, they propose the Multi-Instance Multi-Label Learning Network (AC-MIMLLN), which takes the ACD task as the auxiliary task and shares the same attention layer between the ACD and ACSC blocks. Both two works demonstrate that the aspect term information is useful for the ACSC task.

In this paper, we utilize the aspect term information to tackle the mismatch problem in a more straightforward way. We propose a two-stage strategy named Locate-Combine(LC), as shown in Fig. 1(b). Comparing to the one-stage method shown in Fig. 1(a), which uses a single model to capture the relationship between the aspect category and sentiment words, we introduce the “Locate” stage, in

which we locate the aspect term related to the given aspect category in the sentence and take it as the bridge to connect the aspect category and sentiment words. We consider that explicitly locating the aspect term can bring two benefits. One is that locating the aspect term enables us to capture the interaction between the aspect term and related sentiment words in the dependency tree. The other one is that the located aspect terms can be seen as the explainable intermediate results of the model, which are useful for model analysis. Specifically, our proposed approach consists of two stages, “Locate” and “Combine”. In the “Locate” stage, we detect the corresponding aspect term for the given aspect category. As shown in Fig. 1(b), given the aspect category “service”, we locate the aspect term “waiter” in the sentence. In the “Combine” stage, we first build the dependency tree of the sentence and then apply GCN over the dependency tree of the sentence to combine the information of the aspect term “waiter” and related sentiment word “nice”.

We conduct the experiments on the public datasets and the experimental results demonstrate that our model achieves state-of-the-art performance.

The main contributions of our work can be summarized as follows:

- We propose a novel two-stage strategy named Locate-Combine for aspect category sentiment analysis, which first locates the aspect term in the sentence and then utilizes GCN over the dependency tree to combine the aspect term and related sentiment words to generate the syntax-aware representation for sentiment prediction.
- We explore the method to locate the aspect term corresponding to the given aspect category.
- We exploit syntactical dependency structures for aspect category sentiment classification.

2 Related Work

2.1 Aspect Category Sentiment Classification

Aspect category sentiment classification has been studied for many years and lots of works have been conducted on it. The core idea of these methods can be concluded using a single model to capture the sentiment words corresponding to the aspect category and thus we call this strategy as one-step strategy. Wang et al. [8] introduced the ATAE-LSTM model, which takes aspect category as the query to attend the sentiment words in the sentence. Xue et al. [10] proposed the Gated Tanh-ReLU Units, which can filter out the useless words and pick up the sentiment words according to the given aspect category. However, it is difficult for the model to directly find the appropriate sentiment words for the aspect category. To tackle it, some works design the auxiliary task to implicitly leverage the aspect term information to help the model to capture the meaningful sentiment words. Cheng et al. [1] proposed that the ATE task can help the model to capture aspect-related information, which is useful for finding aspect-specific sentiment words of the sentence, and then introduced the HiErarchical

ATtention (HEAT) network. But this method requires annotating manually the aspect terms indicating the given aspect category, which is time-consuming and expensive. Li et al. [5] proposed the Multi-Instance Multi-Label Learning Network (AC-MIMLLN), which uses the ACD task to help the model to focus on the words related to the aspect category. These works show that the aspect term information is beneficial for the ACSC task, which motivates us to explore a more effective way to utilize such information.

2.2 Aspect Term Sentiment Classification

Aspect term sentiment classification (ATSC) is similar to ACSC. But there is a substantial difference between them, which is that the target of ATSC is a given aspect term in the sentence while the target of ACSC may not occur in the sentence. This characteristic of the ATSC task enables the model to exploit more information, such as syntactical information, to connect the aspect term and the corresponding sentiment words. Huang et al. [2] proposed a target-dependent graph attention network, which utilizes the dependency relationship among words. Zhang et al. [11] presented an Aspect-specific Graph Convolutional Network (AS-GCN) to exploit syntactical information and word dependencies. Sun et al. [7] introduced a convolution over a dependency tree (CDT) model, which applies GCN on the dependency tree of the sentence to obtain the syntax-aware word representation. These previous works have demonstrated that the syntactical information, specifically the dependency relationship among words, is useful for ATSC. Inspired by this conclusion, we consider that syntactical information is also beneficial for the ACSC task, which can help the model to aggregate the target-related information including aspect term information and sentiment information.

3 Approach

In this section, we introduce our two-stage strategy named Locate-Combine shown in Fig. 2. In the “Locate” stage, we locate the aspect term corresponding to the given aspect category. In the “Combine” stage, the aspect category sentiment classification model combines the located aspect term and sentiment words by applying GCN over the dependency tree to learn syntax-aware representation.

3.1 “Locate” Stage

In the “Locate” stage, the goal is to locate the aspect term corresponding to the given aspect category in the sentence, since we consider that locating the aspect term in the sentence can provide two benefits for ACSC, which are generating explainable intermediate results and helping model to leverage syntactical information.

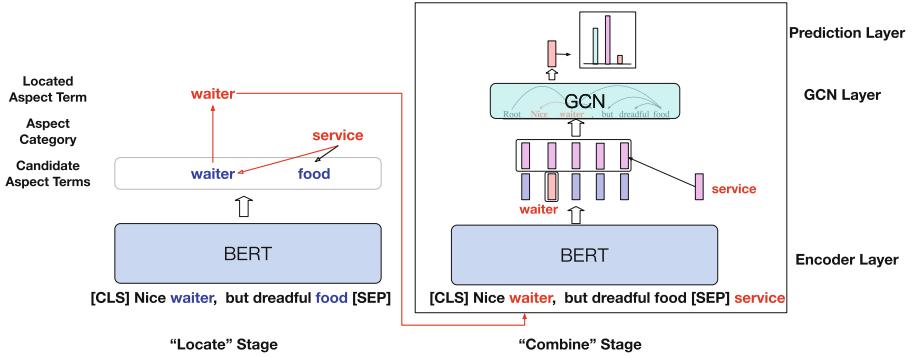


Fig. 2. Illustration of our proposed approach.

This task is formalized as follows. Given a sentence and a pre-defined aspect category set $C = \{c_1, \dots, c_m\}$, the goal is to extract the corresponding aspect term¹ t_c for each aspect category c in set C . We design a simple method to solve it, as shown in Fig. 2. We first extract all aspect terms in the sentence using the ATE model, which is implemented by BERT. Then for each extracted aspect term, we calculate the cosine similarity score between it with each aspect category in C and take the most similar aspect category c_s as the corresponding aspect category. Following this method, each extracted aspect term is assigned to an aspect category. But for some aspect categories, the ATE model may fail to find the corresponding aspect terms. Hence, we utilize an ACD model to extract the pseudo aspect term.

The ACD model consists of three layers, the encoder layer, attention layer, and prediction layer. The encoder layer is implemented by the BERT encoder and for each word, we take the representation of the first sub-token as the representation of it. Then we can obtain the representations of words, $\{e_1, \dots, e_n\}$.

In the attention layer, we calculate the category-specific attention weights p_i^c for the input word representations as follows.

$$s_i^c = W_2^c(\text{Tanh}(W_1^c(e_i) + b_1^c)) \quad (1)$$

$$p_i^c = \frac{e^{s_i^c}}{\sum_{i=1}^n e^{s_i^c}} \quad (2)$$

$$r^c = \sum_{i=1}^l p_i^c e_i \quad (3)$$

where $W_1^c \in \mathbb{R}^{300 \times 768}$, $b_1^c \in \mathbb{R}^{300}$, $W_2^c \in \mathbb{R}^{1 \times 300}$, are the parameters of the attention layer .

¹ In most cases, there is only one aspect term in the sentence corresponding to a given aspect category. Thus, we mainly consider this situation for simplicity.

Finally, we use the prediction layer to predict whether a given aspect category is mentioned.

$$v_c = \text{Sigmoid}(W_3^c(r^c) + b_3^c) \quad (4)$$

where $W_3^c \in \mathbb{R}^{1 \times 768}$, $b_3^c \in \mathbb{R}^1$, are the parameters of the attention layer.

To obtain the pseudo aspect term for the aspect category, we extract the words with the top-2 attention weights as the candidate aspect terms and take the most similar aspect term as the corresponding aspect term.

3.2 “Combine” Stage

In the “Combine” stage, the goal is to predict the sentiment polarity of the given aspect category leveraging the located aspect term information and syntactical information. Benefiting from the located aspect term, we can utilize the dependency tree to capture the sentiment words related to the aspect category in the sentence more easily using the located aspect term as the bridge. Besides, the aspect category is also very important for ACSC. Hence, we propose an aspect category sentiment classification model, which can combine the aspect term, aspect category, and syntactical features to learn target-aware representation for sentiment prediction. This model, as shown in Fig. 2, consists of three layers including the encoder layer, GCN layer, and classification layer.

In the encoder layer, we use BERT to represent the words. To incorporate the aspect category information, we concatenate the sentence with the aspect category and split them by the special token [SEP]. Specifically, the input format is “[CLS] sentence [SEP] aspect category”. We take the representation of the first sub-token for each word as the representation of it. Then we obtain the word representations, $\{w_1, \dots, w_l\}$ and the aspect category representation w_c . Besides, we also implement our encoder layer by LSTM. Specifically, we first concatenate the word embeddings with the word embedding of the given aspect category and feed them into the LSTM to obtain the word representations.

We utilize the GCN layer over the dependency tree obtained by using SpaCy to capture the interaction between the aspect term and sentiment words. Specifically, we first represent the dependency tree as an $l * l$ adjacency matrix A and $A_{i,j} = 1$ if there is an edge between node i and node j , and $A_{i,j} = 0$ otherwise. Then we obtain the node representations $h_i^{(2)}$ as follows.

$$c_i = \frac{1}{\sum_{j=1}^l A_{i,j}} \quad (5)$$

$$h_i^{(1)} = \sum_{j=1}^l c_i A_{i,j} (W_1^g([w_j; w_c]) + b_1^g) \quad (6)$$

$$h_i^{(2)} = \sum_{j=1}^l c_i A_{i,j} (W_2^g h_j^{(1)} + b_2^g) \quad (7)$$

where $W_1^g \in \mathbb{R}^{768 \times 1536}$, $b_1^g \in \mathbb{R}^{768}$, $W_2^g \in \mathbb{R}^{768 \times 768}$, $b_2^g \in \mathbb{R}^{768}$, are the parameters of the GCN layer.

Finally, we take the representation of the located aspect term as the target-aware representation² and use the softmax classifier to predict the sentiment polarity.

4 Experiment

4.1 Datasets

We evaluate our model on four datasets including Rest2014, Rest2014-hard, RestLarge, and RestLarge-hard. RestLarge is obtained by combining the data from SemEval 2014, SemEval 2015, and SemEval 2016, and we remove the samples with conflict polarity just like previous work [5]. Rest2014-hard and RestLarge-hard are constructed by collecting the examples with at least two aspects and two sentiments polarities from Rest2014 and RestLarge, respectively. The statistics of datasets are shown in Table 1.

Table 1. Statistics of the datasets

Polarity	Rest14			Rest14-hard		RestLarge		RestLarge-hard	
	Train	Dev	Test	Test		Train	Test	Train	Test
Positive	1,873	306	657	21		2,710	1,505	182	92
Negative	712	127	222	20		1,198	680	178	81
Neutral	433	67	94	12		757	241	107	61

4.2 Training Details

Aspect Term Extraction (ATE). We initialize the BERT of our ATE model using the pre-trained BERT parameters released by the previous work [9]. We train our ATE model on the training set of the RestLarge dataset. The learning rate is set to 0.00003. We set the dropout probability to 0.2.

Aspect Category Detection (ACD). Similar to our ATE model, we also initialize the BERT of our ACD models using the pre-trained BERT parameters released by the previous work [9]. We utilize the Rest2014 and RestLarge datasets to train our ACD models, which are applied for Rest2014/Rest2014-hard and RestLarge/RestLarge-hard, respectively. The learning rate is set to 0.001. The optimizer is Adam and the dropout probability is set to 0.4.

² If there are multiple aspect terms, we average the representation vectors of them and take the result as the final representation.

Aspect Category Sentiment Classification (ACSC). We initialize the word representation for the non-BERT model using 300-dimension glove word embeddings. The batch size is set to 25 for the non-BERT model and 16 for BERT-based model. The learning rate is set to 0.01 and 0.00001 for the non-BERT model and the BERT-based model, respectively. The optimizer is Adam.

4.3 Baselines

We compare our model with two lines of baselines. One line of works is aspect-aware models including AC-MIMLLN, SCAN, AC-MIMLLN-BERT, and SCAN-BERT, which capture the aspect term information implicitly. The other line of works is attention-based models including AT-LSTM, ATAE-LSTM, GCAE, CapsNet, and CapsNet-BERT, which do not take such information into the consideration.

- AT-LSTM [8] is proposed for the ATSC task, which uses the attention mechanism to obtain the target-aware representation, and we adopt it for ACSC.
- ATAE-LSTM [8] is similar to AT-LSTM, which first obtains the aspect-aware word representations and then uses the attention mechanism to obtain the final representation.
- GCAE [10] uses the Gated Tanh-ReLU Units to generate the sentiment features according to the given aspect category.
- CapsNet [3] is a novel capsule network based model proposed for ACSC.
- AC-MIMLLN [5] utilizes the attention weights extracted from the ACD model as the attention weights of the ACSC model, which can help the model to focus on the target-related words.
- SCAN [4] applies GCN over the constituency parse tree to obtain node representations and applies the attention mechanism to obtain the target-aware representation.
- CapsNet-BERT [4] is similar to CapsNet, which adopts BERT as the encoder.
- AC-MIMLLN-BERT [5] is similar to AC-MIMLLN, which takes BERT as the encoder.
- SCAN-BERT [4] is similar to SCAN, which adopts BERT as the encoder.
- Auxiliary-BERT-QA-M [6] first constructs the auxiliary sentence by generating a question and then concatenates the sentence with the auxiliary sentence as the input of BERT.

4.4 Experimental Results

We conduct the experiments on Rest2014, Rest2014-hard, RestLarge, and RestLarge-hard datasets to evaluate our strategy. As shown in Table 2, our LC-BERT model outperforms all baselines, which demonstrates the effectiveness of our method. We also find that AC-MIMLLN-BERT and SCAN-BERT surpass Auxiliary-BERT-QA-M and CapsNet-BERT, which indicates that the aspect term information is helpful for the model to focus on related sentiment words,

Table 2. Experimental results on the benchmark datasets. † refers to drawing from the original papers. The best results are bold-typed.

Models	Rest14		Rest14-hard		RestLarge		RestLarge-hard	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1
AT-LSTM	79.75	52.96	52.83	39.72	79.93	65.86	45.73	36.41
ATAE-LSTM	77.70	61.44	64.15	48.93	81.04	64.09	62.39	58.21
GCAE	80.88	68.01	62.26	47.26	85.92†	–	70.75†	–
CapsNet	80.88	69.16	67.92	58.77	83.72	72.69	71.79	70.62
AC-MIMLLN	81.50	68.21	66.04	57.33	84.91	76.08	70.94	69.96
SCAN	80.70†	–	68.30†	–	84.38	73.94	67.54†	–
LC-LSTM(Ours)	82.43	69.29	71.70	67.76	85.41	77.20	72.65	71.21
Auxiliary-BERT-QA-M	88.39	79.78	69.81	66.40	89.65	82.39	56.84	51.54
CapsNet-BERT	87.77	79.68	69.81	64.85	87.80	78.66	69.23	67.44
AC-MIMLLN-BERT	89.00	81.14	71.70	68.77	89.82	82.47	71.79	70.62
SCAN-BERT	88.61†	–	70.94†	–	88.91	80.43	71.97†	–
LC-BERT(Ours)	89.72	83.46	75.47	72.54	90.19	82.56	73.50	72.49

Table 3. Results of ablation study in terms of F1-score. The best results are in bold.

Models	Rest14	Rest14-hard	RestLarge	RestLarge-hard
LC-BERT	83.46	72.54	82.56	72.49
LC-BERT w/o aspect category	80.07	69.65	81.67	56.16
LC-BERT w/o aspect term	78.31	70.82	82.39	68.56
LC-BERT w/o GCN	79.67	65.83	82.47	71.23

resulting in more accurate sentiment prediction. Moreover, our proposed LC-BERT obtains better performance than AC-MIMLLN-BERT and SCAN-BERT, which shows that leveraging the aspect term information in this two-stage way is also highly effective. We also observe that there is a larger margin between our model and the baselines on the Rest14-hard and RestLarge-hard datasets, which indicates that our model can handle the mismatching problem better and capture the relationship between the given aspect category and sentiment words more effectively benefiting from the aspect term information and syntactical information.

4.5 Ablation Study

We conduct the ablation experiments to distinguish the contribution of each part and the results are shown in Table 3. There are several different variants of our LC-BERT model. **LC-BERT w/o aspect term** randomly selects a word representation output by the GCN layer as the final representation. **LC-BERT w/o aspect category** only takes the sentence as the input of the encoder layer without incorporating the aspect category information. **LC-BERT w/o GCN** takes the sentence and aspect category as the input and uses the word representation output by BERT corresponding to the aspect term as the final representation.

Table 4. Case studies of LC-LSTM. The “Predicted” and “Located” columns show the predicted results and located aspect terms respectively.

Sentence	Category	Gold	Prediction	Located
When the dish arrived it was blazing with green chillis , definitely not edible by a human	Food	Negative	Negative	Dish
The food is so good and so popular that waiting can really be a nightmare	Service	Negative	Negative	Waiting
At night the atmosphere changes turning into this hidden jewel that is waiting to be discovered	Ambience	Positive	Positive	Atmosphere

As shown in Table 3, ablating each part hurts the model performance, which indicates that each part of our model is useful for the sentiment prediction. Comparing LC-BERT w/o aspect term with LC-BERT, we can see that ablating the “Locate” stage decreases the model performance, which shows that the introduced “Locate” stage is useful for ACSC since we can obtain the exact position of the true target in the sentence through it. We also observe that removing the GCN layer from our model leads to a sharp reduction in performance, which indicates that leveraging the syntactical information to combine the aspect term and sentiment words over the dependence tree is effective for ACSC. Comparing LC-BERT w/o aspect category with LC-BERT, we find that the aspect category is also very important for ACSC. It is in line with our expectation because the model may locate the wrong word in the “Locate” stage and then the aspect category becomes more vital for accurate sentiment prediction.

4.6 Case Study

To have an intuitive understanding of our proposed model, we present some cases in Table 4. In the first case, our model first locates the aspect term “dish” precisely and further utilizes it to detect the sentiment polarity correctly. In the second case, our model is not confused by the word “food” and locates the right aspect term “waiting” corresponding to the aspect category “service”. In the last case, our model locates the aspect term “atmosphere” and predicts the sentiment polarity correctly. These examples show that our model can locate the aspect term for the given aspect category and further utilize the aspect term to obtain better target-aware representation.

4.7 Error Analysis

We conduct the error analysis of our proposed LC-BERT on the test set of the RestLarge-hard dataset. The results are shown in Table 5. We consider two types of errors. One is that the model fails to locate the aspect term correctly. For example, in the second line of Table 5, our model fails to locate the correct

Table 5. Error analysis of LC-BERT on the test set of the RestLarge-hard dataset. The “Stage/Percentage” column lists the types of errors and the percentage of them.

Sentence	Category	Gold	Prediction	Located	Stage/Percentage
it gets crowded at lunchtime but there are lots of seats in back and everyone who works there is so nice.	Service	Positive	Negative	Crowded	Locate/61.9%
The ambiance was pretty cool , but not worth the hassle	Misc	Negative	Positive	Hassle	Combine/38.1%

target “everyone” and extracts the word “crowded” as the target. This kind of error accounts for 61.9% of all errors. To correct it, introducing the syntactic information may be helpful. The other one is that the model locates the aspect term correctly but fails to predict the sentiment polarity of the given aspect category. We take the sample in the third line of Table 5 as an example. The model locates the right aspect term “hassle” but wrongly detects the sentiment polarity. The possible reason is that the model is confused by the words “pretty cool” and classifies this example as “positive”. This kind of error accounts for 38.1% of all errors. To address it, removing words unrelated to the given aspect category may be a potential solution.

5 Conclusion

In this paper, we propose a novel two-stage strategy named Locate-Combine to leverage the aspect term information in a more straightforward way. In the “Locate” stage, we locate the aspect term of the given aspect category, which acts as the bridge between the aspect category and the sentiment words. We then combine the located aspect term and related sentiment words to learn syntax-aware representation by utilizing GCN over the dependency tree in the “Combine” stage. The extensive experiments on public datasets demonstrate the effectiveness of our proposed two-stage strategy, which achieves state-of-the-art performance. Moreover, we conduct the comprehensive error analysis by analyzing the explainable intermediate results output by our model for further improvement. For future work, we would like to investigate more effective ways of locating the corresponding aspect term given the aspect category.

Acknowledgments. This work was supported in part by the following Grants: National Natural Science Foundation of China (No. 61632011, No. 61772153), National Key R&D Program of China (No. 2018YFB1005103).

References

1. Cheng, J., Zhao, S., Zhang, J., King, I., Zhang, X., Wang, H.: Aspect-level sentiment classification with heat (hierarchical attention) network. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 97–106 (2017)
2. Huang, B., Carley, K.: Syntax-aware aspect level sentiment classification with graph attention networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 5469–5477. Association for Computational Linguistics, Hong Kong, November 2019
3. Jiang, Q., Chen, L., Xu, R., Ao, X., Yang, M.: A challenge dataset and effective models for aspect-based sentiment analysis. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 6280–6285. Association for Computational Linguistics, Hong Kong, November 2019
4. Li, Y., Yin, C., Zhong, S.H.: Sentence constituent-aware aspect-category sentiment analysis with graph attention networks. In: Zhu, X., Zhang, M., Hong, Y., He, R. (eds.) NLPCC 2020. LNCS (LNAI), vol. 12430, pp. 815–827. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60450-9_64
5. Li, Y., Yin, C., Zhong, S.H., Pan, X.: Multi-instance multi-label learning networks for aspect-category sentiment analysis. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 3550–3560 (2020)
6. Sun, C., Huang, L., Qiu, X.: Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Long and Short Papers), vol. 1, pp. 380–385. Association for Computational Linguistics, Minneapolis, June 2019
7. Sun, K., Zhang, R., Mensah, S., Mao, Y., Liu, X.: Aspect-level sentiment analysis via convolution over dependency tree. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 5679–5688. Association for Computational Linguistics, Hong Kong, November 2019
8. Wang, Y., Huang, M., Zhu, X., Zhao, L.: Attention-based LSTM for aspect-level sentiment classification. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 606–615. Association for Computational Linguistics, Austin, November 2016
9. Xu, H., Liu, B., Shu, L., Philip, S.Y.: Bert post-training for review reading comprehension and aspect-based sentiment analysis. In: NAACL-HLT, vol. 1 (2019)
10. Xue, W., Li, T.: Aspect based sentiment analysis with gated convolutional networks. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), vol. 1, pp. 2514–2523. Association for Computational Linguistics, Melbourne, July 2018
11. Zhang, C., Li, Q., Song, D.: Aspect-based sentiment classification with aspect-specific graph convolutional networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 4568–4578. Association for Computational Linguistics, Hong Kong, November 2019



Emotion Classification with Explicit and Implicit Syntactic Information

Nan Chen, Qingrong Xia, Xiabing Zhou^(✉), Wenliang Chen, and Min Zhang

School of Computer Science and Technology, Soochow University, Suzhou, China
nchenacs@stu.suda.edu.cn, {zhouxiabing,wlchen,minzhang}@suda.edu.cn

Abstract. Emotion classification has become a hot research topic in natural language processing due to its wide application. Existing studies suffer from the error propagation problem when using the syntax information in emotion classification since the parser can not produce perfect syntax trees. To address this problem, we propose a new approach by comparing and combining different levels of syntactic information to make full use of syntactic information and alleviate the error propagation. First, we propose to use graph convolutional networks (GCN) to encode dependency trees, in which the probability matrix of all dependency arcs (edge-weighted graph) is treated as the GCN adjacent matrix. Next, we extract the dependency parser encoder hidden representations as the implicit syntactic representations, which can directly avoid the error propagation problem. Finally, we fuse the two different syntax-aware information and inject them into our baseline model as extra inputs. Further experimental results show that the explicit and implicit syntactic information can improve the performance of a BERT-based system which is much stronger than the baseline. In addition, we find that the syntactic knowledge that BERT can express is limited, and the syntactic information of our model brings more contributions, which makes our model consistently outperform the BERT on different sentence lengths.

Keywords: Emotion classification · Syntactic information · BERT

1 Introduction

Emotion classification is an essential task in natural language processing (NLP), which aims to detect the emotion labels from the text, such as *joy*, *anticipation*, *trust*, *optimism*, and so on. The advent of social media and its prosperity enables the creation of massive online user-generated content including opinions and product reviews. For example, one of the most popular platforms, Twitter, has reached 192 million daily active users in the third quarter of 2020 [10]. Analyzing such user-generated content and detecting emotions can be widely used in

This work was supported by National Natural Science Foundation of China (Grant No.61936010).

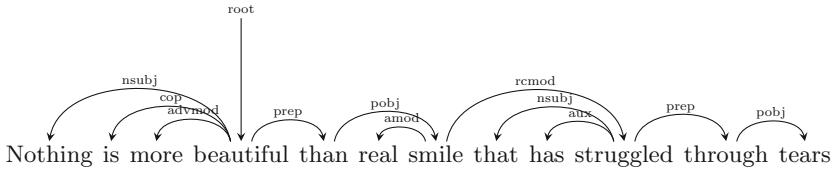


Fig. 1. An example of the full dependency tree.

artificial intelligence customer service [14], social events analysis [20], and even election prediction [25].

In literature, there is a large amount of studies on emotion classification, and both statistic and neural models have been used to predict emotions from the post in social media, such as k-nearest neighbours (KNN) [26], support vector machines [2], long short-term memory (LSTM) [13], attention mechanism [15], transformer [3], and so on. However, most previous studies paid too much attention to improve the model architectures with advanced neural network techniques but ignored the valid syntactic information of texts. Figure 1 shows a syntactic dependency result of an example. The syntactic dependency information identifies subject-verb, adverbial, and other grammatical structures and then analyzes the relationship between different elements in the text, which can help the model understand the text content better and facilitate the emotion classification.

In recent years, some emotion classification studies consider to leverage the syntactic information to help understand the text and detect the emotions. Wang et al. designed a new neural network model by encoding sentences' syntactic dependency trees and document topical information into the document representation [17]. Lai et al. applied a graph convolutional network (GCN) on the preliminary word features from Bi-LSTM according to the syntactic trees [9]. Wang explored a syntax-level self-attention layer to learn a syntax-aware vector for each word [16]. However, most of them only consider the syntactic tree information (1-best dependency trees) directly from the syntactic parser outputs, which may contain some incomplete and inaccurate information. Because the present syntactic parsers are trained under the standard texts, while the texts of Twitter include some informal statements [24], which affects the parsing accuracy in Twitter posts. In addition, using the automatic generated syntactic trees may lead to error propagation problem, which is a common problem in pipeline based methods. Accordingly, we argue that 1) the dependency arc scores of dependency trees from decoder provide confidence of each dependencies that are more valid than the 1-best tree, 2) we can extract encoder output representations from dependency parsers to serve as a kind of implicit syntactic representations, which can avoid the error propagation problem. In this paper, we propose an emotion classification model with both explicit and implicit syntactic information. Specifically, we first extract the hidden layer representations from the encoder of dependency parser model, which is used as implicit syntactic information. Then, we extract the probability matrix of all dependent arcs (edge-

weighted graph) after biaffine scorers from the decoder of dependency parser, and employ GCN to generate the explicit representation. Finally, we incorporate the two types of syntax-aware representations into the emotion classification model as external inputs of the input layer.

Our proposed model not only provides more valid linguistic information, but also alleviates the error propagation problem by using the explicit and implicit syntactic representations. Experimental results on the public SemEval2018 Task1 dataset validate the superiority of our proposed model over the baseline models. Our main contributions of this paper are as follows:

1. Compared to the 1-best tree, our model achieves better results when the GCN encodes explicit syntactic information with the probability matrix of all dependency arcs (edge-weighted graph), containing more structural information.
2. Combining implicit (Imp) and explicit (Exp) syntactic representations brings further improvement compared with using them separately on our base model.
3. The benefit from syntactic information is not entirely overlapped by BERT representations in this task, especially on long-distance sentences.

2 Related Work

The current mainstream approaches of emotion classification are deep neural-network models. Early on, the bidirectional long-short term memory (BiLSTM) network models consider the sequential order between words on emotion analysis [13]. FastText model uses a subword embedding strategy which is much faster than most deep learning models and is comparable in performance to some deep learning models for the emotion classification task [7]. In recent years, considering relevant text sentiment ranking (RERc) is constrained by sentiment relations [27]. Attention mechanism predictive model focuses on sentiment content, incorporating convolutional neural networks fused with transfer learning (NTUA-SLP) [1]. After 2019, there is more progress on emotion classification with the development of large-scale pre-trained language model like Pre-training of Deep Bidirectional Transformers for Language Understanding (BERT) [3]. After that, researchers consider the inclusion of more external information, such as basic information about the individual with personal information for emotion prediction [28], multi-label emotion classification considering latent emotional memories (LEM) [6]. BERT-based graph convolutional network (BERT-GCN) is used to focus on the association between emotion and emotion [19].

The accuracy of syntax can reach over 90% in standard texts with the development of deep neural network techniques [24]. Existing methods effectively integrate syntactic information into the corresponding natural language processing tasks to boost the performance. Xia et al. use a multi-task learning approach for dependency parsing and semantic role labeling that obtains good performance [18]. Duan et al. use syntax-awareness for data augmentation to improve the performance of machine translation [5]. Zhang et al. use the syntactic model to improve the performance of opinion role labeling [23]. Of course, there are

various syntax-aware works in emotion classification, such as considering combining syntactic tree information with self-attention [16] and combining syntactic tree information with graph convolutional network [9].

Different from previous works, we add different level of syntactic information (implicit and explicit syntactic information) into the fine-tuned BERT-based baseline model to alleviate the error propagation problem and make full use of the syntactic information to boost the performance of model.

3 Proposed Framework

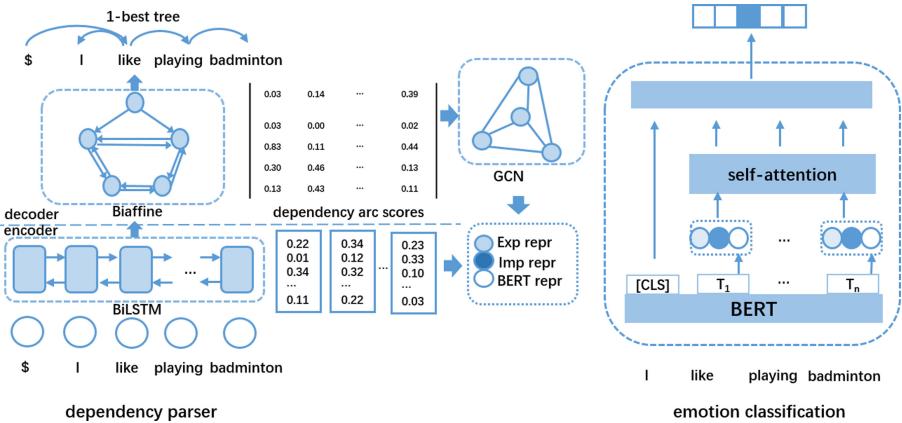


Fig. 2. Overview of our model. The left is the dependency parser model, and the right is the emotion classification model (BERT). Our model concatenates the explicit and implicit representations with the basic emotion classification model.

As shown in Fig. 2, our model framework consists of two models, a sentiment classification model and a dependency parsing model. Overall, given one sentence, we first feed it into the dependency parsing model to get the explicit and implicit syntactic representations. Then, we combine the two different representations and inject them into our basic emotion classification model as extra inputs. In the following, we will introduce the baseline model, the employed dependency parser, the explicit and implicit syntactic representations one by one.

3.1 Baseline Model

In general, we denote a sentence with n words $S = \{W_1, W_2, \dots, W_n\}$. We choose the emotion classifier with pre-trained language model (BERT-base)¹ as the

¹ www.huggingface.co.

baseline model. To fit the BERT model, the data are preprocessed to obtain token embeddings, segment embeddings, and position embeddings. Following the common practice, the first token of each sequence is treated as the special classification embedding ([CLS]). After the BERT model, we get the output representation, denoted as $\{t_{[CLS]}, t_1, t_2, \dots, t_n\}$. Please note that since BERT is based on sub-word operations. For example, the word “smartest” becomes [“smart”, “##est”] after the BERT sub-word operation. This paper takes the first sub-word representation (“smart”) representing the whole word (“smartest”).

The sentence representation is denoted as $T = \{t_1, t_2, \dots, t_n\}$. Next, the self-attention is used to obtain attention for the context words to enhance the role of emotion-specific words. Formally, the self-attention uses word representations to construct the weight for each word in one sample:

$$v_i = \tanh(Wt_i + b) \quad (1)$$

$$a_i = \frac{\exp(v_i)}{\sum_{i=1}^n \exp(v_i)} \quad (2)$$

$$V = \sum_{i=1}^n a_i t_i \quad (3)$$

W_i is the weight matrix, b_i is the bias, t_i is the input vector of the BERT output for word W_i . a_i denotes the importance of the i -th word of the current text, where $\sum_{i=1}^n a_i = 1$ and n represents the total number of words. The output of the attention V is combined with [CLS] and then fed into one middle hidden layer (a fully connected linear layer). Finally, the classifier passes the middle hidden representation into last fully connected linear layer with label number output size (number of emotion labels). We use *Sigmoid* function over the final output with a threshold to predict the presence/absence of one emotion label of the emotion categories and use a cross-entropy loss function.

The right part of Fig. 2 shows the overall workflow of our baseline model except for *Exp repr* and *Imp repr*.

3.2 Dependency Parser

As shown in the left part of Fig. 2, we use the state-of-the-art biaffine parser [4] as the dependency parser module. The dependency parser consists of a multi-layer BiLSTM encoder and a biaffine scorer decoder. The BiLSTM encoder is used to encode the input sentence to get the contextualized word representations. Given the encoder output, the dependency scores of each word pair ($w_i \rightarrow w_j$) is computed with the biaffine scorer. With the scores of all dependencies, we use the maximum spanning tree (MST) algorithm to obtain the highest-score unlabelled dependency tree, and then independently decide the syntactic label for each dependency arc. For more details, please refer to the original paper of Dozat and Manning (2016) [4].

We argue that the parser encoder learnt implicit syntactic information and the dependency score matrix provides more valid information than the 1-best tree, which we will discuss in the following sections.

3.3 Explicit Method

As a classical way to utilize syntactic information, previous works usually use syntactic parser to parse sentences into syntactic trees, and then use various methods to explicitly encode the trees to get specific representations for target tasks [9, 16]. However, this kind of methods usually have error propagation problem, because the employed syntactic parser cannot generate fully correct syntactic trees. To alleviate this problem, we propose an enhanced syntax-aware graph convolutional networks (SYNGCN) that uses the dependency arc score matrix as the adjacent matrix. On the one hand, using the dependency scores as the adjacent matrix of the GCN module allows all connections for every nodes that alleviates the error propagation problem of using automatic dependency tree. On the other hand, the dependency score matrix provides valid dependency attention for every word pair in one sentence, which conveys richer structural information than the traditional 1-best dependency tree.

As show in Fig. 2, we first calculate the basic GCN [8]. We represent a graph as $G = (V, E)$, where V and E are the nodes and edges, respectively. The i -th node of layer l is computed as follows:

$$h_i^l = \rho\left(\sum_{j=1}^n A_{ij} W^l h_j^{l-1} + b^l\right) \quad (4)$$

where A is the adjacency matrix, which is the dependency arc score matrix (“dependency arc scores” in Fig. 2) that summed a diagonal matrix. W and b are the model parameters, and ρ is the activation function. Specifically, h_0 is the initial input word vector that are generated by the word2vec² tool. The output of SYNGCN is fed into the baseline model as extra inputs.

3.4 Implicit Method

Recently, the very popular pre-trained language models (e.g. ELMo [12] and BERT [3]) are trained from large-scale training corpus. These large-scale training corpus trained language models have gained much attention, which produce strong implicit representations, boosting many NLP tasks. Inspired by pre-trained language models and previous works [18, 22], we try to migrate the implicit syntactic representations to our sentiment classification model to verify the effectiveness. Different from the multi-task strategy [18], we extract the implicit syntactic representation as follows:

$$\begin{aligned} \vec{h}_t &= \text{LSTM}(h_{t-1}, x_t) \\ \overleftarrow{h}_t &= \text{LSTM}(h_{t+1}, x_t) \\ h_t &= W_{\vec{h}} \vec{h}_t + W_{\overleftarrow{h}} \overleftarrow{h}_t + b \end{aligned} \quad (5)$$

² <https://code.google.com/archive/p/word2vec/>.

\overrightarrow{h}_t and \overleftarrow{h}_t are the t -th step left-to-right LSTM output and right-to-left LSTM output in the top encoder layer of the dependency parser. h_t is the extracted implicit syntactic representation (*Imp repr* in Fig. 2). The left bottom part of Fig. 2 shows the process.

3.5 Fusion of Explicit and Implicit Information

The explicit syntactic representations encode the dependency structure information and the implicit syntactic representations convey the structural information. We hypothesize that the two syntactic representations are different and somewhat complementary. Therefore, we fuse the two information at the input layer in our final model, which is denoted as:

$$E_i = Exp_i \oplus Imp_i \oplus BERT_i. \quad (6)$$

Figure 2 shows the overall workflow of our proposed framework.

4 Experiments

4.1 Settings

Dataset. We conduct experiments on the public SemEval 2018 Twitter English dataset [11]. SemEval contains 11 emotion labels, namely: *anger*, *anticipation*, *disgust*, *fear*, *happiness*, *love*, *optimism*, *pessimism*, *grief*, *surprise*, and *trust*. Each sentence may contain more than one emotion label. The total number of sentences is 10983, including 6,838 training sentences, 886 validation sentences, and 3,259 test sentences. Besides, the total number of words is 32,557, of which 3,419 sentences have more than three emotion labels, 4,442 sentences have two emotion labels, and 1,563 sentences have one emotion label.

Dependency Parsing. Following the standard training approach for syntactic parsing, biaffine parser is used to train the dependency parsing according to the Stanford Parser V3.0 with an unmarked dependency success rate (UAS) 95% on the English dataset Penn Treebank data (PTB). In this paper, the biaffine parser-trained model is used to predict SemEval data to obtain the 1-best syntactic tree, the explicit and the implicit syntax information of the syntactic state. For the other settings, following the work of Dozat [4].

Evaluation. For comparison with previous works, we conduct the experiments and report the macro F1-score.

Hyper-parameters. For the emotion classification module, we follow most hyper-parameter settings of Devlin [3]. The Adam optimizer trains our models with a learning rate of 3e-5 and weight decay of 1e-8. We use the dropout rate of 0.5 and the batch size is set to 64. For the parameters of the dependency parser, we mostly follow Dazat [4]. The hidden size of SYNGCN is 150 and the hidden size of BiLSTM is 800. The input of SYNGCN model is 300-dimensional word embeddings. The threshold of the final output is set to 0.5. The dimension of middle linear layer is 256. The training process is early stopped if the peak performance on the development data does not increase in 10 consecutive epochs.

4.2 Base Models

We compare our model with the following models.

BiLSTM [13] is used to encode contextualized information of sentences for predicting emotions.

FastText [7] adopts sub-word embedding strategy. FastText is often on par with deep learning classifiers in terms of accuracy.

RERc [27] utilizes relevant sentiment ranking in texts subject to sentiment relations.

NTUA-SLP [1] predicts affective content in tweets with deep attentive Recurrent Neural Networks and transfer learning.

BERT-DK [21] devises a simple method to obtain domain knowledge and further propose a method to integrate domain knowledge with general knowledge based on deep language models to improve performance of emotion classification.

BERT-GCN [19] captures the dependencies among different emotions through graph networks. These graphs are constructed by leveraging the co-occurrence statistics among different emotion categories.

LEM [6] considers prior emotion distribution in a sentence and effectively captures the context information closely related to the corresponding emotion.

Baseline [3] chooses the emotion classifier with pre-trained language model (BERT-base) as the baseline model. The details are described in the base model section.

Baseline + Exp^{1-best} + Imp is the same model as our proposed model (imp+exp). The only difference is that the adjacency matrix of the GCN is constructed in the same way as proposed by Lai et al.(2020) [9], which aims to verify the difference in performance between our model and the 1-best syntactic tree GCN.

4.3 Main Results

Table 1 shows the results of our proposed framework and comparison with previous works. First, we can see that both the explicit and implicit methods outperforms our strong BERT-based baseline model. Second, our final model that combines the explicit and implicit syntactic information achieves the best reported result of 0.571 F1 score. Finally, we can see that using the 0-1 adjacent matrix

Table 1. Experimental results and comparison with previous works on SemEval (2018) test data.

Models	F1
BiLSTM	0.427
FastText	0.438
RERc (2018) [27]	0.539
NTUA-SLP (2018) [1]	0.528
BERT-DK (2019) [21]	0.549
BERT-GCN (2020) [19]	0.563
LEM (2020) [6]	0.567
Baseline	0.550
Baseline + Exp	0.559
Baseline + Imp	0.556
Baseline + Exp^{1-best} + Imp	0.560
Baseline + Exp + Imp	0.571

from the 1-best tree in the explicit method hurt the performance, which only achieves 0.560 F1 score, illustrating the effectiveness of using the dependency arc scores matrix as the adjacent matrix. Furthermore, we list the detailed F1 scores regrading to different emotions in Table 2. We can observe that our syntax-aware framework outperforms our baseline model in most emotions.

Table 2. Detailed F1 scores of each emotion on SemEval2018.

Emotion	Baseline	Baseline+Exp	Baseline+Imp	Baseline+Exp+Imp
Fear	0.716	0.721	0.731	0.742
Anticipation	0.316	0.303	0.295	0.319
Disgust	0.738	0.718	0.725	0.719
Anger	0.775	0.760	0.767	0.763
Joy	0.840	0.837	0.836	0.838
Love	0.546	0.594	0.580	0.602
Optimism	0.688	0.722	0.713	0.725
Pessimism	0.300	0.351	0.322	0.377
Sadness	0.681	0.678	0.686	0.686
Surprise	0.294	0.275	0.301	0.282
Trust	0.160	0.194	0.153	0.227
F1	0.550	0.559	0.556	0.571

4.4 The Influence of Sentence Length

As shown in Fig. 3, we investigate the performance of different models at different sentence lengths. We can find that our proposed method consistently outperforms both BiLSTM and BERT, especially in the range of 35–45, which we think because the syntactic trees effectively capture long-range dependencies. Further, we can find that our syntax-aware framework outperforms than the BERT-based baseline model, indicating that the contribution of syntactic information is not entirely replaced by BERT representations.

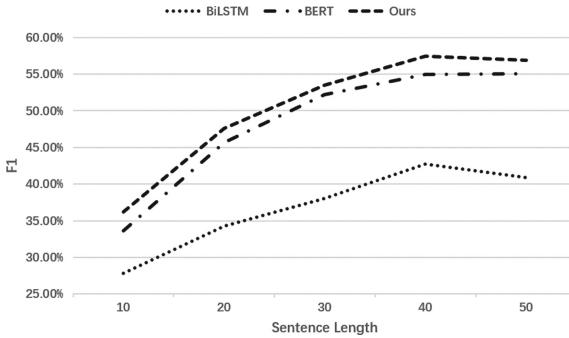


Fig. 3. The F1 values of BiLSTM, BERT and ours against the sentences with different lengths on the SemEval2018.

Table 3. The examples of case study.

Post	Baseline	Ours	Gold
<i>Nothing is more beautiful than real smile that has struggled through tears</i>	<i>Joy optimism</i>	<i>Joy optimism trust</i>	<i>Joy optimism trust</i>
<i>Life is too short to be jealous, hating, keeping up mess, and worrying about things that do not concern you</i>	<i>None</i>	<i>Optimism</i>	<i>Optimism</i>
<i>When you are with your friend and you are still laughing @Tereza_Gray #laugh</i>	<i>Joy</i>	<i>Joy love</i>	<i>Joy love</i>

4.5 Case Study

To better understand the usefulness of syntactic information. We give a case study of using baseline and our proposed syntax-aware framework from SemEval test set in Table 3. From the results of models, we presume that our baseline does not capture enough abundant syntactic information, resulting in bias in

recognition. In detail, for example, in the sentence *Nothing is more beautiful than real smile that has struggled through tears.*, the results of the baseline model are *joy* and *optimism*. As show in Fig. 1, we can see that the syntactic structure gives an implicit meaning of *trust* by the phrase *Nothing is more beautiful than real smile*, so our syntax-aware framework correctly predict it.

5 Conclusion

We present a syntax-aware emotion classification approach that utilizes both explicit and implicit syntactic information, where the explicit information refers to syntactic structure information and the implicit information refers to the representations extracted from the encoder of a dependency parser. Experimental results and detailed analyses demonstrate that our approach effectively captures syntactic information and successfully integrated into BERT-based model. In the future, we will explore more useful external expertise and combine it with the emotion classification model to improve the performance.

References

1. Baziotis, C., et al.: NTUA-SLP at semeval-2018 task 1: predicting affective content in tweets with deep attentive rnns and transfer learning. In: Proceedings of SemEval@NAACL-HLT, pp. 245–255 (2018)
2. Chandra, M.A., Bedi, S.S.: Benchmarking tree-based least squares twin support vector machine classifiers. Int. J. Bus. Intell. Data Min. **16**(3), 381–395 (2020)
3. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT (1), pp. 4171–4186 (2019)
4. Dozat, T., Manning, C.D.: Deep biaffine attention for neural dependency parsing. In: Proceedings of SIGIR (2016)
5. Duan, S., Zhao, H., Zhang, D., Wang, R.: Syntax-aware data augmentation for neural machine translation. CoRR (2020)
6. Fei, H., Zhang, Y., Ren, Y., Ji, D.: Latent emotion memory for multi-label emotion classification. In: Proceedings of AAAI, pp. 7692–7699 (2020)
7. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: Proceedings of EACL, pp. 427–431 (2017)
8. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: Proceedings of ICLR (2017)
9. Lai, Y., Zhang, L., Han, D., Zhou, R., Wang, G.: Fine-grained emotion classification of chinese microblogs based on graph convolution networks. In: Proceedings of WWW, pp. 2771–2787 (2020)
10. Liu, R.: The number of twitter users has accelerated. Website (2021). <https://finance.sina.com.cn/tech/2021-02-10/doc-ikftpnnny6189670.shtml>
11. Mohammad, S., Bravo-Marquez, F., Salameh, M., Kiritchenko, S.: Semeval-2018 task 1: affect in tweets. In: Proceedings of SemEval@NAACL-HLT, pp. 1–17 (2018)
12. Peters, M.E., et al.: Deep contextualized word representations. In: Proceedings of NAACL-HLT, pp. 2227–2237 (2018)

13. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. *IEEE Trans. Sig. Process.* **11**, 2673–2681 (1997)
14. Shuangyong, S., Chao, W., Chenglong, C., Wei, Z., Haiqing, C.: Sentiment analysis for intelligent customer service chatbots. *J. Chin. Inf. Process.* **2**, 80–95 (2020)
15. Vaswani, A., et al.: Attention is all you need. In: *Proceedings of NIPS*, pp. 5998–6008 (2017)
16. Wang, C., Wang, B.: Encoding sentences with a syntax-aware self-attention neural network for emotion distribution prediction. In: *Proceedings of NLPCC* (2), vol. 12431, pp. 256–266 (2020)
17. Wang, C., Wang, B., Xiang, W., Xu, M.: Encoding syntactic dependency and topical information for social emotion classification. In: *Proceedings of SIGIR*, pp. 881–884 (2019)
18. Xia, Q., Li, Z., Zhang, M.: A syntax-aware multi-task learning framework for chinese semantic role labeling. In: *EMNLP/IJCNLP* (1), pp. 5381–5391 (2019)
19. Xu, P., Liu, Z., Winata, G.I., Lin, Z., Fung, P.: Emograph: capturing emotion correlations using graph networks. *CoRR* (2020)
20. Yang, Q., et al.: Senwave: monitoring the global sentiments under the COVID-19 pandemic. *CoRR* (2020)
21. Ying, W., Xiang, R., Lu, Q.: Improving multi-label emotion classification by integrating both general and domain-specific knowledge. In: *Proceedings of W-NUT@EMNLP*, pp. 316–321 (2019)
22. Yu, N., Zhang, M., Fu, G.: Transition-based neural rst parsing with implicit syntax features. In: *Proceedings of COLING*, pp. 559–570 (2018)
23. Zhang, B., Zhang, Y., Wang, R., Li, Z., Zhang, M.: Syntax-aware opinion role labeling with dependency graph convolutional networks. In: *Proceedings of ACL*, pp. 3249–3258 (2020)
24. Zhang, Y., Li, Z., Zhang, M.: Efficient second-order treecrf for neural dependency parsing. In: *Proceedings of ACL*, pp. 3295–3305 (2020)
25. Zhao, J., Liu, K., Xu, L.: Sentiment analysis: mining opinions, sentiments, and emotions. *Comput. Linguist.* **3**, 595–598 (2016)
26. Zheng, R., Zhang, S., Liu, L., Luo, Y., Sun, M.: Uncertainty in bayesian deep label distribution learning. *Appl. Soft Comput.* **101**, 107046 (2021)
27. Zhou, D., Yang, Y., He, Y.: Relevant emotion ranking from text constrained with emotion relationships. In: *Proceedings of NAACL-HLT*, pp. 561–571 (2018)
28. Zhou, X., Wang, Z., Li, S., Zhou, G., Zhang, M.: Emotion detection with neural personal discrimination. In: *Proceedings of EMNLP/IJCNLP*, pp. 5498–5506 (2019)



MUMOR: A Multimodal Dataset for Humor Detection in Conversations

Jiaming Wu, Hongfei Lin, Liang Yang^(✉), and Bo Xu

Dalian University of Technology, Dalian 116024, China
peibost@mail.dlut.edu.cn, {hflin,liang,xubo}@dlut.edu.cn

Abstract. Humor detection attracts increased attention in natural language processing for its potential applications. Prior work focus on analyzing humor on isolated, textual data, but humor usually comes from the interaction among speakers in a multimodal way. In this paper, we proposed a novel dataset named MUMOR, which consists of multimodal dialogues in both English and Chinese. It contains a total of 29,585 utterances belonging to 1,298 dialogues from two TV-sitcoms. We manually annotated each utterance with humor, emotion, and sentiment labels. To our best knowledge, this is the first corpus containing Chinese conversations for humor detection. This dataset could be used for research on humor detection, humor generation, and multi-task learning on emotion and humor analysis. We released this dataset publicly.

Keywords: Multimodal · Sentiment analysis · Humor detection

1 Introduction

Humor plays an important role in human communication. It not only creates an entertaining atmosphere, but also helps regulate conversations, reduce stress, and build trust between partners [1]. Humor recognition attracts increased attention for its potential application in human-computer interaction, which can be used in advertising, healthcare and education area [2].

Recent years, there are many papers about construction of humor detection dataset. Mihalcea and Strapparava [3] introduced a dataset containing 16,000 humorous and 16,000 non-humorous text in English. The humorous data are one-liners collected from Internet, and the non-humorous sentences were collected from one-liners, Reuters titles, BNC sentences, and proverbs. Zhang and Liu [4] constructed an English tweets dataset for recognizing humor on Twitter. The corpus contains 3,000 tweets with annotation of 3 categories: humorous tweets, non-humorous tweets, and humorous non-tweets. Castro et al. [5] established a Spanish corpus for humor detection. They collected 39,363 tweets from both humorous accounts and non-humorous accounts. After filtering and manually labeling, they finally got a corpus containing 33,531 Spanish tweets with humorous and non-humorous labels. Khandelwal et al. [6] proposed a corpus with 3,543 English-Hindi code-mixed tweets. Each tweet is annotated with humorous or non-humorous label. Blinov et al. [7] constructed a large size dataset for humor

recognition in Russian. They collected jokes and funny dialogues from various online resources. This dataset contains more than 300,000 short tests, which is significantly larger than any previous humor-related corpus.

Most of the previous work in the field of humor detection focusing on isolated, textual data, few works have been devoted to detecting humor in conversations. Bertero and Fung [8] proposed a LSTM based network to detect humor in dialogues. They constructed a corpus with 43,672 utterances from 1,589 scenes. The corpus is collected from the subtitles and scripts of the TV-sitcom “*The Big Bang Theory*”. In their later work [9], they combined acoustic and language features to detect humor in conversations. Experiment results on three English sitcom corpus showed that combining the acoustic features brought improvement on humor detection.

It is important and difficult to detect humor in conversations. The formation of humor has a setup process. Sometimes one sentence itself is not humorous, but it becomes humorous when combined with the context. And the interaction between speakers in a dialogue often produces humorous effect. In addition, multimodal information also helps detect humor. Sometimes the reason makes an utterance funny is the vocal tonality, facial expressions, and body gesture of the speaker but not the meaning of the utterance text. And a multimodal dialogue scene is a common scenario in reality. Detecting humor in multimodal dialogue is a very challenging task. It requires the model to obtain the contextual information and integrate the features of different modalities.

Figure 1 shows an example of humor in multimodal dialogue. The topic of this dialogue is quitting smoke. Each utterance in this conversation, if being treated separately, is not humorous. However, considering the contextual information the emotional changes of the characters in the dialogue, utterance 3, 4, 6 become humorous.

In this work, we constructed *Multimodal Utterance-level Humor Dataset* (MUMOR), a dataset for humor detection in multimodal conversations. It contains two language corpora: English and Chinese. Both corpora contain textual dialogues with their corresponding video and audio segments, which means each utterance in this dataset has three modal sources. MUMOR provides humor label for recognizing humor in dialogues. Furthermore, each utterance is annotated with emotion and sentiment labels, which can be used in multi-task learning on emotion and humor analysis as research has shown that modeling sentiment is effective for humor detection [10]. Our contributions are as follows.

- We proposed a dataset, MUMOR, contains both English and Chinese corpus. To our knowledge, this is the first Chinese conversational corpus for humor detection.
- We introduced the data processing and labeling process of this dataset, which has reference significance for the construction of multi-modal data set. This paper shows the data distribution and statistical information of MUMOR.
- MUMOR provides audio, visual, and textual modal sources. It is a multilingual, multi-label dialogue dataset. It can be used for multi-modal sentiment analysis, humor recognition and dialogue generation research.



Fig. 1. A example of humor in multimodal dialogue.

2 Dataset

2.1 Data Source

The MUMOR dataset contains English and Chinese corpus which we name as MUMOR-EN and MUMOR-ZH, respectively.

MUMOR-EN is constructed based on the MELD dataset [11]. MELD is a multimodal dataset used for emotion recognition task. It contains 1,433 dialogues from TV-sitcom “*Friends*”. Each dialogue contains several utterances belonging to the same scene. And each utterance encompasses audio, visual and textual modalities. Since the purpose of our dataset is to recognize humor in long conversations rather than short texts, we discarded the conversations with a small number of utterance in the original dataset. We filtered out dialogues with less than 3 utterances and made humor annotation on the rest data.

For MUMOR-ZH, we choose a popular Chinese sitcom “我爱我家” (I Love My Family) as our data source. We collected 81 episodes of this TV series video and extracted utterance text and its timestamps from subtitle files. We cut the video into clips according to the timestamps of each utterance. The utterances are grouped into dialogues following the constraint that all the utterances in a dialogue comes from the same episode and scene. Finally, we got 19,103 utterances belonging to 519 dialogues.

2.2 Data Format

Each utterance is identified by a dialogue ID and an utterance ID, which also name the corresponded video clip file saved in *.mp4* format.

In Table 1, we show the format of our dataset, which contains the information of the utterance, the speaker, the humor label, the emotion label, the sentiment label, the dialogue ID and the utterance ID.

Table 1. Dataset format.

Utterance	Speaker	Humor	Emotion	Sentiment	D_ID	U_ID
All right, there you go!	Ross	Non-humorous	Joy	Positive	439	12
Yeah, you hang in there Teddy!	Joey	Non-humorous	Anger	Negative	439	13
I'm Andrew, and I didn't pay for this pear.	Older Scientist	Humorous	Neutral	Neutral	439	14
Okay, good-good for you.	Ross	Non-humorous	joy	Positive	439	15
I'm Rhonda, and these aren't real!	Tour Guide	Humorous	Neutral	Neutral	439	16

2.3 Data Annotation

The MUMOR dataset contains humor, emotion, and sentiment labels for each utterance. We ask three annotators to watch the video clips with subtitles of each utterance, and let them decide whether this utterance is humorous or not and which kind of emotion it belongs to.

The annotators are Chinese postgraduate students with at least 10 years of English learning experience. In addition, we displayed both English and translated Chinese subtitles while annotating English data. Before the formal annotation, all annotators did some pre-annotation test to ensure the quality of the annotation.

For humor label, we provide two categories: *humorous* and *non-humorous*. The overall Fleiss' kappa score of humor annotation process is 0.81, which indicates a substantial agreement among annotators.

For emotion label, We keep the original emotion labels in MELD dataset for MUMOR-EN, and make our emotion annotation by 3 annotators on MUMOR-ZH. The emotion label contains six universal emotions *Joy*, *Sadness*, *Fear*, *Anger*, *Surprise*, and *Disgust* [12] in addition with *Neutral*. For utterances that 3 annotators can not reach agreement, we label it with a *None* label. There exists 410 *None* in totally 19,103 utterances. The Fleiss' kappa score of emotion task is 0.45 (kappa of MELD emotion annotation process is 0.43).

For sentiment label, we apply the scheme proposed by Poria et al. [11]. It considered *Anger*, *Disgust*, *Fear*, *Sadness* as *Negative*, *Joy* as *Positive* and made further annotation in class *Surprise* to decide whether is *Positive* or *Negative*. Our 3 annotators labeled the utterances in MUMOR with the tactic mentioned above. The Fleiss' kappa score of sentiment annotation process is 0.84.

3 Dataset Analysis

First, we display the main statistical information of the two language data in Table 2.

It can be seen that the scale of the Chinese data is about twice that of the English data, showing a roughly 2:1 ratio between the total duration of the video and the total number of utterance. The length of an utterance is the

Table 2. Data statistics.

	MUMOR-EN	MUMOR-ZH
Total video duration (h)	9.03	18.12
Avg duration of utterance (s)	3.11	3.42
# Dialogues	779	519
# Utterances	10,482	19,103
D-length	13.46	36.81
U-length	10.37	10.04
Humorous percentage (%)	24.59	28.36
# Speakers	259	91
Total number of words	108746	191770
Number of unique words	6869	17804

number of words in it, this number on the Chinese and English data is very close, being 10.04 and 10.37 respectively. The average duration of one utterance is also relatively close, 3.42 s and 3.11 s respectively. It can be seen that the narrative length and speech speed of the actors in the two sitcoms are relatively close. The big difference between the two languages is the length of the conversation. Among them, the value of Chinese data is 36.81, which is close to three times of 13.64 on English data. It indicates that a paragraph in the Chinese data has a longer sequence and contains more context information. In addition, the proportion of humor in the data of different languages is similar, and there is not much difference. The percentage of humor in the Chinese data is 28.36%, and the median value of the English data is slightly lower, at 24.58%, and the ratio of positive and negative cases is about 3:1. The Chinese data contains a total of 191,770 words, of which the size of the non-repeated vocabulary is 17,804. In contrast, there are only 6,869 different words in the English data. It can be seen that the vocabulary variety of the Chinese data is higher than that of the English data.

We split our dataset into training, development, and testing set on two corpus, respectively. Table 3 shows the data statistics on the 3 sets. It can be seen that the main statistical information on the training, development, and testing set is very close.

Figure 2 shows the distribution of humor percentage in one dialogue on two corpora. There are 9 Chinese dialogues that do not contain humorous utterances, while the number is 20 in English corpus. The proportion of humorous utterances in most conversations is 10% to 40%. While the 20%-30% range had the largest number of dialogues, with 252 dialogues in the Chinese corpus and 192 dialogues in the English corpus.

We also calculate the utterance proportion for all sitcom characters. For those with less than 2% utterances, we group them as *Other*. The result is shown at Fig. 3. We can see both corpus contain 6 main characters, and the utterance proportion of main characters in the English corpus is more balanced.

Table 3. Dataset division.

	MUMOR-EN			MUMOR-ZH		
	Train	Dev	Test	Train	Dev	Test
# Dialogues	551	70	158	348	43	128
# Utterances	7,472	914	2,096	12,677	1,632	4,794
D-length	13.56	13.06	13.27	36.43	37.95	37.45
U-length	10.34	10.20	10.59	9.99	10.49	10.02
Humorous percentage %	23.73	27.02	26.62	28.76	26.23	28.04
# Speakers	215	36	75	76	24	51

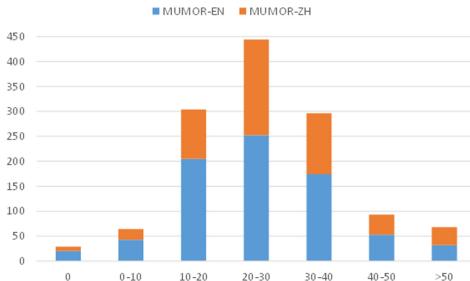
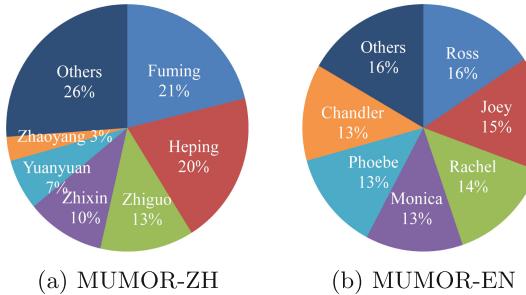
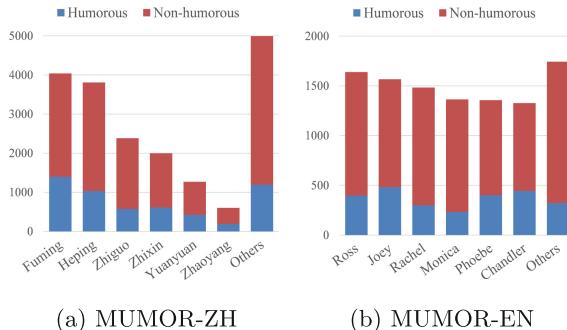
**Fig. 2.** Humor distribution.

Figure 4 shows the humorous percentage of each character in two corpora, respectively. We can see some characters with lower utterance proportion but have higher humor percentage, they played an important role in pleasing the audience, like *Yuanyuan* and *Chandler* in their respective sitcoms. Both of them have a humorous percentage of over 33%.

4 Comparison with Existing Dataset

In this section, we will compare MUMOR dataset with two related multimodal datasets and introduce the potential applications of our dataset.

UR-FUNNY [13] is a multimodal dataset for humor detection. It contains 16,514 speech data extracted from TED speech. Each speech segment contains several utterances and labeled with humours or non-humours label. The positive instances end with a punchline utterance, the negative instances sampled from sentences in the same distribution but not end with a punchline. UR-FUNNY dataset is making classification on dialogue level while MUMOR dataset works on utterance level. From the example in Fig. 1, we can see that compared with the punchline in the speech, the humor in the dialogue does not only appear in the ending utterance, but is distributed in the whole dialogue. Furthermore, the average length of context in UR-FUNNY dataset is 2.86 which is much shorter

**Fig. 3.** Character distribution.**Fig. 4.** Humor distribution.

than that in MUMOR dataset, which means models need powerful context modeling capabilities to achieve good results on MUMOR dataset.

MUStARD [14] is a multimodal dataset for sarcasm detection. Its data are extracted from three English sitcoms. Sarcasm utterance in this dataset is accompanied with several historical utterances as its context which is much shorter than the length of context in MUMOR dataset. Unlike MUStARD dataset, MUMOR dataset focus on humor detection. Although there is a strong relationship between sarcasm and humor, humor does not only come from sarcasm. Table 4 shows the comparison between MUMOR dataset and the existing dataset.

Table 4. Comparison with existing dataset.

	MUMOR	UR-FUNNY	MUStARD
Number of videos	1298	16514	690
Avg duration of utterance(s)	3.31	4.64	5.22
Annotation granularity	Utterance-level	Dialogue-level	Dialogue-level
Labels	Humor, sentiment, emotion	Humor	Sarcasm
Languages	English & Chinese	English	English

MUMOR dataset can be used in the research of detecting humor in conversations involving multiple speakers. It can also be used to study humor differences in multiple languages as it provides corpus in both Chinese and English. In addition, the emotion and sentiment labels can be used to analysis the relationship between emotion and humor through multi-task learning.

5 Conclusion and Future Work

In this work, we constructed MUMOR, a multimodal dialogue dataset. It provides two language corpus: English and Chinese. It totally contains 29,585 utterances from 1,298 dialogues from two sitcoms. Each utterance in MUMOR has textual, audio, and visual modal sources. We introduced the process of building this dataset and the kappa score indicated a high quality of our dataset.

Our dataset provided emotion, sentiment and humor label. Moreover, it can be used for emotion recognition, humor response generation, and multi-task learning on emotion and humor analysis. In addition, research about multimodal feature extraction and fusion can be explored on our dataset.

References

1. Morse, D.: Use of humor to reduce stress and pain and enhance healing in the dental setting. *J. N.J. Dent. Assoc.* **78**(4), 32–36 (2007)
2. Nijholt, A., Niculescu, A.I., Alessandro, V., Banchs, R.E.: Humor in human-computer interaction: a short survey (2017)
3. Mihalcea, R., Strapparava, C.: Making computers laugh: Investigations in automatic humor recognition. In: HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, Vancouver, Canada, 6–8 October 2005, pp. 531–538 (2005). <https://www.aclweb.org/anthology/H05-1067/>
4. Zhang, R., Liu, N.: Recognizing humor on twitter. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, 3–7, November 2014, pp. 889–898 (2014). <https://doi.org/10.1145/2661829.2661997>, <https://doi.org/10.1145/2661829.2661997>
5. Castro, S., Cubero, M., Garat, D., Moncecchi, G.: Is this a joke? Detecting humor in spanish tweets. In: Montes-y-Gómez, M., Escalante, H.J., Segura, A., Murillo, J.D. (eds.) IBERAMIA 2016. LNCS (LNAI), vol. 10022, pp. 139–150. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47955-2_12
6. Khandelwal, A., Swami, S., Akhtar, S.S., Shrivastava, M.: Humor detection in english-hindi code-mixed social media content : corpus and baseline system. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, 7–12, May 2018 (2018). <http://www.lrec-conf.org/proceedings/lrec2018/summaries/363.html>
7. Blinov, V., Bolotova-Baranova, V., Braslavski, P.: Large dataset and language model fun-tuning for humor recognition. In: Korhonen, A., Traum, D.R., Màrquez, L. (eds.) Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Long Papers, vol. 1, pp. 4027–4032. Association for Computational Linguistics (2019). <https://doi.org/10.18653/v1/p19-1394>

8. Bertero, D., Fung, P.: A long short-term memory framework for predicting humor in dialogues. In: NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, 12–17, June 2016, pp. 130–135 (2016). <https://www.aclweb.org/anthology/N16-1016/>
9. Bertero, D., Fung, P.: Multimodal deep neural nets for detecting humor in TV sitcoms. In: 2016 IEEE Spoken Language Technology Workshop, SLT 2016, San Diego, CA, USA, 13–16, December 2016, pp. 383–390 (2016). <https://doi.org/10.1109/SLT.2016.7846293>
10. Liu, L., Zhang, D., Song, W.: Modeling sentiment association in discourse for humor recognition. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, 15–20, July 2018, Short Papers, vol. 2, pp. 586–591 (2018). <https://doi.org/10.18653/v1/P18-2093>, <https://www.aclweb.org/anthology/P18-2093/>
11. Poria, S., Hazarika, D., Majumder, N., Naik, G., Cambria, E., Mihalcea, R.: MELD: a multimodal multi-party dataset for emotion recognition in conversations. In: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Long Papers, vol. 1, pp. 527–536 (2019). <https://www.aclweb.org/anthology/P19-1050/>
12. Ekman, P., Friesen, W.V., O’sullivan, M., Chan, A., Diacoyanni-Tarlatzis, I., Heider, K., Krause, R., LeCompte, W.A., Pitcairn, T., Ricci-Bitti, P.E., et al.: Universals and cultural differences in the judgments of facial expressions of emotion. *J. Pers. Soc. Psychol.* **53**(4), 712 (1987)
13. Hasan, M.K., et al.: UR-FUNNY: a multimodal language dataset for understanding humor. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, 3–7, November 2019, pp. 2046–2056. Association for Computational Linguistics (2019). <https://doi.org/10.18653/v1/D19-1211>
14. Castro, S., Hazarika, D., Pérez-Rosas, V., Zimmermann, R., Mihalcea, R., Poria, S.: Towards multimodal sarcasm detection (an *_obviously_* perfect paper). In: Korhonen, A., Traum, D.R., Màrquez, L. (eds.) Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28–August 2, 2019, Long Papers, vol. 1, pp. 4619–4629. Association for Computational Linguistics (2019). <https://doi.org/10.18653/v1/p19-1455>

NLP Applications and Text Mining



Advertisement Extraction from Content Marketing Articles via Segment-Aware Sentence Classification

Xiaoming Fan and Chenxu Wang^(✉)

Xi'an Jiaotong University, Xianning West Road 28, Xi'an, China
cxwang@mail.xjtu.edu.cn

Abstract. The rapid development of social media has brought the prosperity of online economy. Recently, product promotion in social networks has become an essential way of online marketing. As one of the most common marketing means, Content Marketing (CM) inserts advertisements into regular articles in a roundabout and covert way. However, the values and characteristics of products are often exaggerated to attract users' attention. It could cause severe economic losses to users and influence the creditworthiness of the platforms. In this paper, we model the problem of advertisement extraction from CM articles as a sentence classification task. We propose a topic-enhanced deep neural network to encode the semantic information of a sentence for classification. Motivated by the characteristics of CM articles, we develop a segment-aware optimization method that considers the label transitions of sentences in different segments of an article to improve the performance of the classifier. Experimental results based on real-world datasets demonstrate the superiority of the proposed method over state-of-the-art approaches.

Keywords: Content marketing articles · Classification · Topic.

1 Introduction

With the rapid development of online social media, more and more people use social media to communicate and obtain news information. As the leading social media platform in China, WeChat has about 1.213 billion monthly active users, covering more than 200 countries by January 2021 [1]. WeChat has an important function named “Official Accounts”, where users can subscribe to interesting accounts. WeChat allows any legal users to register an “Official Account”, where they can post their opinions and knowledge through articles, pictures, videos, and other kinds of content to attract followers. Due to their high influence, some select accounts are hired for online product marketing. However, advertisers usually exaggerate facts to attract consumers, causing economic losses of consumers and harming the creditworthiness of the platform. Therefore, it is desirable to identify advertising content for the purpose of supervision, which benefits both users and the platform.

A CM article is a new form of popular online mode for “advertisorials”. It is manifested with a specific purpose for product promotion. As direct product promotion articles are more likely to be detected and blocked, advertisers usually create CM articles by embedding advertising content into normal articles about hot spots. Thus, users read the advertisement parts involuntarily. Besides, the advertising content usually contains positive and affirmative words and phrases. Figure 1 shows an example of a CM article from a WeChat Official Account. The left part is a screenshot of the CM article, and the right is the English version obtained by Google Translate. The red box contains the normal content, the yellow part contains the transitional sentences from the normal part to the advertisements, and the green part is the advertising content.

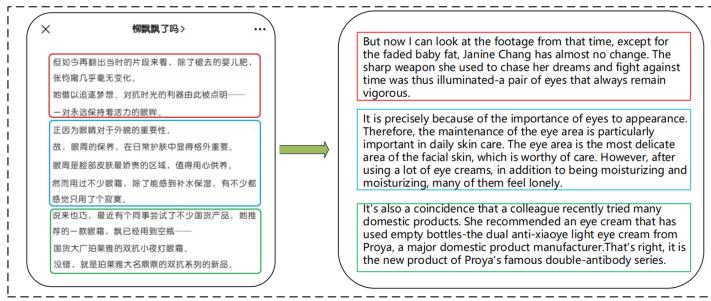


Fig. 1. An example of a CM article. (Color figure online)

There are already some researches to detect CM articles. Liang et al. [9] proposed a graph-based approach to detect CM articles in WeChat. However, the method fails to extract the advertisements from the detected CM articles. Extracting advertisements from marketing articles can benefit both users and the platform from two aspects. First, the platform can remove identified advertisements and thus improves the quality of the experience. Second, the platform can further identify illegal advertisements such as illegal drugs and phishing content. However, this task faces two challenges. First, the transition from normal content to advertisements is carefully designed, usually with few sentences connecting two topic-separated parts. Second, there are much fewer advertising sentences than normal ones in a CM article, which significantly increases the difficulty of identifying them.

In this paper, we study the task of extracting advertising content from a CM article, which is modeled as a sentence classification task. To address the first challenge, we propose a topic-enhanced deep neural network to enrich sentence encoding. Both the semantic information and topic features are utilized to generate sentence embeddings.

To address the second difficulty, we develop a segment-aware optimization method for sentence classification.

Our proposed model can capture the characteristics of CM articles and we conduct extensive experiments to evaluate the effectiveness of our model based on real-world datasets.

2 Related Work

2.1 Text Classification

Extracting advertisements from CM articles is essentially a text classification problem, which is fundamental in natural language processing (NLP). Some researchers employed the long and short-term memory (LSTM) models for text classification and achieved encouraging results [14, 16]. Character-level CNNs are designed for short text classification [2]. Wang et al. [17] incorporated entities and relations from knowledge graphs (KGs) to enrich the semantics of short texts. Yao et al. [7] employed the standard graph convolutional networks (GCN) for both long and short text classification and achieved superior performance over state-of-the-art text classification methods. Hu et al. [19] proposed a heterogeneous information network framework, which integrates several types of additional information to address the semantic sparsity problem in text classification. Wieting et al. [8] explore various methods for computing sentence representations from pretrained word embeddings without any training.

2.2 Topic Models

Topic models have been proposed to uncover the latent semantic structure of text corpus. Hoffmann [3] proposed a probabilistic latent semantic analysis (PLSA) for topic extraction. Beli et al. [4] proposed the most commonly used and effective topic model of Latent Dirichlet Allocation (LDA). However, traditional topic models are not suitable for short texts. Some researchers tried to aggregate short texts into lengthy pseudo-documents based on some additional information. Hong et al. [5] conducted a comprehensive empirical study of topic models in Twitter. Yan et al. [6] proposed a Biterm topic model (BTM), which learns topics by directly modeling the generation of word co-occurrence patterns in the whole corpus. BTM uses the aggregated patterns in the whole corpus to solve the sparsity problem encountered at the document-level. Angelov [21] proposed a top2vec model to find topics which are significantly more informative and representative of the corpus trained on than probabilistic generative models.

3 Method

3.1 Problem Definition

Content Marketing starts with attractive topics (e.g., hot spots) and transits to product promotion gradually. The two parts are usually about different topics and have different narrative styles. This paper focuses on distinguishing between

advertising sentences and normal ones in Content Marketing articles. Specifically, a CM article A contains a sequence of n sentences $A = \{s_1, \dots, s_i, \dots, s_n\}$. Our purpose is to find the advertising sentences in A . We propose a sentence classification method to achieve this goal. Figure 2 presents an overview of the proposed method. The input of the model is a sentence of an article, which contains a sequence of words. The input is that in order to play the role, Zhang Junning cut a head of hair, clever temperament receded, the whole beautiful and handsome.

We use $P(y|s, \theta)$ to denote the probability of sentence s being classified into label $y \in \{0, 1\}$, where θ represents the parameters of the method, and $y=1$ represents that s is advertising.

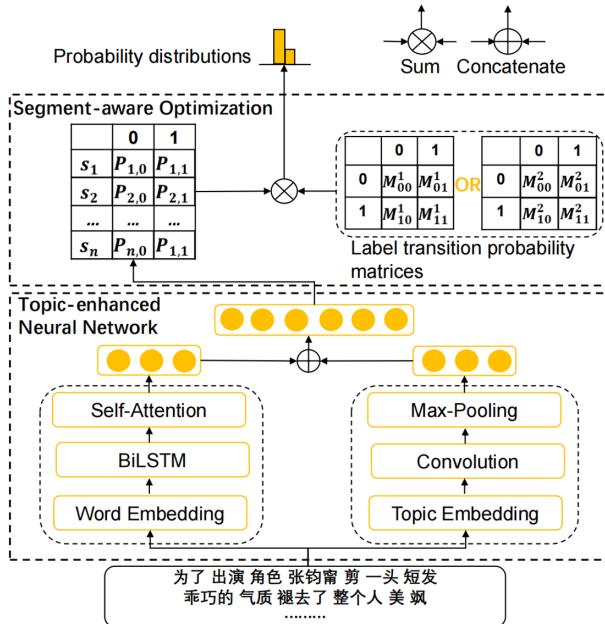


Fig. 2. Model architecture.

3.2 Topic-Enhanced Neural Network

Sentence Encoding. Given a sentence contains n words, we denote the sentence as $s = \{w_1, w_2, \dots, w_n\}$. The goal of this model is to learn the semantic representation \mathbf{s}_s for a given sentence s . We use the pre-trained word embeddings trained by word2vec for word vector initialization. In this paper, we employ the Bi-directional LSTM (BiLSTM) proposed by Hao et al. [10].

$$\overrightarrow{\mathbf{h}_t} = \overrightarrow{LSTM}(\mathbf{w}_t, \overleftarrow{\mathbf{h}_{t-1}}) \quad (1)$$

$$\overleftarrow{\mathbf{h}_t} = \overleftarrow{LSTM}(\mathbf{w}_t, \overleftarrow{\mathbf{h}_{t-1}}) \quad (2)$$

We concatenate $\overrightarrow{\mathbf{h}_t}$ and $\overleftarrow{\mathbf{h}_t}$ to obtain a hidden state \mathbf{h}_t of a sentence. Let the number of hidden units for each unidirectional LSTM be u . For simplicity, we denote all the \mathbf{h}_t 's as a matrix $\mathbf{H} \in \mathbb{R}^{n \times 2u}$:

$$\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_i, \dots, \mathbf{h}_n] \quad (3)$$

Self-attentions do not depend on the order of words, which could improve the expressive power of RNN. So we introduce a self-attention mechanism to address this problem. Specifically, we use the scaled dot-product attention [11]. The purpose is to learn the word dependence within a sentence and capture the internal structure of a sentence. Given a matrix of n query vectors $\mathbf{Q} \in \mathbb{R}^{n \times 2u}$, keys $\mathbf{K} \in \mathbb{R}^{n \times 2u}$ and values $\mathbf{V} \in \mathbb{R}^{n \times 2u}$, the scaled dot-product attention computes the attention scores as follows:

$$\mathbf{A} = \text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{2u}}\right)\mathbf{V} \quad (4)$$

where \mathbf{Q} , \mathbf{K} , \mathbf{V} are obtained by linear transformations, i.e., $\mathbf{Q} = \mathbf{HW}^Q$, $\mathbf{K} = \mathbf{HW}^K$, $\mathbf{V} = \mathbf{HW}^V$, \mathbf{W}^Q , \mathbf{W}^K , and \mathbf{W}^V are randomly initialized matrices, and $\sqrt{2u}$ is the scaling factor to avoid focusing too much on a word. The output of the attention layer is a matrix denoted by $\mathbf{A} \in \mathbb{R}^{n \times 2u}$.

Next, we use a max-pooling layer over \mathbf{A} to acquire the sentence representation $\mathbf{s}_s \in \mathbb{R}^{2u}$, i.e., choosing the highest value on each dimension of the vectors to capture the most important feature.

Topic Enriching. We use a convolutional neural network to obtain the topic features of a sentence. The input is the topic feature matrix of a sentence. The topic feature matrix $\mathbf{Z} \in \mathbb{R}^{N \times K}$ is obtained by the Biterm topic model (BTM). BTM is a topic model designed for short text, which models the generation of word co-occurrence patterns in the whole corpus. where N is the number of words contained in the whole corpus, and the K is the number of the topics.

In the topic feature matrix, each row represents the probability of a word belonging to different topics. We combine the topic probability distributions of words to get the topic feature representation of a sentence, which is represented as \mathbf{T} :

$$\mathbf{T} = [\mathbf{t}_1, \dots, \mathbf{t}_i, \dots, \mathbf{t}_n] \quad (5)$$

where \mathbf{t}_i is the i -th word topic feature of the sentence, and the n is the number of words in the sentence. The feature matrix of the input layers are feed to the convolutional layers to extract local topic features of sentences:

$$\mathbf{C}_t^{(i)} = f(\mathbf{W}_t \cdot \mathbf{T}_{i:i+l-1} + b_t) \quad (6)$$

where $\mathbf{C}_t^{(i)}$ is the output of the convolution layer, $f()$ is the ReLU activation function, $\mathbf{T}_{i:i+l-1}$ represents the convolution operation from i -th to $(i+l-1)$ -th, l is the size of the convolution kernel, \mathbf{W}_t is a weight matrix, and $b_t \in \mathbb{R}$ is a bias term. Then, we apply the max-pooling over the feature map produced

by the convolution layer. Finally, we concatenate \mathbf{s}_s and \mathbf{s}_t to obtain the final embedding of a sentence s , which is fed into a fully connected layer. Finally, we use an output layer to acquire the probability of each class label.

3.3 Segment-Aware Sentence Classification

CM articles implicitly follow some patterns, which can be exploited to improve the classification performance. For example, advertising sentences often appear at the end of the CM articles. Moreover, an advertising sentence is usually followed by another one. These patterns allow us to boost the classification performance. However, the number of advertising sentences is much fewer than that of normal ones in a CM article. The comparison is shown in Table 1. Therefore, directly counting the label transitions in the entire article will weaken the transition probability. That is, sentences have different label transition probability matrices at different positions in a CM article. To address this issue, we set a parameter $\alpha \in \{0, 1\}$ to control the statistical scope of an article. Specifically, we calculate two transition matrices in different segments of a CM article. The first segment is from the first sentence to $[n\alpha]$ -th sentence, and the second is from $([n\alpha] + 1)$ -th to the end. In this way, we can capture the label transition probability of CM articles in fine granularity. Let $\mathbf{M}^{(1)} \in \mathbb{R}^{2 \times 2}$ denotes the transition matrix of different labels in successive sentences for the first segment. The label transition probability matrix computes the probability as follows:

$$\mathbf{M}_{ij} = \frac{\#N_{ij}}{\#N} \quad (7)$$

where $\#N_{ij}$ is the number that a sentence with label i is followed by a sentence with label j in the first segment, $\#N$ is the total number of sequential label transitions. Similarly, we can calculate the transition matrix $\mathbf{M}^{(2)}$ for the second segment. The label transition matrix reflects the dependencies between the labels of adjacent sentences. Let \mathbf{P} be the predicted probability matrix output by the neural network, where \mathbf{P}_{i,y_i} indicates the score that the label y_i assigned to the i -th sentence. In order to model dependencies between subsequent labels, the score of a label is defined as the sum of the probabilities of individual labels and the transition probabilities. The score of a prediction label is defined as follows:

$$\hat{\mathbf{P}}_{i,y_i} = \mathbf{P}_{i,y_i} + \mathbf{M}_{y_{i-1}y_i} \quad (8)$$

where y_{i-1} is the label of the predecessor sentence.

The first sentence in a CM article has no label transition probability. The prediction of the first sentence takes the probability \mathbf{P}_{i,y_i} . Finally, the obtained probability matrix $\hat{\mathbf{P}}$ is normalized by taking a softmax operation over all possible labels:

$$\bar{\mathbf{P}}_{i,y_i} = \text{softmax}(\hat{\mathbf{P}}) = \frac{e^{\hat{\mathbf{P}}_{i,y_i}}}{\sum_{\hat{y}_i \in Y} e^{\hat{\mathbf{P}}_{i,\hat{y}_i}}} \quad (9)$$

where $Y \in \{0, 1\}$ denotes the set of labels. The objective of the training is to minimize the cross-entropy loss:

$$Loss = - \sum_{i=1}^n y_i \log \bar{\mathbf{P}}_{i, \hat{y}_i} \quad (10)$$

where y_i is the ground-truth label, and n is the number of sentences in a CM article.

4 Experiment

4.1 Experimental Settings

Dataset. We crawled the articles from WeChat official accounts and manually marked 700 CM articles with the assistance of the graph-based algorithm [9]. The paper proposed a novel approach to enhance the detection based on the sentence and word graph analysis. And they extract both the graph-related and community-related features from the graphs of the two types, respectively. After that, a supervised classifier is trained based on a manually labeled dataset.

After getting the dataset, we use Chinese and English punctuation to split a CM article into a sentence sequence. We use “jieba” to cut words and remove the stopwords. Then, we labeled the sentences containing advertising content. Table 1 shows a summary of the dataset.

Table 1. Summary of the dataset.

Statistic	Results
# of CM articles	700.00
Training set	560.00
Test set	140.00
of normal sentences	48719.00
of advertising sentences	11944.00
Avg. # of a CM article contains normal sentences	69.60
Avg. # of a CM article contains advertising sentences	17.06
Avg. # of a normal sentence contains words	23.64
Avg. # of an advertising sentence contains words	36.93

Compared Methods. To evaluate the effectiveness of our model, we compare it with the following methods.

- **Text CNN:** This model is a classic baseline for sentence-level text classification. In this experiment, we use pre-trained word embeddings to initialize the representations of words [22].

- **RCNN:** Lai et al. [20] proposed a recurrent convolutional neural network for text classification. It uses RNN to capture contextual information and uses CNN to capture the most prominent information of the text.
- **Topic CNN:** This model solves the task of advertising extraction from CM articles, which is to construct a convolutional neural network with two channels, including the traditional semantic CNN channel and the topic CNN channel [18].
- **Fast Text:** Joulin [15] proposed a simple and efficient text classification method. It treats the average of word/n-grams embeddings as document embeddings and then feeds them into a linear classifier.
- **Text GCN:** Yao et al. [7] employed the standard graph convolutional network for text classification. It outperforms state-of-the-art text classification method without using external knowledge.

Paramater Settings. For the comparative experimental models, we keep the parameter settings as those in the original paper. We use the pre-trained word embeddings trained by word2vec, and the dimension d is set to 300. The BTM model is used to obtain the potential topic information of sentences and uses the optimal parameter settings in the original paper except for the number of topics K . In our model, we set the number of topics K to be 90, and the value of parameter α is 0.5. We train this model for a maximum of 100 epochs using Adam optimizerf [12] and stop training if the test loss does not decrease. We set the learning rate as 0.001, dropout as 0.5.

4.2 Experimental Results

Table 2 shows the experimental results of different methods. It is shown that our approach achieves the best performance and significantly outperforms all baseline models. The results demonstrate that CNN-based models better precision than other baseline methods. However, these methods have lower recall than other methods. Different from these methods, our model achieves both the highest precision and recall. These findings verify the effectiveness of the proposed method in the task of extracting advertising content.

Table 2. Performance comparison.

Model	Precision	Recall	F_1
TextCNN	0.8151	0.5155	0.6316
RCNN	0.8273	0.5278	0.6445
TopicCNN	0.8318	0.5301	0.6475
FastText	0.7667	0.6326	0.6932
TextGCN	0.7876	0.6450	0.7092
Ours	0.8662	0.6476	0.7411

4.3 Impacts of Topics

The above analysis shows that the topic information is an important feature. Adding latent topic information can enrich sentences' semantic encoding. When the number of topics is too small, there is little difference in the topic information of each word. When the number of topics is too large, the topic granularity is finer and the words will be overly divided into topics.

Figure 3 displays the impacts of the number of topics on the performance of the proposed model. The results display that the performance of our model increases as the number of topics increases. When the number of topics reaches 90, the model reaches its optimum. When the number of topics is larger than 90, the performance of the model decreases as more topics are added. Therefore, $K = 90$ is an optimum setting of the number of topics.

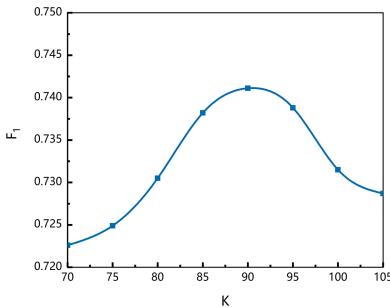


Fig. 3. The changes of F_1 versus the number of topics K .

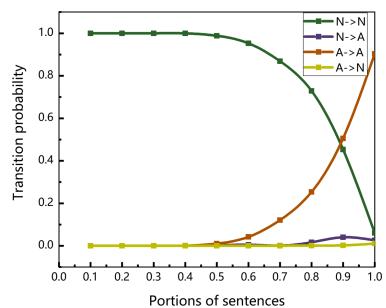


Fig. 4. The probability of label transfer for different percentages of sentences.

4.4 Impacts of the Parameter α

The parameter α , which controls the segmentation of an article, is also an important factor affecting the performance of the model. It is used to obtain more accurate label transition probability matrices to improve the classification accuracy.

In this section, we employ “N->A” to represent that a sentence labeled as normal is followed by a sentence labeled as advertising; “N->N” indicates that a sentence labeled as normal is followed by a sentence labeled as normal; “A->A” represents that a sentence labeled as advertising is followed by a sentence labeled as advertising; and “A->N” suggests that a sentence labeled as advertising is followed by a sentence labeled as normal.

We analyze the probabilities of different transition cases at different positions of CM articles. We evenly divide a CM article into ten portions and calculate the transition probabilities for each portion. Figure 4 shows the statistical results. It is shown that all CM articles begin with normal sentences. Moreover, the transition probability of “A->A” increases gradually, indicating that advertising

content is more likely to appear in the second half of CM articles. The transition probability of “N->A” reaches its peak at the 9th portion, suggesting that most advertisements are short and appear at the end of CM articles.

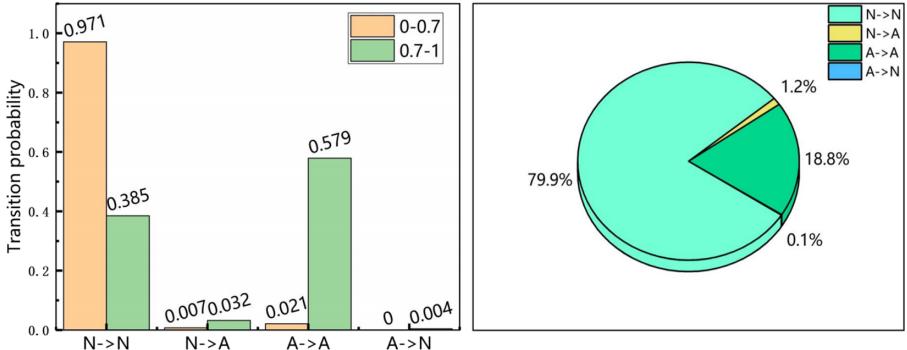


Fig. 5. Transition probabilities under different conditions

The left side of Fig. 5 presents the label transition probabilities calculated by setting α to be 0.7. It is shown that the transition probabilities calculated based on the two segments are very different. And the other side of Fig. 5 shows the label transition matrix calculated based on whole CM articles without using segments. From the Fig. 5, we can see that directly calculating a label transfer matrix without considering the advertising sentence positions in the CM articles will reduce calculated transition probabilities and thus harm the performance.

Figure 6 displays the effects of parameter α on the performance of the proposed model. The results show that F_1 scores of the proposed model increase as α becomes larger when $\alpha < 0.5$. The performance reaches the peak when $\alpha = 0.5$ and then decreases gradually with the increase of α . According to the results, we set $\alpha = 0.5$ as the default.

4.5 Ablation Analysis

In this experiment, we perform an ablation study to evaluate the effectiveness of two schemes. We compare the performance of four different methods: the full model, a model without considering topics (-topics), a model without considering segments (-segments), and a model without considering the effects of topics and segments (-topic&segments). The experimental results are presented in Table 3. The full model achieves the best performance without surprise. And, the topics and segments are complementary to each other.

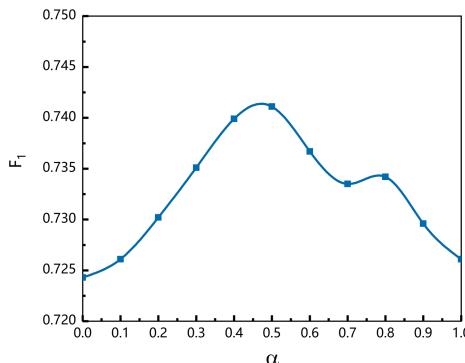


Fig. 6. The change of F_1 with the value of parameter α .

Table 3. Ablation analysis.

Model	Precision	Recall	F_1
Full Model	0.8662	0.6476	0.7411
-segments	0.8497	0.6313	0.7243
-topic	0.8533	0.6238	0.7202
-topic&segments	0.8349	0.6077	0.7034

5 Conclusion

In this paper, we propose a segment-aware sentence classification method for advertising sentences extraction from CM articles. We develop a topic enriching module to capture the semantic information of sentences. A segment-aware optimization scheme is proposed to capture the characteristics of CM articles. Experimental results demonstrate that our method outperforms state-of-the-art approaches for the advertisement extraction task.

Acknowledgment. The research presented in this paper is supported in part by National Natural Science Foundation of China (No. 61602370, U1736205, 61922067, 61902305), Shenzhen Basic Research Grant (JCYJ20170816100819428), Natural Science Basic Research Plan in Shaanxi Province (2021JM-018).

References

1. Most popular social networks worldwide as of January (2021). <https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/>
2. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. *Adv. Neural Inf. Process. Syst.* **28**, 649–657 (2015)
3. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 50–57 (1999)

4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (2003)
5. Hong, L., Davison, B.D.: Empirical study of topic modeling in twitter. In: Proceedings of the First Workshop on Social Media Analytics, pp. 80–88 (2010)
6. Yan, X., Guo, J., Lan, Y., Cheng, X.: A biterm topic model for short texts, In: Proceedings of the 22nd International Conference on World Wide Web, pp. 1445–1456 (2013)
7. Yao, L., Mao, C., Luo, Y.: Graph convolutional networks for text classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 7370–7377 (2019)
8. Wieting, J., Kiela, D.: No training required: Exploring random encoders for sentence classification. [arXiv:1901.10444](https://arxiv.org/abs/1901.10444) (2019)
9. Liang, X., Wang, C., Zhao, G.: Enhancing content marketing article detection with graph analysis. *IEEE Access* **7**, 94869–94881 (2019)
10. Hao, Y., et al.: An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Long Papers), vol. 1, pp. 221–231 (2017)
11. Luong, M.-T., Pham, H., Manning, C. D.: Effective approaches to attention-based neural machine translation. [arXiv:1508.04025](https://arxiv.org/abs/1508.04025) (2015)
12. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
13. Sinha, K., Dong, Y., Cheung, J.C.K., Ruths, D.: A hierarchical neural attention-based text classifier, In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 817–823 (2018)
14. Liu, P., Qiu, X., Huang, X.: Recurrent neural network for text classification with multi-task learning, 2873–2879 (2016)
15. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. arXiv preprint [arXiv:1607.01759](https://arxiv.org/abs/1607.01759) (2016)
16. Luo, Y.: Recurrent neural networks for classifying relations in clinical notes. *J. Biomed. Inf.* **72**, 85–95 (2017)
17. Wang, J., Wang, Z., Zhang, D., Yan, J.: Combining knowledge with deep convolutional neural networks for short text classification., In: IJCAI, vol. 350, pp. 2915–2921 (2017)
18. Fan, X., Wang, C., Liang, X.: Extracting advertisements from content marketing articles based on topiccnn. In: 2020 IEEE International Conference on Dependable, Autonomic and Secure Computing, IEEE International Conferenceon Pervasive Intelligence and Computing, IEEE International Conferenceon Cloud and Big Data Computing, IEEE International Conferenceon Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), pp. 355–360. IEEE (2020)
19. Linmei, H., Yang, T., Shi, C., Ji, H., Li, X.: Heterogeneous graph attention networks for semi-supervised short text classification. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 4823–4832 (2019)
20. Lai, S., Xu, L., Liu, K., Zhao, J.: Recurrent convolutional neural networks for text classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29, pp. 2267–2273 (2015)
21. Angelov, D.: Top2Vec: distributed representations of topics. [arXiv:2008.09470](https://arxiv.org/abs/2008.09470) (2020)
22. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 1746–1751 (2014)



Leveraging Lexical Common-Sense Knowledge for Boosting Bayesian Modeling

Yashen Wang^(✉)

China Academy of Electronics and Information Technology, Beijing, China
yswang@bit.edu.cn

Abstract. Recent research has shown that, since the ultimate goal of Bayesian perspective is to infer a posterior distribution, it is demonstrably more direct to impose domain knowledge directly on the posterior distribution. This paper presents a model by imposing lexical common-sense knowledge as constraints on the posterior distribution, under the conventional regularized Bayesian framework. We then improve the latent topic modeling with help of the aforementioned model, and experimental results show that, combining lexical common-sense knowledge and Bayesian modeling, is beneficial for prediction.

Keywords: Lexical knowledge · Common sense · Bayesian modeling · Topic model

1 Introduction

Recently, many research has demonstrated that, incorporating structural domain knowledge into the conventional machine learning task, is an effective way to improve the accuracy [12, 19, 24] or the interpretability of latent representations [1, 10]. Especially, [31] and [14] show that, Bayesian perspective provides a precise mathematical framework to incorporate extra domain knowledge via Bayes' rule. The regularized Bayesian framework (RegBayes) [31] imposes domain knowledge via posterior constraints, using a variational representation of Bayes' rule. RegBayes has had significant success in learning discriminatory Bayesian models [29, 30]. [14] present a direct approach by imposing First-Order Logic (FOL) rules on the posterior distribution, which unifies FOL and Bayesian modeling under the RegBayes framework. Besides, this approach could automatically estimate the uncertainty of FOL rules when they are produced by humans, so that reliable rules are incorporated while unreliable ones are ignored. [14] provides a reasonable and clear Implementation path for introducing domain knowledge into Bayesian perspective.

Inspired by [14], we investigate whether lexical common-sense knowledge could be adopted for Bayesian model. In this paper, we follow the overall architecture of [14], and introduce lexical RegBayes (lRegBayes), a principled framework to robustly incorporate rich and uncertain lexical common-sense knowledge—concept, mainly consists of *isA* semantics and *isAttributeOf* semantics provided by lexical knowledge graph Probbase, in machine learning tasks, such as topic modeling. Psychologist Gregory Murphy began his highly acclaimed book [15] with the statement “*Concepts are*

the glue that holds our mental world together”. Still, Nature magazine book review calls it an understatement, because “Without *concepts*, there would be no mental world in the first place” [3]. Doubtless to say, the ability to conceptualize is a defining characteristic of humanity. [10, 23] The idea of using learned concepts [27] to improve machine learning and natural language processing tasks has been explored previously, including text embedding [25], text conceptualization [21], information retrieval [24], entity disambiguation [4], query understanding [22], semantic conceptualization [17], text segmentation [9], knowledge graph completion [26], and so on. Previous work has shown concept’s strong interpretability and *anti-noise* capability, which is effective and robust in helping understanding semantic and could be combined with the conventional machine learning methodology. Especially, we propose the use of concepts to guide the conventional RegBayes. This provides more flexibility in text modeling and also the ability to infer the posterior on latent codes, which could be useful for visualization and downstream machine learning tasks.

2 Preliminary

To enhance the representation ability of the proposed framework, this paper introduces extra lexical knowledge (i.e., *concept* knowledge from Probbase [21, 27] emphasized here), which has been proved to be effective in helping understanding semantic in many NLP tasks [24, 27, 28].

2.1 Definition

(Def.1) **Concept.** Following [10, 23], we define a “concept” as a set or class/category of “entities” or “things” within a domain, such that words belonging to similar classes get similar representations. E.g., “microsoft” and “amazon” could be represented by concept COMPANY. Probbase [27] is used in our study as knowledge graph.

(Def.2) **Conceptualization.** Given a text $s_i = \{w_1, w_2, \dots, w_{|s_i|}\}$, wherein w_i denotes a word, text conceptualization algorithm enables to select the open-domain concepts $C_{s_i} = \{\langle c_i, p_i \rangle | i = 1, \dots, \}$ from the knowledge graph Probbase which own the optimal ability for discriminatively representing the given text s_i . E.g., given a text as input (e.g., “microsoft unveils office for apple’s ipad”), we generate the concepts $C_{s_i} = \{\langle \text{COMPANY}, 0.8567 \rangle, \langle \text{BRAND}, 0.7457 \rangle, \langle \text{PRODUCT}, 0.5471 \rangle, \dots\}$ from Probbase for this text context. Besides, the concept vector θ_i is generated based on c_i and its corresponding probability p_i : each dimensionality of θ_i represents the probability p_i of the concept c_i in the given text.

2.2 Probbase

Probbase¹ is widely used in research about text understanding [21, 22, 27], text representation [10, 25], information retrieval [24], and knowledge graph completion [26]. Probbase uses an automatic and iterative procedure to extract concept knowledge from 1.68 billion Web pages. It contains 2.36 millions of open-domain terms, and each term is

¹ <https://concept.research.microsoft.com/>.

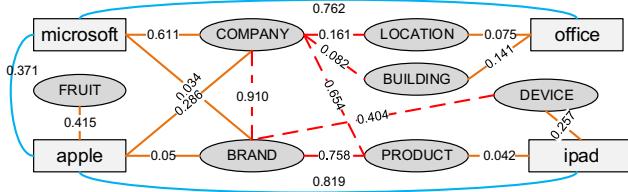


Fig. 1. The example sketch of the lexical knowledge graph **Probase** [27].

a concept, an instance (respect to a word occurring in given text in this study), or both. Meanwhile, it provides around 14 millions relationships with two kinds of important knowledge related to concepts: concept-attribute co-occurrence (*isAttributeOf*) and concept-instance co-occurrence (*isA*). For clarity, Fig. 1 sketches the organization of instances and their corresponding concepts defined in Probase [25]. Moreover, Probase provides huge number of high-quality and robust concepts without builds. Therefore, lexical knowledge graph Probase is utilized in this paper for leveraging lexical semantics for boosting efficiency of language modeling, with help of its strong interpretability and anti-noise capability.

3 Methodology

3.1 RegBayes Framework

Suppose a Bayesian latent variable model with observed random variables $\mathbf{O} \in \mathcal{O}$ and hidden variables $\mathbf{H} \in \mathcal{H}$. In this case, conventional Bayesian inference calculates the *posterior* distribution $\mathcal{P}(\mathbf{H}|\mathbf{O})$ from a *prior* $\mathcal{P}_0(\mathbf{H})$ and a *likelihood* model. It is usually hard to make sure whether the aforementioned posterior ($\mathcal{P}(\mathbf{H}|\mathbf{O})$) satisfies (all) common-sense knowledge constraints or not. On the contrary, the recently-proposed **RegBayes** framework [31] allows domain knowledge to directly influence the posterior, i.e., $(\mathcal{P}(\mathbf{H}|\mathbf{O}))$. Especially, it RegBayes reaches this target by penalizing distributions that differ in the expected value of feature functions. Nonetheless, the domain knowledge considered in RegBayes has been max-margin posterior constraints, which could be too narrow and inapplicable to machine learning [14]. In our study, each feature function σ_i , and the “belief label” of the corresponding feature ρ_i , are induced from lexical common-sense knowledge, which is also reviewed as a kind of semantic constraint.

3.2 LRegBayes with Lexical Common-Sense Knowledge

Based on [14, 31] introduces First-Order Logic (FOL) rules into regularized Bayesian model, and proposes an elastic framework for absorbing extra domain knowledge, named as Robust RegBayes. Following the architecture of [14], this paper investigating how to leveraging lexical common-sense knowledge for boosting Bayesian modeling. Therefore, in this paper, we propose a LRegBayes framework by introducing lexical common-sense knowledge into conventional RegBayes. Inspired by [14], we could also

define the inference procedure of our lRegBayes as a constrained optimization problem:

$$\min_{\hat{\mathcal{P}}(\mathbf{H}) \in \mathcal{P}, \eta_i \in \mathbb{R}_+^L} \text{KL}[(\mathcal{P}(\mathbf{H}) \parallel \mathcal{P}(\mathbf{H}|\mathbf{O})) + \Omega \sum_i \eta_i] \quad (1)$$

$$\text{s.t. } |\mathbb{E}_{\hat{\mathcal{P}}(\mathbf{H})}[\sigma_l(\mathbf{H}, \mathbf{O})] - \rho_i| \leq \epsilon + \eta_i \quad (2)$$

wherein, \mathcal{P} represents the appropriate probability; $\eta_i \in \mathbb{R}_+^N$ is the vector of N slack variables, one for each lexical knowledge constraint; $\mathcal{P}(\mathbf{H}|\mathbf{O})$ indicates the posterior distribution via Bayes' perspective; ϵ indicates a positive precision parameter which is usually small; and notation Ω indicates a regularization parameter.

To adequately broaden the scope of common-sense knowledge used in RegBayes, we consider `isA` constraints [10, 25] between *concepts* and *instances*² and `isAttributeOf` constraints among between *concepts*³ in this paper, which could be derived easily from the high-quality lexical knowledge graph Probbase [27]. Probbase provides huge number of high-quality, structural and robust `isA/isAttributeOf` constraints without builds, which have been demonstrated readily represented and adopted for many NLP tasks [21, 23, 24]. Moreover, Probbase has the goal of modeling lexical common-sense constraints in probabilistic terms.

Formally, we denote μ_i be the i -th lexical common-sense constraint represented in “triple” form (i.e., “(instance, `isA`, concept)” for `isA` relation, and “(concept, `isAttributeOf`, concept)” for `isAttributeOf` relation) over instantiations (\mathbf{h}, \mathbf{o}) of the variables (\mathbf{H}, \mathbf{O}) . Feature function σ_i , respect to the corresponding constraint μ_i , is defined as follows: $\sigma_i = \frac{1}{|\mathcal{U}_i|} \sum_{\mu_i \in \mathcal{U}_i} \mathbb{I}(\mu_i(\mathbf{h}, \mathbf{o}))$. Wherein, $\mathbb{I}(\cdot)$ represents a indication function.

Similar to conventional RegBayes, the proposed lRegBayes aims at effecting a separate Bayes model. Therefore, lRegBayes truly combines lexical common-sense and Bayesian modeling. Hence, we could automatically learn the weights of lexical constraint. In the proposed framework, we learns the constraint’s weights from relatively easier-to-obtain belief labels ρ_i via solving a dual optimization problem, like [14].

3.3 LDA Driven by lRegBayes

This section describes how the proposed lRegBayes to learn LDA topics by incorporating lexical common-sense knowledge. In conventional LDA, each document d is drawn from an mixture of K topics, i.e., $\{\tau_k | k \in [1, \dots, K]\}$. Each topic τ_k is defined as a multinomial distribution over a given vocabulary and follows a Dirichlet prior $\mathcal{P}(\tau_k|\beta) = \text{Dir}(\tau_k|\beta)$. For document d , we draw a topic distribution θ_d from Dirichlet distribution $\mathcal{P}(\theta_d|\alpha) = \text{Dir}(\theta_d|\alpha)$. For the j -th word in document d : (i) we draw a topic assignment $z_{d,j}$ from the multinomial parametrized by θ_d , $\mathcal{P}(z_{d,j} = \tau_k|\theta_d) = \theta_{d,k}$; (ii) and then draw the word $w_{d,j}$ from the selected topic $\tau_{z_{d,j}}$, $\mathcal{P}(w_{d,j}|z_{d,j}, \tau) = \tau_{z_{d,j}, w_{d,j}}$. With efforts above, the joint distribution of LDA can be formulated as follows:

$$\mathcal{P}(\mathbf{W}, \mathbf{Z}, \tau, \theta|\alpha, \beta) = [\prod_k \mathcal{P}(\tau_k|\beta)] (\prod_d \mathcal{P}(\theta_d|\alpha) \prod_j \mathcal{P}(z_{d,j}|\theta_d) \mathcal{P}(w_{d,j}|z_{d,j}, \tau)) \quad (3)$$

² E.g., word-formed instance “Microsoft” `isA` concept COMPANY.

³ E.g., concept BIRTHDAY `isAttributeOf` concept PERSON.

wherein, notation \mathbf{W} denotes the set of the observed words and $\mathbf{W} = \{w_{d,j}\}$. Besides, $\mathbf{Z} = \{z_{d,j}\}$, $\theta = \{\theta_{d,k}\}$, and $\tau = \{\tau_{d,k}\}$ are the hidden variables. Bayesian modeling tries to infer the *posterior* over hidden variables $\mathcal{P}(\mathbf{W}, \mathbf{Z}, \tau, \theta, \alpha, \beta)$.

In this situation, for lexical common-sense knowledge, we assume that all the lexical constraints are defined over the instantiation of words \mathbf{W} and hidden topic assignments \mathbf{Z} , following [14]. For simplicity, the we omit the uncertainty belief label proposed by [14], which is designed as a spike-slab likelihood form, for handling the uncertainty in domain knowledge. Note that, Bayesian methods naturally handle *noise* in extra knowledge, which is especially important when domain knowledge is collected from the crowd [5, 18].

With efforts above, we now have $\mathbf{H} = \{\mathbf{Z}, \theta, \tau\}$ and $\mathbf{O} = \mathbf{W}$. Based on Eq. (1), we get the optimization of learning lRegBayes driven LDA, as follows:

$$\min_{\hat{\mathcal{P}}(\mathbf{H}) \in \mathcal{P}, \eta_i \in \mathbb{R}_+^L} \text{KL}[(\mathcal{P}(\mathbf{H}, \rho) \parallel \mathcal{P}(\mathbf{H}, \rho|\mathbf{W}, \alpha, \beta)) + \Omega \sum_i \eta_i] \quad (4)$$

$$\text{s.t. } \mathbb{E}\{\mathbb{E}_{\hat{\mathcal{P}}(\mathbf{Z})}[\sigma_l(\mathbf{Z}, \mathbf{W})] - \mathbb{E}_{\hat{\mathcal{P}}(\rho_i)}[\rho_i]\} \leq \epsilon + \eta_i, \forall i = 1 \dots N \quad (5)$$

4 Experiments

Many recent research have been done on releasing an informative prior, either *directly* [8] or by imposing parameter constraints and confidence values *indirectly* [13]. In this experimental section, we evaluate our lRegBayes on text clustering task, which belongs to *indirect* evaluation task, and interpret its improvements quantitatively.

4.1 Datasets

For text clustering task, we use three datasets: **NewsHeadline**, **Twitter** and **TREC**, described as follows:

NewsHeadline: Because news headlines usually contain few words and many special or ambiguous words, this paper, we extract news headlines and first-sentences from a news corpus containing 8.14 million articles searched from Reuters and New York Time. The news articles are classified into six categories: *economy*, *religion*, *science*, *traffic*, *politician*, and *sport*. These categories are utilized as the ground-truth labels for short-text clustering experiments. Similar to [21], we randomly select 30,000 news headlines and first-sentences in each category.

Twitter: We utilize two official tweet collections released in TREC Microblog Task 2011/2012 [16] and TREC Microblog Task 2013/2014 [11], to construct this dataset. By manually labeling, the dataset contains 435,987 tweets which are in four categories: *food*, *sport*, *entertainment*, and *device/IT company*. Similarly, these categories are utilized as the ground-truth labels for short-text clustering experiments. The URLs and stop-words are all removed, and the average length of the tweets is 10.05 words. Because of noise and sparsity, this dataset is more challenging.

WikiFirst: This dataset includes 330,000 Wikipedia articles, and is divided into 110 categories based on the mapping relationship between Wikipedia articles and Freebase

topics [28]. E.g., Wikipedia articles “The Big Bang Theory” are mapped into category *TV_program*. Each category contains 3,000 Wikipedia articles. We only keep the first sentence of each Wikipedia article in this dataset, and the average length of the first sentence is 15.43 words. It is a very challenging data set because of its large number of categories, strong diversity of categories and its distinct correlation among many categories.

4.2 Alternative Algorithms

We compare the proposed **IRegBayes-LDA** with the following baselines:

BOW: It represents short-text as bag-of-words with the TF-IDF scores [20]. Words with high frequency in the current short-text while low frequency in the entire dataset, will be assigned with a higher TF-IDF score.

LDA: It represents short-text as its inferred topic distribution [2], and the dimensions of the short-text vector of is number of topics as we presuppose.

ESA: Unlike **LDA**, which utilizes the distribution of latent topics to represent short-text, this algorithm calculates TF-IDF scores of words and concepts on the Wikipedia articles (i.e., and dataset **Wiki**, and Wikipedia articles is regarded as “topics” in this algorithm), and uses the distribution of Wikipedia articles to represent short-text [6,7].

Moreover, **TWE** [12] is recently proposed for word vector and short-text vector generation with help of conventional **LDA**. Similarly, we replace **LDA** module with our **IRegBayes-LDA**, and obtain the variant of **TWE** as another baseline, denoted as **IRegBayes-TWE**.

4.3 Experiment Settings

For auxiliary training the aforementioned **LDA**, **ESA**, **BOW** and so on, we introduce a Wikipedia snapshot, and hypothesize that retrieving from a large and high-fidelity corpus will provide cleaner language. Therefore, we construct a Wikipedia dataset (denoted as **Wiki** dataset here) for training comparative models if necessary, with the following rules proposed in [10]: (i) we remove the articles less than 100 words, as well as the articles less than 10 links; (ii) we remove all the category pages and disambiguation pages; (iii) we move the content to the redirection pages. Finally, we obtain about 3.74 million Wikipedia articles for indexing and training.

In text clustering task: (i) we set the topic number to be the cluster number or twice, and report the better of the two. (ii) we use two methods to train all the alternative algorithms: train them only on the datasets used in short-text clustering experiment (because the data used here is short-text, so the topic model is greatly affected by data’s sparsity); train them on the Wikipedia dataset **Wiki** as well as the datasets used in the following short-text clustering experiment. For **ESA**, we select the Top-1,000, Top-2,000, Top-5,000 and Top-10,000 concepts (respect to Wikipedia articles) as the clustering features respectively, and report the better of them.

4.4 Performance Summary

The results in Table 1 show that, the proposed **IRegBayes** framework has the ability for boosting conventional machine learning algorithms. It demonstrates the improved

Table 1. Performance of short-text clustering task. The superscript † denotes statistically significant improvements over **TWE** [12] ($p^* < 0.05$).

Models	NewsHeadline			Twitter			WikiFirst		
	Purity	ARI	NMI	Purity	ARI	NMI	Purity	ARI	NMI
BOW [20]	0.614	0.566	0.674	0.275	0.260	0.292	0.294	0.415	0.525
ESA [7]	0.725	0.669	0.797	0.404	0.382	0.430	0.337	0.476	0.602
LDA [2]	0.614	0.567	0.675	0.314	0.297	0.334	0.311	0.439	0.555
IRegBayes-LDA (Ours)	0.650	0.600	0.714	0.332	0.314	0.353	0.329	0.464	0.587
TWE [12]	0.741	0.684	0.814	0.378	0.358	0.402	0.375	0.529	0.669
IRegBayes-TWE (Ours)	0.781 [†]	0.721	0.858 [†]	0.398	0.327	0.424	0.375	0.557 [†]	0.705 [†]

task performance and topic interpretability in both machine learning, and the improvement ability from common-sense knowledge for traditional machine learning methods. Interestingly, the performance of the simplest baseline algorithm, **BOW**, is comparable to that of **LDA**, both of which are significantly worse than other algorithms. We try to set the number of topics of **LDA** as the number of clustering clusters or twice following [21, 22], and the former’s experimental results are better, which indicates that with the increase of the topic number, the clustering effect actually shows a downward trend. Compared with **LDA**, the performance of **ESA** based on Wikipedia improves significantly, and this phenomenon could be explained as that it is important to leverage extra knowledge resources for understanding concepts. The comparison between **TWE** and its lexical common-sense driven variant **IRegBayes-TWE**, shows a novel way for improving model’s performance. That is, the proposed **IRegBayes** plays like a *plug-in*, and experimental results demonstrate that by plugging our **IRegBayes**, the performance of **TWE** is improved significantly. E.g., **IRegBayes-TWE** exceeds original **TWE** by 5.41%, 5.43% and 5.38% respectively in the aforementioned datasets (measured by metric NMI). Without doubt, this indicates that the proposed **IRegBayes** has provided a flexible and natural modeling tool to improve conventional machine learning algorithms by integrating extra knowledge, such as lexical common-sense knowledge emphasized in this paper. We could conclude that, the proposed model could extend the scope of RegBayes prior knowledge by allowing lexical common-sense constraints. Moreover, as discussed above, no existing RegBayes model has explicitly modeled the noise in extra knowledge. We do not deliberately and explicitly model the uncertainty in lexical common-sense knowledge, compared with [14] which introduces a spike-and-slab prior and allows to automatically and selectively incorporate high-quality lexical domain knowledge while ignoring low-quality ones. This is mainly because the quality of lexical knowledge graph Probbase is guaranteed. Note that, other types of domain knowledge could also be introduced into our **IRegBayes**, because of the flexibility and universality of this framework.

5 Conclusions

For boosting Bayesian modeling, this paper proposes lexical RegBayes (abbreviated as **IRegBayes**) for utilizing lexical common-sense knowledge. We then apply our approach

to latent topic modeling tasks. Experimental results demonstrate that, incorporating domain knowledge (e.g., concept emphasized here) as constraints, is beneficial for prediction. By combining lexical common-sense knowledge and Bayesian modeling, we not only improve the task performance but also model interpretability.

Acknowledgements. We thank anonymous reviewers for valuable comments. This work is funded by: (i) the National Natural Science Foundation of China (No. U19B2026); (ii) the New Generation of Artificial Intelligence Special Action Project (No. AI20191125008); (iii) the National Integrated Big Data Center Pilot Project (No. 20500908, 17111001, 17111002).

References

1. Andrzejewski, D., Zhu, X., Craven, M., Recht, B.: A framework for incorporating general domain knowledge into latent Dirichlet allocation using first-order logic. In: IJCAI (2011)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (2003)
3. Bloom, P.: Glue for the mental world. *Nature* **421**, 212–213 (2003)
4. Chen, L., Liang, J., Xie, C., Xiao, Y.: Short text entity linking with fine-grained topics. In: CIKM 2018 (2018)
5. Chu, Z., Ma, J., Wang, H.: Learning from crowds by modeling common confusions. ArXiv abs/2012.13052 (2020)
6. Egozi, O., Gabrilovich, E., Markovitch, S.: Concept-based feature generation and selection for information retrieval. In: AAAI (2008)
7. Gabrilovich, E., Markovitch, S.: Overcoming the brittleness bottleneck using Wikipedia: enhancing text categorization with encyclopedic knowledge. In: National Conference on Artificial Intelligence, pp. 1301–1306 (2006)
8. Garthwaite, P., Kadane, J., O'Hagan, A.: Statistical methods for eliciting probability distributions. *J. Am. Stat. Assoc.* **100**, 680–701 (2005)
9. Hua, W., Wang, Z., Wang, H., Zheng, K., Zhou, X.: Short text understanding through lexical-semantic analysis. In: IEEE International Conference on Data Engineering, pp. 495–506 (2015)
10. Huang, H., Wang, Y., Feng, C., Liu, Z., Zhou, Q.: Leveraging conceptualization for short-text embedding. *IEEE Trans. Knowl. Data Eng.* **30**(7), 1282–1295 (2018)
11. Lin, J., Efron, M., Wang, Y., Sherman, G.: Overview of the TREC-2014 microblog track (2015)
12. Liu, Y., Liu, Z., Chua, T.S., Sun, M.: Topical word embeddings. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
13. Mao, Y., Lebanon, G.: Domain knowledge uncertainty and probabilistic parameter constraints. In: UAI (2009)
14. Mei, S., Zhu, J., Zhu, J.: Robust RegBayes: selectively incorporating first-order logic domain knowledge into Bayesian models. In: ICML (2014)
15. Murphy, G.L.: The Big Book of Concepts. MIT Press, Cambridge (2002)
16. Ounis, I., Macdonald, C., Lin, J.: Overview of the TREC-2011 microblog track (2011)
17. Park, J.W., Hwang, S.W., Wang, H.: Fine-grained semantic conceptualization of framenet. In: AAAI, pp. 2638–2644 (2016)
18. Raykar, V.C., Yu, S., Zhao, L., Hermosillo, G., Florin, C., Bogoni, L., Moy, L.: Learning from crowds. *J. Mach. Learn. Res.* **11**, 1297–1322 (2010)
19. Richardson, M., Domingos, P.M.: Markov logic networks. *Mach. Learn.* **62**, 107–136 (2006)

20. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, New York (1983)
21. Song, Y., Wang, H., Wang, Z., Li, H., Chen, W.: Short text conceptualization using a probabilistic knowledgebase. In: Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume, vol. 3, pp. 2330–2336 (2011)
22. Song, Y., Wang, S., Wang, H.: Open domain short text conceptualization: a generative + descriptive modeling approach. In: Proceedings of the 24th International Conference on Artificial Intelligence, pp. 3820–3826 (2015)
23. Wang, F., Wang, Z., Li, Z., Wen, J.R.: Concept-based short text classification and ranking. In: The ACM International Conference, pp. 1069–1078 (2014)
24. Wang, Y., Huang, H., Feng, C.: Query expansion based on a feedback concept model for microblog retrieval. In: International Conference on World Wide Web, pp. 559–568 (2017)
25. Wang, Y., Huang, H., Feng, C., Zhou, Q., Gu, J., Gao, X.: CSE: conceptual sentence embeddings based on attention model. In: 54th Annual Meeting of the Association for Computational Linguistics, pp. 505–515 (2016)
26. Wang, Y., Liu, Y., Zhang, H., Xie, H.: Leveraging lexical semantic information for learning concept-based multiple embedding representations for knowledge graph completion. In: APWeb/WAIM (2019)
27. Wu, W., Li, H., Wang, H., Zhu, K.Q.: Probbase: a probabilistic taxonomy for text understanding. In: ACM SIGMOD International Conference on Management of Data, pp. 481–492 (2012)
28. Yu, Z., Wang, H., Lin, X., Wang, M.: Understanding short texts through semantic enrichment and hashing. IEEE Trans. Knowl. Data Eng. **28**(2), 566–579 (2016)
29. Zhu, J.: Max-margin nonparametric latent feature models for link prediction. ArXiv abs/1206.4659 (2012)
30. Zhu, J., Chen, N., Xing, E.: Infinite latent SVM for classification and multi-task learning. In: NIPS (2011)
31. Zhu, J., Chen, N., Xing, E.: Bayesian inference with posterior regularization and applications to infinite latent SVMs. J. Mach. Learn. Res. **15**, 1799–1847 (2014)



Aggregating Inter-viewpoint Relationships of User's Review for Accurate Recommendation

Xingchen He, Yidong Chen, Guocheng Zhang, and Xuling Zheng^(✉)

Xiamen University, Xiamen, China

{xingchen,zgccc}@stu.xmu.edu.cn, {ydchen,xlzheng}@xmu.edu.cn

Abstract. User reviews contain rich information about user interests in items. Recently, many deep learning methods have attempted to integrate review contents into user preference rating prediction, helping to solve data sparseness problems. However, existing methods suffered from an inherent limitation that many reviews are noisy and contain non-consecutive viewpoints, besides, they are insufficient to capture inter-viewpoint relationships. Incorporating useful information is helpful for more accurate recommendations. In this paper, we propose a neural recommendation approach with a **Diversity Penalty** mechanism and **Capsule Networks**, named DPCN. Specifically, the diversity penalty component employs weight distributed matrices with the penalization term to capture different viewpoints in textual reviews. The capsule networks are designed to aggregate individual viewpoint vectors to form high-level feature representations for feature interaction. Then we combine the review feature representations with the user and item ID embedding for final rating prediction. Extensive experiments on five real-world datasets validate the effectiveness of our approach.

Keywords: Recommender systems · User reviews · Capsule networks · Rating prediction

1 Introduction

A core task in recommendation systems [1] is learning accurate representations of users and items to capture users' preferences. The earlier recommendation methods [11] learn user and item presentations from the user-item rating matrix. For example, Collaborative Filtering (CF) [15] methods are the successful approaches for recommender systems, which utilizes user-item interaction records for rating prediction. However, there are numerous users and items in the online platforms, the user-item rating matrix will be very sparse, matrix-based methods may suffer from the data sparsity problem.

To remedy this problem, review-based recommendation methods [7] are proposed. In recent years, many online shopping platforms allow users to write reviews after purchasing commodities, along with rating scores to express users'

likes or dissatisfaction. For example, a review like “I only wish I would have purchased it sooner” may infer the user’s favorite for the product. Because of the abundant information contained in the review content, it is potential to dig out the textual review to make a personality recommendation.

Several recommendation systems have harnessed the information of both ratings and reviews [7] to predict user ratings for more accurate recommendations. Further, owing to the excellent capability of deep learning techniques, many recent models [8, 9, 13, 20] use the deep neural network to model the user and item representations by the reviews. In these works, the convolutional neural network (CNN) is widely used, by encoding the word embedding to capture the corresponding latent features in the reviews. Examples include DeepCoNN [20], D-attn [13], CARL [17] and DAML [8].

While these proposed methods have performance in good results, some limitations prevent their performance from improving further. Firstly, the existing methods may be insufficient to model the long-term and non-consecutive viewpoints written in the same review. Secondly, one user may have different emotional expressions for one item, for example, customers like the color of the product but don’t like the shape of it. Hence, it is beneficial to consider diverse aspects of review content.

To overcome the above issues, we proposed a model based on the penalty mechanism and capsule networks for rating prediction, named DPCN. To capture different viewpoints and aspects in review content, DPCN employs a weight distribution matrix with penalization term to model the review texts into diversity feature vectors. The capsule networks are used to aggregate various feature vectors into high-level feature representations by dynamic routing [12]. Finally, the represent vectors are concatenated and feed into the feature interaction module for rating prediction. Our experiments demonstrate the effectiveness of DPCN in rating prediction tasks on five datasets. The main contributions of our work are summarized as follows:

- We proposed DPCN, a novel review-based recommendation model that can effectively model the diverse viewpoints of review texts to improve the recommendation performance.
- A diversity penalty mechanism is employed to capture long-term or non-consecutive viewpoints. We manually annotate a subsets of multi-viewpoint reviews to verify the effectiveness of penalization strategy.
- We design some experiments to demonstrate that capsule networks can preserve and aggregate the distribution of feature to help improve recommendation performance.
- Extensive experiments conducted on five real datasets demonstrate that our proposed framework generates certain better results compared to existing recommendation methods.

2 The Proposed Model

In this section, we describe our model framework for learning and predicting users’ preferences in detail. Users’ review texts contain numerous semantic

information that can effectively express customers' tendencies and interests. So we design a hierarchical model that contains two components: the diversity contextual viewpoints learning component and the high-level features aggregation component. Figure 1 illustrates the overall architecture of DPCN model.

- **The diversity contextual viewpoints learning component:** This part utilizes transformer encoder layers accompanying with the penalty mechanism to gain the semantically meaningful distributed representations of individual words.
- **The high-level features aggregation component:** This part makes use of the capsule networks that make it possible to capture the features which are consistently important for disparate aspects [19].

After getting user and item high-level feature distributions respectively. We combine the feature vectors with ID embedding and employ the Neural Factorization Machine (NFM) [2] to model the nonlinear interactions between user and item representations.

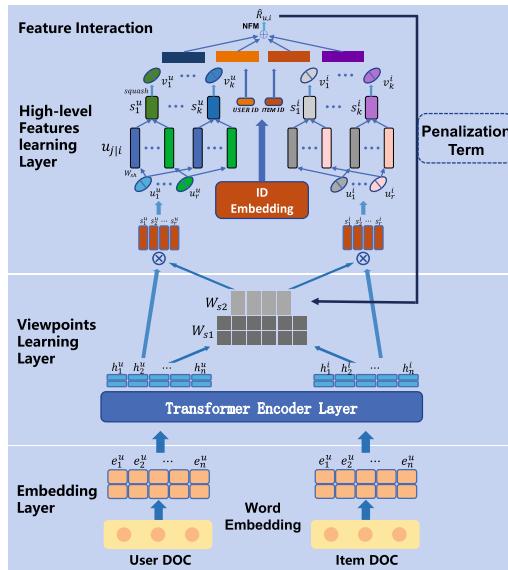


Fig. 1. The architecture of DPCN model in detail. w_{s1} and w_{s2} are trainable weight matrices. The \otimes denotes element-wise product, the \oplus denotes vector concatenation operation.

2.1 Contextual Viewpoints Learning Component

Users' Review Texts Encoder: Given $D^{u,i} = w_1, w_2, \dots, w_n$ denote the user or item review document, where n is the length of the document words, we

firstly map each word into its d-dimensional embedding representation: $E^{u,i} = e_1, e_2, \dots, e_n$ by an embedding layer.

Now each element in the sequence $E^{u,i}$ is independent with each other. To acquire some dependency between context words within a single sentence, we use a shared transformer encoder layer [14] to process the embedding texts and get the sentence-level hidden state representations h^u and h^i .

Diversity Viewpoints Learning Layer: The viewpoints learning layer focuses on the informative parts of the sentence, and it is expected to reflect aspects or significance information in the context. We take the whole hidden states as input, and output a vector of viewpoint distributed weight \mathbf{a} :

$$\mathbf{a} = \text{softmax}(v_{s2}\tanh(W_{s1}H^T)) \quad (1)$$

Here $v_{s2} \in \mathbb{R}^{1 \times d_a}$ and $W_{s1} \in \mathbb{W}^{d_a \times l}$ are the trainable weight vector and the trainable weight matrix respectively. The *softmax* ensures all the computed weight sum up to 1. The calculation result of the equation is a vector \mathbf{a} which represents tendencies distribution of the sentence, and in the next section we will use it as input for capsule networks.

Vector \mathbf{a} can indicate one aspect of one document text. However, there are multiple attributes in a sentence that compose a complete semantic space. We want to capture different information on content and need various weight vectors to get diversity information of the sentence. Thus we extends the v_{s2} vector to a $r \times d_a$ matrix, note it as W_{s2} , r denote the different focus points of the distributed representation, and the vector \mathbf{a} will extend to a matrix A :

$$A = \text{softmax}(W_{s2}\tanh(W_{s1}H^T)) \quad (2)$$

To avoid similar summation weights for all the weight vectors a_r in the matrix A , we utilize penalization term [6] to encourage the diversity of summation weight vector and force each vector to be focused on a single aspect, and the penalization term calculation formula is as follows:

$$P = \|AA^T - E\|_F^2 \quad (3)$$

Here $\|\cdot\|$ represents the Frobenius norm of a matrix, which ensure that the weight distribution vectors \mathbf{a}_r have no overlap with each other. E is an identity matrix. If two vectors \mathbf{a}_k and \mathbf{a}_t have the same distribution, their summation over elements $\mathbf{a}_k^T \mathbf{a}_t$ at corresponding position in penalization P will be a positive value, otherwise very small value. We will optimize P along with the original loss to obtain diverse focus to the contextual information.

Distribution weight matrix A is shared by both user document and item document. According to the distribution weight, the meaningful distribution of each word in the user and item documents can be computed as follows:

$$S = AH \quad (4)$$

Hence we obtain user and item distributed feature matrix, representing as $S^u = s_1^u, s_2^u, \dots, s_r^u$ and $S^i = s_1^i, s_2^i, \dots, s_r^i$.

2.2 High-Level Features Aggregation Component

As argued in paper [12], the structure of capsule networks makes it possible to model the complex latent features to generate high-level features. We abstract the process as learning the child-parent relationship.

In order to generate high-level parent vector v_j from child capsule u_i , we use an intermediate capsule $u_{j|i}$ and a share weight matrix $W_{sh} \in \mathbb{R}^{c \times d_c \times n}$, where n is the dim of input vector, c is the number of parent capsules in the layer above, and d_c is the dim of parent capsules. Thus the intermediate vector $\hat{u}_{j|i}$ will be calculated as:

$$\hat{u}_{j|i} = W_{sh}u_i + \hat{b}_{j|i} \quad (5)$$

Where u_i is a child capsule in the layer below and $\hat{b}_{j|i}$ is the bias term. In this paper, u_i is the user or item distributed feature vector calculating in Eq. 4, this means that input vector of capsule networks is $u_i = s_r \in \mathbb{R}^n$.

In order to ensure that every intermediate capsule map into appropriate output parent vector: $\hat{u}_{j|i} \rightarrow v_j$. A dynamic routing strategy is used to iteratively change the connection weight and detect whether a feature is present in any position of the feature distribution. The pseudo-code of dynamic routing algorithm is shown in Algorithm 1. b_{ij} is the logarithmic prior probability from the i^{th} capsule to the j^{th} capsule, and initialized to 0.

Algorithm 1. Dynamic routing algorithm

Input: $u_{j|i}; \tau$

Output: v_j ;

- 1: for all capsule i in layer l and capsule j in layer $l+1$: $b_{ij} \leftarrow 0$
 - 2: **for** τ iterations **do**
 - 3: for all capsule i in layer l : $c_{ij} \leftarrow \frac{\exp(b_{ij})}{\sum_j \exp(b_{ij})}$;
 - 4: for all capsule j in layer $l+1$: $s_j \leftarrow \sum_i c_{ij} u_{j|i}$;
 - 5: for all capsule j in layer $l+1$: $v_j \leftarrow \text{Squash}(s_j)$;
 - 6: for all capsule i in layer l and capsule j in layer $l+1$: $b_{ij} \leftarrow b_{ij} + u_{j|i} \dot{v}_j$;
 - 7: **end for**
 - 8: **return** v_j
-

In the routing procedure, b_{ij} is normalized by softmax to gain c_{ij} which represents the connection weight from capsule i to capsule j . Then we sum the products of all intermediate vectors $u_{j|i}$ and corresponding connection weights c_{ij} to get original parent capsule s_j . A squash function is used on original parent capsule s_j to get output parent capsule v_j . To adapt the squash function to our task, we modify the formula as:

$$v_j = \frac{\|s_j\|}{\sqrt{\|s_j\|^2 + 0.5}} \frac{s_j}{\|s_j\|} \quad (6)$$

The squash result \hat{v}_j is used to update logarithmic probability b_{ij} . After dynamic routing process, we gain output parent capsule v_j that denote high-level features compressed from latent feature capsules s_r .

2.3 Feature Interaction Component

We use one-hot encoded vector o^u and o^i to describe the user and item ID number respectively. Then the one-hot vector is mapped to low-dimensional factor vectors through the latent factor matrices Q^u and Q^i in the embedded layer, which expressed as $q^u = Q^u o^u$ and $q^i = Q^i o^i$.

User or item ID information can contain some inherent information to some extent, thus we combine the ID embedding with the output parent capsules to form a concatenated vector \mathbf{x} .

$$x = \delta[v^u \oplus q^u \oplus v^i \oplus q^i] \quad (7)$$

Where \oplus is concatenation operation; δ is the activation function. We feed $x \in \mathbb{R}^m$ into a Neural Factorization Machine [2] to predict rating:

$$\hat{r}_{u,i}(x) = \hat{w}_0 + \sum_{k=0}^{|x|} \hat{w}_k x_k + f(x) \quad (8)$$

In which \hat{w}_0 is the global bias term, $\sum_{k=0}^{|x|} \hat{w}_k x_k$ denotes the weight of feature and \hat{w}_k is the coefficient for latent feature vector. The third term $f(x)$ is used for feature interactions, and can be expressed as:

$$f(x) = \frac{1}{2} \left[\left(\sum_{k=1}^{|x|} x_j v_j \right)^2 - \left(\sum_{t=1}^{|x|} x_t v_t \right)^2 \right] \quad (9)$$

Where x_i represent the i^{th} feature value in vector x . $v_i \in \mathbb{R}^m$ is the embedding vector for i^{th} feature. Finally, we optimize parameters by useing mean squared error (MSE) as the loss function:

$$\mathcal{L} = \sum_{(u,i) \in (U,I)} (\hat{r}_{u,i} - r_{u,i})^2 + \lambda \|AA^T - E\|_F^2 \quad (10)$$

Where (U, I) denotes the set of user-item pairs in training data. λ is a constant. $\|AA^T - E\|_F^2$ is the Frobenius norm mentioned in Eqs. 3, which punishes redundancy problems if the diversity penalization mechanism provide similar vector for different inputs of the capsule networks.

Table 1. Statistics of the five datasets.

Datasets	Users	Item	Ratings	Words per user	Words per item	Density
Office Products	4,905	2,420	53,228	197.93	229.52	0.448%
Digital Music	5,540	3,568	64,666	216.21	266.51	0.327%
Video Games	24,303	10,672	231,577	188.79	260.60	0.089%
Tools Improvement	16,638	10,217	134,345	162.53	212.48	0.079%
Beauty	22,363	12,101	198,502	252.31	328.03	0.070%

3 Experiments

We conduct experiments on five benchmark datasets in different domains from Amazon Product Review corpus¹ for performance evaluation.

3.1 Experimental Settings

Datasets: Amazon review datasets has been widely used for product recommendation, and the following 5-core review subsets are used for evaluation: *Office Products*, *Digital Music*, *Video Games*, *Tools Improvement* and *Beauty*. For each dataset, we randomly split the user-item pairs into 80% training set, 10% validation set and 10% testing set, and for each document of user-item pairs, we amputate(pad) the long(short) review document to the same length. Note that the user-item pairs must have at least one interaction review in the training set, and for the test set, the interaction reviews are excluded. Table 1 summarizes the details of these experimental datasets.

Baselines: We compare our model with both conventional methods and state-of-art rating prediction methods:

- **PMF:** Probability Matrix Factorization [11], which makes use of rating between users and items to predict final scores via matrix factorization.
- **CMLE:** Collaborative Multi-Level Embedding model [18] integrates word embedding with standard matrix factorization model, which allows the model to have the ability to capture word local context information.
- **ConvMF:** Convolutional Matrix Factorization [3] utilizes CNN to extract latent features from review documents and makes prediction through PMF.
- **DeepCoNN:** Deep Cooperative Neural Networks [20] makes use of two CNN networks to extract latent feature from review documents and then uses FM to predict the rating.
- **D-attn:** Dual Attention CNN model [13] designs local and global attention to form review representations, and the dot product between the user and item embedding is used to gain the rating.
- **PARL:** This method proposes a method to exploit user-item pair-dependent features from auxiliary reviews written by like-minded users to address the data sparsity problem of recommender system [16].

¹ <http://jmcauley.ucsd.edu/data/amazon/>.

- **CARL:** Context-Aware User-Item Representation Learning [17], which uses attention mechanism and convolution operation for feature learning and factorization machine for high-order feature interactions.
- **CARP:** Capsule Networks Based Model For Rating Prediction [5] devises to extract the informative logic units from the reviews and infer their corresponding sentiments.
- **DAML:** Dual Attention Mutual Learning [8] utilizes local and mutual attention of the convolutional neural network to jointly learn the features of reviews.

Among these methods, PMF only utilizes the user-item rating for prediction. The other methods all make use of the review documents for extracting features and making predictions.

Hyper-Parameters Setting: We use the hyper-parameters reported in baseline method papers. The word embeddings are randomly initialized and the dimension size of word embedding is set to 300 (*i.e.*, $d = 300$). All neural methods are trained using Adam [4], applying a learning rate of 0.002, the batch size for all datasets are set to 200.

For DPCN, we use two transformer encoder layers with 8 heads, and the distribution weight matrix has 30 rows (*i.e.*, $r = 30$), which means we want them to focus on 30 overall aspects of the review documents. The penalization term coefficient λ is set to 0.3. And in the capsule networks part, the iteration number τ is set to 3 for each dynamic routing, the capsule nums are set to 16 and the dimension of each capsule is set to 32.

Evaluation Metric: The performance of the recommendation methods are evaluated by Mean Squared Error (MSE),

$$MSE = \frac{1}{|(U, I)|} \sum_{(u, i) \in (U, I)} (\hat{r}_{u,i} - r_{u,i})^2 \quad (11)$$

where (U, I) is the set of the user-item pairs in the test set.

3.2 Performance Evaluation

The overall performances of all the methods are reported in Table 2. Several observations can be made:

As shown in Table 2, DPCN consistently achieves the best performance across the five datasets. We can observe that the average improvement of DPCN against the best baseline is 2.91%, which indicates that learning diversity inter-viewpoint information yields a better understanding of customers' interests. Moreover, DPCN achieves 3.14% relative improvement on the sparsest dataset *Beauty*, which shows that the DPCN model can solve the sparse problem to a certain extent. The performance gap compared with other baseline methods validates DPCN model can capture more knowledge about customers' preferences and makes great prediction.

Table 2. The performance of different recommendation algorithms evaluated by MSE. The best results are in **boldfaces** and the second best results are underlined. $\Delta \%$ denotes the relative improvement of DPCN over the best baseline.

Method	Office products	Digital music	Video games	Tools improvement	Beauty
PMF	1.091	1.211	1.669	1.564	2.113
CMLE	0.761	0.883	1.254	1.023	1.508
ConvMF	0.960	1.084	1.449	1.240	1.601
DeepCoNN	0.860	1.060	1.238	1.063	1.498
D-attn	0.824	0.914	1.142	1.046	1.476
PARL	0.731	0.849	1.117	0.955	1.334
CARL	0.723	0.832	1.103	<u>0.941</u>	1.226
CAPR	<u>0.719</u>	0.820	<u>1.084</u>	0.960	1.243
DAML	0.728	<u>0.816</u>	1.112	0.943	<u>1.211</u>
DPCN	0.681	0.781	1.080	0.927	1.173
$\Delta \%$	5.28	4.28	0.37	1.49	3.14

Table 3. Model Performance on the subsets of discontinuous views with and without the penalization term.

Train on subset	Method	Office	Digital	Video	Tools	Beauty
<input checked="" type="checkbox"/>	With penalization	0.687	0.783	1.079	0.929	1.174
	Without penalization	0.713	0.798	1.088	0.945	1.181
	$\Delta \%$	3.65	1.88	0.83	1.69	0.60
<input type="checkbox"/>	With penalization	0.691	0.781	1.087	0.934	1.178
	Without penalization	0.723	0.803	1.106	0.967	1.197
	$\Delta \%$	4.42	2.74	1.71	3.41	1.59

Overall Score : 4/5

This Fender cable is the perfect length for me! Sometimes I find it a bit too long but I don't mind. The build quality is great and I know that it will last. The only gripe I have with this cable is that the metal sleeve gets unscrewed way too easily, requiring me to tighten it often. Sound quality is not affected, and the color is cool, definitely try this cable out.

Fig. 2. Review text contains multiple and non-consecutive viewpoints. For example, underline and bold part expressed different preferences of one user.

Table 4. Impact of different numbers of iterations in dynamic routing. The best results are highlighted in boldface.

τ	Office products	Digital music	Video games	Tools improvement	Beauty
1	0.693	0.792	1.093	0.938	1.182
2	0.690	0.785	1.086	0.931	1.174
3	0.681	0.781	1.080	0.927	1.173
4	0.687	0.780	1.082	0.926	1.174

3.3 Effectiveness of Penalization Term

In this section, we conduct experiments to explore the effectiveness of the penalization term in our recommendation model. Figure 2 shows the situation of reviews which contain discontinuous viewpoints. The reviewers may have some dissatisfaction with the commodity, but it is acceptable on the whole, so the reviewers will give ratings that are not very extreme (1/5 or 5/5).

We manually annotate datasets of non-consecutive reviews to reveal the penalization term has a contribution to capture diversity viewpoints. We select comments with scores between 2 and 4 and then judge whether they contain more than two viewpoints. After these steps, we get subsets of five benchmark datasets, each contain about 300 pieces of data. We fine-tune the model on each subset with and without the penalization term, Table 3 shows the MSE of two methods. The score of manual annotating subset is between 2 and 4, so the performance of the fine-tune model is declining compared to the original model. We compare the percentage of performance improvement, and can find that the model trained on the subset is more sensitive to expression fluctuations. The results indicate that the penalization term is helpful for learning discontinuous viewpoints.

3.4 Analysis of Capsule Networks

The Impact of Dynamic Routing: We present the results of performance with different iteration numbers τ in dynamic routing. As shown in Table 4. It is obvious to see that more than two iterations lead to performance improvement of our model, it justifies our assumption that the usefulness of input capsules are aggregated to form high-level features. The optimal iterations for learning latent features are 3 in general.

The Effectiveness of The Capsule Networks: We visualize the sparse pattern of the children capsules and parent capsules in the embedding space by applying t-SNE visualization [10]. As shown in Fig. 3, we evaluate DPCN model on one review document of *Tools Improvement* datasets. Extracting the vectors from the capsule networks and visualizing high-dimensional data by giving each datapoint a location in a two-dimensional map. It is obvious that the parent

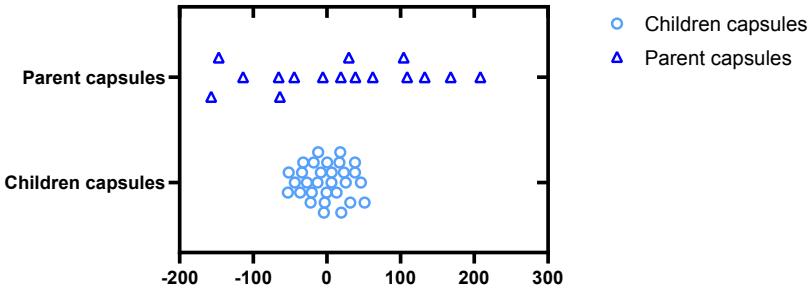


Fig. 3. t-SNE visualization for the children capsules and parent capsules on the Tools Improvement datasets.

capsules are more sparse than the child capsules, which indicates that the parent capsules can similarly abstract away from different surface realizations in the embedding space [19].

4 Conclusion

In this paper, we propose a CapsNets based model for predicting users' preferences by learning reviews. Specifically, the diversity penalization mechanism is applied to obtain meaningful distributed representations of individual viewpoints. The capsule networks are employed to aggregate the features which are consistent important for disparate viewpoints. We evaluate our model on five Amazon datasets, the experimental results show that our method achieves certain improvement over existing recommendation methods.

Acknowledgments. We thank the anonymous reviewers for their valuable comments. I would particularly like to acknowledge my supervisor and team members, for their wonderful collaboration and patient guidance.

References

1. Cheng, P., Wang, S., Ma, J., Sun, J., Xiong, H.: Learning to recommend accurate and diverse items. In: Proceedings of the 26th International Conference on World Wide Web, pp. 183–192 (2017)
2. He, X., Chua, T.S.: Neural factorization machines for sparse predictive analytics. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 355–364 (2017)
3. Kim, D., Park, C., Oh, J., Lee, S., Yu, H.: Convolutional matrix factorization for document context-aware recommendation. In: Proceedings of the 10th ACM Conference on Recommender Systems, pp. 233–240 (2016)
4. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)

5. Li, C., Quan, C., Peng, L., Qi, Y., Deng, Y., Wu, L.: A capsule network for recommendation and explaining what you like and dislike. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 275–284 (2019)
6. Lin, Z., et al.: A structured self-attentive sentence embedding. arXiv preprint [arXiv:1703.03130](https://arxiv.org/abs/1703.03130) (2017)
7. Ling, G., Lyu, M.R., King, I.: Ratings meet reviews, a combined approach to recommend. In: Proceedings of the 8th ACM Conference on Recommender Systems, pp. 105–112 (2014)
8. Liu, D., Li, J., Du, B., Chang, J., Gao, R.: DAML: dual attention mutual learning between ratings and reviews for item recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 344–352 (2019)
9. Liu, H., et al.: NRPA: neural recommendation with personalized attention. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1233–1236 (2019)
10. Maaten, L.V.D., Hinton, G.: Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**(Nov), 2579–2605 (2008)
11. Mnih, A., Salakhutdinov, R.R.: Probabilistic matrix factorization. In: Advances in Neural Information Processing Systems, pp. 1257–1264 (2008)
12. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: Advances in Neural Information Processing Systems, pp. 3856–3866 (2017)
13. Seo, S., Huang, J., Yang, H., Liu, Y.: Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In: Proceedings of the Eleventh ACM Conference on Recommender Systems, pp. 297–305 (2017)
14. Vaswani, A., et al.: Attention is all you need. arXiv (2017)
15. Wang, C., Blei, D.M.: Collaborative topic modeling for recommending scientific articles. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 448–456 (2011)
16. Wu, L., Quan, C., Li, C., Ji, D.: PARL: let strangers speak out what you like. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, pp. 677–686. Association for Computing Machinery, New York (2018). <https://doi.org/10.1145/3269206.3271695>
17. Wu, L., Quan, C., Li, C., Wang, Q., Zheng, B., Luo, X.: A context-aware user-item representation learning for item recommendation. *ACM Trans. Inf. Syst. (TOIS)* **37**(2), 1–29 (2019)
18. Zhang, W., Yuan, Q., Han, J., Wang, J.: Collaborative multi-level embedding learning from reviews for rating prediction. In: IJCAI, vol. 16, pp. 2986–2992 (2016)
19. Zhao, W., Peng, H., Eger, S., Cambria, E., Yang, M.: Towards scalable and reliable capsule networks for challenging NLP applications. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1549–1559 (2019)
20. Zheng, L., Noroozi, V., Yu, P.S.: Joint deep modeling of users and items using reviews for recommendation. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, pp. 425–434 (2017)



A Residual Dynamic Graph Convolutional Network for Multi-label Text Classification

Bingquan Wang, Jie Liu^(✉), Shaowei Chen, Xiao Ling, Shanpeng Wang,
Wenzheng Zhang, Liyi Chen, and Jiaxin Zhang

College of Artificial Intelligence, Nankai University, Tianjin, China
{wangbq,shaoweichen,1711524,1811579,wzzhang,liyichen,
nkuzjx}@mail.nankai.edu.cn, jliu@nankai.edu.cn

Abstract. Recent studies often utilize the Graph Convolutional Network (GCN) to learn label dependencies features for the multi-label text classification (MLTC) task. However, constructing the static label graph according to the pairwise co-occurrence from training datasets may degrade the generalizability of the model. In addition, GCN-based methods suffer from the problem of over-smoothing. To this end, we propose a Residual Dynamic Graph Convolutional Network Model (RDGCN) (<https://github.com/ilove-Moretz/RDGCN.git>) which adopts a label attention mechanism to learn the label-specific representations and then constructs a dynamic label graph for each given instance. Furthermore, we devise a residual connection to alleviate the over-smoothing problem. To verify the effectiveness of our model, we conduct comprehensive experiments on two benchmark datasets. The experimental results show the superiority of our proposed model.

Keywords: Graph Convolutional Network · Multi-label text classification · Dynamic label graph · Residual connection

1 Introduction

Multi-label text classification (MLTC) task, which aims to assign multiple non-exclusive labels to the given text, is a fundamental task in natural language processing. It plays a critical role in wide applications [2, 17].

Traditional methods [1, 3] generally encode the feature of the given text and adopt multiple binary classifiers to predict the corresponding labels, which neglects the correlations between different labels. To learn the label interactions, recent studies adopt the Graph Convolutional Network (GCN) [24], which constructs the label graph to learn semantic interactions between labels. These methods have achieved great progress due to the topology of label graphs treating labels as graph vertices and the feature aggregation ability of the graph.

Despite the progress, previous GCN-based methods are still limited. Most existing studies construct the static label graph according to the pairwise co-occurrence [14] from training datasets. However, there exists bias between the

static label graph and the specific text. As shown in Fig. 1, the label ‘Horror’ strongly correlates to the label ‘Thriller’, while it weakly correlates to the label ‘Romance’ in the static label graph constructed from the training datasets. However, the specific text describes a love story of the vampire whose label ‘Horror’ is co-occurrence with ‘Romance’ rather than ‘Thriller’. This bias between the static label graph and the specific text may make the model incorrectly predict the label ‘Thriller’ and miss the label ‘Romance’. To deal with the problem of bias between the static label graph and the specific text, a label attention mechanism is devised to capture the label-specific representations for each instance. Based on the above label representations, we further utilize the convolutional operation to construct a dynamic label graph, which reflects the unique label interactions for each instance. However, GCN-based methods may suffer from the problem of over-smoothing, which leads to features of graph vertices converging to the same value [11]. The decrease of the diversity between the features of labels will make the GCN-based model predict the labels incorrectly. Although decreasing layers of the label graph can alleviate the problem of over-smoothing, the shallow layers structure can not learn “high level” features of labels, which is not conducive to labels prediction. We introduce the residual connection [6] to the GCN, which can alleviate the problem of over-smoothing and learn the “high level” features of labels at the same time. Different from the work [6] which connects the input of the layer and the output of the layer, our model adopts the residual connection between the initial input of the first layer and the output of each layer. The initial input of the first layer is more diverse. Adding the initial input of the first layer to the output of each layer can alleviate the trend of features of labels to be the same value. To demonstrate the effectiveness of our model, we evaluate our model on two benchmark datasets. We show that: 1) The dynamic label graph performs better than the static label graph. 2) The residual connection can boost the performance of GCN on the MLTC task, and the proposed residual connection strategy in this paper outperforms the traditional residual connection raised by [6].

The main contributions of this paper can be summarized as follows:

- To learn the unique label interactions for each instance, we devise a label attention mechanism and construct a dynamic label graph to adequately learn the label-specific features.
- To alleviate the problem of over-smoothing and learn the “high level” features of labels on the MLTC task, we propose to adopt the residual connection to the GCN.
- Extensive experiments show that the dynamic label graph and the proposed residual connection can improve the classification performance on two public MLTC datasets.

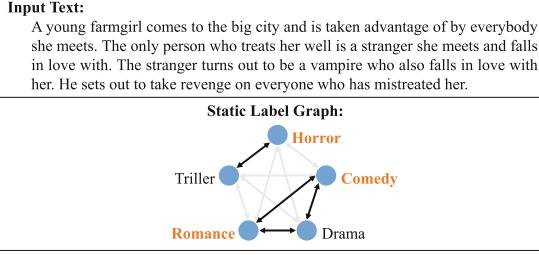


Fig. 1. An example of the static label graph. The black lines represent strong correlations between labels, while the gray lines represent weak correlations. And the standard labels of the input text are marked with orange. (Color figure online)

2 Related Work

Significant progress has been made on the MLTC task. Some studies [3–5] transform the MLTC task into a set of binary classification tasks, which ignores the label correlations.

To explore the label correlations, many transformation methods [1, 22] take advantage of correlations between labels. With the development of the neural networks, many studies are proposed based on CNN [10], RNN [16] and etc. They explore the “high-level” features from instances and label words. The work [21] adopts the attention mechanism to capture long-distance correlations among words and labels. However, none of these studies could learn structure information among labels.

To deal with this problem, some studies resort to graphical architectures. [8] constructs a label co-exist graph to extract the label correlation information. [23] establishes an explicit label graph to better explore the label space. Some studies adopt different graph models such as Graph Attention Network(GAT) [14], Graph Isomorphism Network (GIN) [25] and Graph Convolutional Network(GCN) [24] to capture the attentive dependency structure among the labels. Unfortunately, they can not work well because they neglect the problem of over-smoothing and construct the static label graph from statistics of training datasets which leads to the bias between the static label graph and the specific text. Recent work [11] introduces residual connection [6] used in the CNN model to the GCN on the task of point cloud semantic segmentation to alleviate the problem of over-smoothing. However, the traditional residual framework of [11] connecting the input of the layer and the output of the layer still has the problem. In the traditional residual, the input of the layer is the output of the previous layer which still has the trend of over-smoothing and the trend of over-smoothing will accumulate. To deal with this problem, we adopt the residual connection between the initial input of the first layer and the output of each layer. This residual connection strategy can alleviate the problem of accumulation of the trend of over-smoothing, which boosts the performance of the MLTC task.

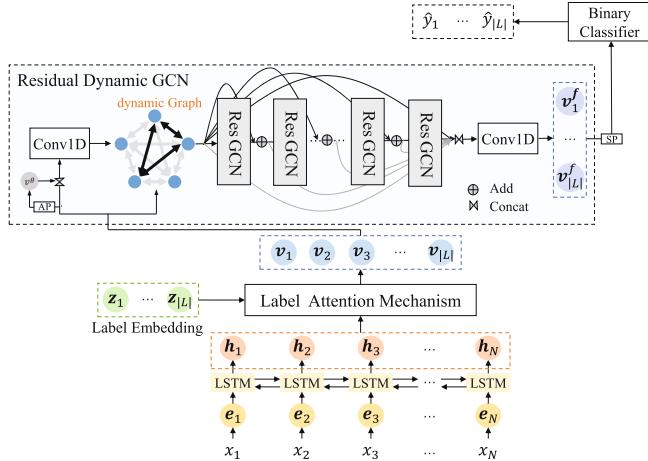


Fig. 2. The framework of our model.

3 Method

Given a sentence $X = \{x_1, x_2, \dots, x_N\}$ with N tokens, the MLTC task aims to predict its corresponding label set $Y = \{y_i\}_{i=1}^S$, where S is the number of labels. Note that each label y_i is selected from a predefined set $L = \{l_1, l_2, \dots, l_{|L|}\}$ with $|L|$ labels.

To deal with the MLTC task with GCN, three issues should be considered, including capturing the interactions between different labels, learning useful features, and alleviating the over-smoothing problem caused by deep layers GCN. To this end, we propose a residual dynamic graph convolutional networks model in this paper. The overall framework of our model is illustrated in Fig. 2. Specifically, our model consists of an encoding layer, a label attention mechanism, a residual dynamic GCN, and a label prediction layer. We first utilize the bidirectional long short-term memory network(BLSTM) [7] as the encoding layer to learn the contextual semantics of each token. Then, we devise a label attention mechanism to obtain the label-specific representations for each instance, which aims to highlight the important semantics corresponding to each label. Based on these representations, we further design a residual dynamic GCN to learn the features of labels. Finally, we use the binary classifier as the label prediction layer to assign the corresponding labels to each instance.

3.1 Encoding Layer

Given a sentence X , we first adopt the BLSTM to capture the contextual semantics for each token. Concretely, we use the word embedding as the initial representation $\mathbf{e}_i \in \mathbb{R}^{d_e}$ of each token and feed the initial representation sequence $E = \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N\}$ into BLSTM to encode the contextual representation sequence $H = \{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_N\}, \mathbf{h}_i \in \mathbb{R}^{d_h}$:

$$\mathbf{h}_i = \left[\overrightarrow{\mathbf{h}}_i; \overleftarrow{\mathbf{h}}_i \right], \quad \overrightarrow{\mathbf{h}}_i = \overrightarrow{\text{LSTM}} \left(\mathbf{e}_i, \overrightarrow{\mathbf{h}}_{i-1} \right), \quad \overleftarrow{\mathbf{h}}_i = \overleftarrow{\text{LSTM}} \left(\mathbf{e}_i, \overleftarrow{\mathbf{h}}_{i+1} \right), \quad (1)$$

where $\overrightarrow{\text{LSTM}}$ and $\overleftarrow{\text{LSTM}}$ are the forward LSTM and the backward LSTM, respectively. And d_h is the dimension of the hidden representations.

3.2 Label Attention Mechanism

We couple a label attention mechanism upon the encoding layer to capture the specific semantics of each label according to the given text. Specifically, we first initialize the label embeddings $Z = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_{|L|}\}$, $\mathbf{z}_i \in \mathbb{R}^{d_z}$ with the random values, where d_z represents the dimension of the label embedding. Then, we project the hidden representations of tokens and the label embeddings into the key $K \in \mathbb{R}^{N \times d_l}$ and the query $Q \in \mathbb{R}^{|L| \times d_l}$ as follows:

$$K = HW_K, \quad Q = ZW_Q, \quad (2)$$

where $W_K \in \mathbb{R}^{d_h \times d_l}$ and $W_Q \in \mathbb{R}^{d_z \times d_l}$ are the trainable parameters. And d_l denotes the dimension of the attention mechanism.

Based on these, We obtain the label-specific representations V for each instance as follows:

$$V = \text{softmax}(QK^T)H, \quad (3)$$

where $V = \{v_1, v_2, \dots, v_{|L|}\}$, $v_i \in \mathbb{R}^{d_h}$, v_i denotes the feature of the i -th label.

3.3 Residual Dynamic GCN

Construct Dynamic GCN. For the multi-label classification task, it is crucial to capture and utilize the interactions between different labels. Thus, we adopt the label graph to represent the correlations between labels, which treats the features of graph vertices as the features of labels. Previous studies usually construct the correlation matrix of the static label graph from statistics of training datasets which causes the frequency-bias problem. In this paper, we adopt the dynamic label graph. For each instance, we construct the correlation matrix $A^d \in \mathbb{R}^{L \times L}$ of the dynamic label graph using the label-specific representations V obtained in the previous section.

$$A^d = \sigma(\text{Conv1}(V')), \quad (4)$$

where $\sigma(\cdot)$ is the sigmoid activation function, $\text{Conv1}(\cdot)$ is the 1-D convolutional neural network, $V' \in \mathbb{R}^{|L| \times (2*d_h)}$ is obtained by concatenating V and $v^g \in \mathbb{R}^{d_h}$, which is obtained by Average Pooling(AP) on the V . Formally, V' is defined as:

$$V' = [(v_1; v^g), (v_2; v^g), \dots, (v_{|L|}; v^g)]. \quad (5)$$

The general graph convolution operation at the l -th layer can be formulated as:

$$V^{l+1} = f(A^d V^l W_d), \quad (6)$$

where $f(\cdot)$ is the LeakyReLU activation function, $V^l = \{v_1^l, v_2^l, \dots, v_{|L|}^l\}$, $v_i^l \in \mathbb{R}^{d_h}$ is the input of the l -th layer. V^{l+1} is the output of the l -th layer and the input of $l+1$ -th layer, W_d is the learnable weights of GCN. V is the initial input of the first layer.

Residual Connections in GCN. To reduce the accumulation of the trend of over-smoothing, we propose a new residual connection. As shown in Fig. 2, the initial input V of the first layer is added to the output of each layer. We update the graph convolution operation as:

$$V_{res}^{l+1} = f(A^d V_{res}^l W_d) + V, \quad (7)$$

where V_{res}^{l+1} is the output of l -th layer after adding a residual connection. After the last layer, we fuse the output of each layer. We first concatenate the output of each layer as follows:

$$V^{concat} = [V_{res}^1 : V_{res}^2 : \dots : V_{res}^{k+1}], \quad (8)$$

where, k is the number of layers, $V^{concat} \in \mathbb{R}^{|L| \times ((k+1)*d_h)}$. Then, we utilize 1-D convolutional neural network to fuse V^{concat} to the final output V^f as follows:

$$V^f = f(Conv1(V^{concat})), \quad (9)$$

where, $f(\cdot)$ is the LeakyReLU activation function, $Conv1(\cdot)$ is the 1-D convolutional neural network, $V^f = \{v_1^f, v_2^f, \dots, v_{|L|}^f\}, v_i^f \in \mathbb{R}^{d_h}$.

3.4 Label Prediction

To accurately predict the multiple labels for the given sentence, we formulate the MLTC task to a binary classification task on each label. Specifically, we use the Sum-Pooling(SP) operation to distill the final output of the residual GCN and predict the probability of each label \hat{y}_i as follows:

$$\hat{y}_i = \sigma(SP(v_i^f)), \quad (10)$$

where v_i^f is the final output of the i -th label, $i \in \{1, \dots, |L|\}$, and $\sigma(\cdot)$ is the sigmoid function.

During training, we minimize the binary cross-entropy loss function as follows:

$$\mathcal{L} = - \sum_{j=1}^M \sum_{i=1}^{|L|} (y_{ij} \log(\hat{y}_{ij})) + (1 - y_{ij}) \log(1 - \hat{y}_{ij}), \quad (11)$$

where M is the number of training instances, $y_{ij} \in \{0, 1\}$ denotes the ground truth, and \hat{y}_{ij} is the final prediction. Note that $y_{ij} = 1$ represents that the i -th label appears in the j -th instance, and vice versa.

During testing, the i -th label is predicted for the j -th instance if \hat{y}_{ij} is higher than the given threshold β .

Table 1. Statistics of datasets. “Samples” and “Labels” denote the total number of samples and labels, respectively. “W/S” is the average number of words per sample and “L/S” represents the average number of labels per sample.

Dataset	Samples	Labels	W/S	L/S
AAPD	55840	54	150.72	2.41
SLASHDOT	3782	22	31.70	1.18

4 Experiments

In this section, we first introduce the datasets, evaluation metrics, baselines, and experimental details. Then we compare our model with the baselines. Finally, we conduct extensive ablation studies and analyze experimental results.

4.1 Datasets

To evaluate the performance of our model, we conduct experiments on two datasets. The dataset statistics are summarized in Table 1.

AAPD: This dataset is provided by [20]. AAPD consists of the abstracts and corresponding subjects of 55840 papers in the field of computer science. There are 54 subjects in total, and each paper may be related to multiple subjects.

SLASHDOT: This dataset is provided by MEKA¹, a multi-label extension to WEKA. It is a collection of news about science and technology from the Slashdot website, containing 3782 pieces of news with 22 subjects.

4.2 Evaluation Metrics

We use the Micro Precision, Recall, and F1-score as main evaluation metrics.

Micro Precision and Recall: For binary classification tasks on the single label, Precision represents the fraction of positive instances among the instances predicted as positive, and Recall represents the fraction of successfully predicted positive instances among all positive instances. Micro Precision and Recall are used in multi-label tasks, which calculate Precision and Recall on all labels.

Micro F1-score: Micro F1-score [13] is used to evaluate the quality of multi-label binary models, which measures the F1-score of the aggregated contributions of all labels. It is the harmonic mean of Micro Precision and Recall.

4.3 Baselines

To demonstrate the effectiveness of our model, we adopt the following baselines.

LP: [18] transforms a multi-label problem to a multi-class problem with one multi-class classifier trained on all unique label combinations.

XML-CNN: [12] uses Convolutional Neural Network(CNN) and adopts a dynamic max pooling scheme to capture fine-grained features of the texts.

¹ <https://sourceforge.net/projects/meka/>.

Table 2. The mean value and standard deviation of the results on two benchmark datasets (%). The best results are marked as bold.

	SLASHDOT			AAPD		
	Precision	Recall	F ₁ -Score	Precision	Recall	F ₁ -Score
LP	52.59	53.00	55.97	52.79	55.97	54.82
XML-CNN	56.91(±1.43)	37.4(±1.43)	45.12(±1.10)	67.67(±1.3)	57.16(±0.67)	61.96(±0.37)
LSAN	55.85(±2.48)	45.56(±4.90)	50.02(±2.75)	72.71(±0.74)	63.96(±1.39)	68.05 (+0.94)
LAHA	65.20(±3.22)	52.18(±1.32)	57.92(±1.15)	75.73(±0.71)	64.16(±2.16)	69.44(±1.09)
Ours	65.36(±2.59)	61.18(±2.03)	63.15(±1.09)	77.89(±2.57)	65.86(±2.03)	71.32(±0.63)

LSAN: [19] adopts a label-specific attention network to extract semantic relations between labels and documents, identifying the label-specific document representation.

LAHA: [8] uses a hybrid attention deep neural network model to extract semantic relations between documents and labels with a label co-exist graph, to establish an explicit label-aware representation for each document.

4.4 Experimental Setting

Each instance is truncated at the length of 500. We initialize the word embedding with pre-trained Glove 840B vectors [15]. The dimension of hidden states for LSTM and label embeddings is 200. The dimension of key and query is 50. During training, we use the Adam [9] optimizer. The learning rate, dropout, and batch size are set to 0.001, 0.5, and 32, respectively. We set β to 0.5. To obtain an estimate of the generalization performance of each method, we train each neural network model five times using random seed 1000, 1001, 1002, 1003, 1004. And we show the mean value and standard deviation of the results. LP is not the neural network model, we only report the best result. We conduct experiments with a number of different GCN layers and obtain the best results with three layers of GCN on the AAPD and SLASHDOT.

4.5 Results

The experimental results are shown in Table 2, where “Ours” represents our model RDGCN.

LSAN utilizes an attention mechanism to detect the semantic connections between labels and instances. So LSAN outperforms XML-CNN on all datasets. However, it only focuses on the simple interactions between labels and content. Based on LSAN, LAHA adopts the graph to capture the structure information between labels, which makes the model achieve higher scores. Our model adopts residual connection to GCN to alleviate the problem of over-smoothing and learn “high level” structure information between labels. Our model also constructs a dynamic label graph to alleviate the bias between the label graph and the specific text. So the effectiveness of our model is significantly improved on

Table 3. The mean value and standard deviation of the ablation results(%). The best results are marked as bold.

	SLASHDOT			AAPD		
	Precision	Recall	F ₁ -Score	Precision	Recall	F ₁ -Score
w/o res	64.94(±2.70)	60.04(±2.73)	62.31(±0.88)	77.35(±1.93)	64.13(±1.03)	70.10(±0.53)
w/o dynamic	63.18(±2.45)	58.33(±3.54)	60.55(±1.25)	77.47(±1.16)	64.73(±0.86)	70.52(±0.32)
Ours	65.36 (±2.59)	61.18 (±2.03)	63.15 (±1.09)	77.89 (±2.57)	65.86 (±2.03)	71.32 (±0.63)

all datasets in F1-score, precision, and recall. Compared with baselines, the proposed model achieves 0.16% precision, 9% Recall, 5.23% F1-score improvements on the SLASHDOT dataset and 2.16% precision, 1.7% recall, 1.88% F1-score improvements on the AAPD dataset.

4.6 Ablation Study

We try to demonstrate the effectiveness of the residual connection via the ablation experiment. As shown in Table 3, “Ours” represents the proposed model RDGCN, “w/o res” represents our model without residual connection. As expected, “w/o res” performs poorly, which verifies the effectiveness of residual connection. The experimental results of “Ours” gets 0.42% precision, 1.14% recall, 0.84% F1-score improvements on the SLASHDOT dataset and 0.54% precision, 1.73% recall, 1.22% F1-score improvements on the AAPD dataset.

To demonstrate that constructing the dynamic label graph boosts the classification performance of our model, we compare with experimental results using the static label graph. The correlation matrix of the static label graph is constructed according to the pairwise co-occurrence from training datasets [14]. We replace the dynamic label graph of our model with a static label graph, which is represented by “w/o dynamic” in Table 3. Compared with “w/o dynamic”, our model shows the improvement over “w/o dynamic”, which demonstrates that the dynamic label graph can capture the associations between different labels in a specific text by automatically learning the dynamic correlation matrix. The experimental results of “Ours” gets 2.18% precision, 2.85% recall, 2.6% F1-score improvements on the SLASHDOT dataset and 0.42% precision, 1.13% recall, 0.8% F1-score improvements on the AAPD dataset.

4.7 Comparison with the Traditional Residual Connection

We also conduct the experiments using the traditional residual connection from [11] which is used in the CV task. In Fig. 3, “Ours” represents the proposed model RDGCN, “TradRes” represents the model using the traditional residual connection. For the “TradRes”, we also experiment with different GCN layers. Consistent with our model, “TradRes” obtains the best experimental results with the three-layer GCN on both two datasets.

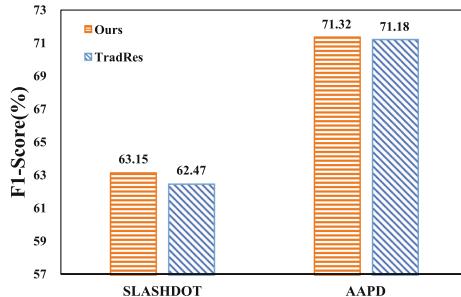


Fig. 3. Comparison of our model and the traditional residual connection method.

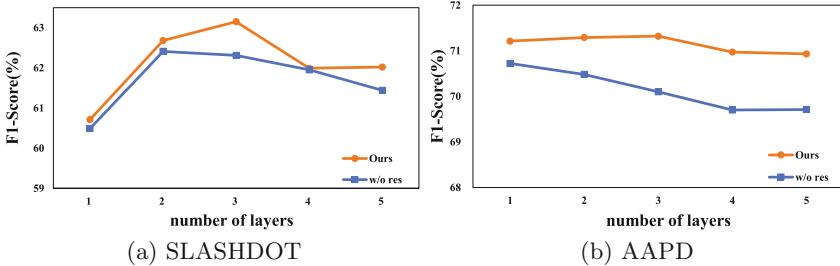
As shown in Fig. 3 the residual connection we proposed performs better than the traditional method on two datasets. The traditional residual connection connects the input of the layer and the output of the layer, while our model adopts the residual connection between the initial input of the first layer and the output of each layer. It demonstrates that connecting the initial input of the first layer and the output of each layer can reduce the accumulation of the trend of over-smoothing, which can boosts the performance of the model.

4.8 Analysis of Different GCN Layers

In Fig. 4, we also report the mean value of the F1-score of different GCN layers. “Ours” represents our model RDGCN. “w/o res” represents our model without residual connection.

As the number of layers increases, the F1-score of “Ours” first increases and then decreases on both two datasets. And when the number of GCN layers is three, our model achieves the highest F1-score. The results of “w/o res” show a similar trend, first increases and then decreases, on the SLASHDOT dataset. The difference is that “w/o res” obtains the best performance on the SLASHDOT dataset based on the two-layer GCN. On the AAPD dataset, “w/o res” achieves the best result using one layer GCN, and as the number of layers increases, the overall trend decreases.

We can observe that the result of “Ours” is always better than “w/o res” using any number of GCN layers on both two datasets. The best results of “Ours” are 0.74% (F1-score) and 0.58% (F1-score) higher than the best results of “w/o res” on the SLASHDOT and AAPD datasets, respectively. It demonstrates that utilizing residual connection is better than utilizing the GCN with the shallow layers structure since the deeper GCN layers can learn “high level” label features. In addition, the results of “Ours” can perform well on deeper GCN layers due to the residual connection, which demonstrates that the residual connection can alleviate the problem of over-smoothing and boost the performance on the MLTC task.

**Fig. 4.** Comparison of different GCN layers.

5 Conclusion

In this paper, we focused on the MLTC task and proposed a residual dynamic GCN model to learn “high level” features of labels caused by multi-layers GCN. We devised a label attention mechanism to capture the label-specific features for each instance. Based on these features, a dynamic label graph was designed to alleviate the bias between the static label graph and the specific text. The residual connection was proposed to alleviate the problem of over-smoothing and learn the “high level” label features. The extensive experiments demonstrated that our model achieves significant improvements.

Acknowledgement. This research is supported by the National Natural Science Foundation of China under the grant No. 61976119 and the Natural Science Foundation of Tianjin under the grant No. 18ZXZNGX00310.

References

1. Alvares-Cherman, E., Metz, J., Monard, M.C.: Incorporating label dependency into the binary relevance framework for multi-label classification. *ESWA* **39**(2) (2012)
2. Kumar, A., et al.: Ask me anything: dynamic memory networks for natural language processing. *arXiv Pre-Print* 97 (2015)
3. Boutell, M.R., Luo, J., Shen, X., Brown, C.M.: Learning multi-label scene classification. *Pattern Recognit.* **37**(9), 1757–1771 (2004)
4. Elisseeff, A., Weston, J., et al.: A kernel method for multi-labelled classification. In: *NIPS*, vol. 14, pp. 681–687 (2001)
5. Gopal, S., Yang, Y.: Multilabel classification with meta-level features. In: Crestani, F., Marchand-Maillet, S., Chen, H., Efthimiadis, E.N., Savoy, J. (eds.) *SIGIR*, pp. 315–322. ACM (2010)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *CVPR* (2016)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
8. Huang, X., Chen, B., Xiao, L., Jing, L.: Label-aware document representation via hybrid attention for extreme multi-label text classification. *arXiv preprint arXiv:1905.10070* (2019)

9. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015, Conference Track Proceedings (2015). <http://arxiv.org/abs/1412.6980>
10. Kurata, G., Xiang, B., Zhou, B.: Improved neural network-based multi-label classification with better initialization leveraging label co-occurrence. In: NAACL, June 2016
11. Li, G., Muller, M., Thabet, A., Ghanem, B.: DeepGCNs: can GCNs go as deep as CNNs? In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9267–9276 (2019)
12. Liu, J., Chang, W., Wu, Y., Yang, Y.: Deep learning for extreme multi-label text classification. In: Kando, N., Sakai, T., Joho, H., Li, H., de Vries, A.P., White, R.W. (eds.) Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, 7–11 August 2017, pp. 115–124. ACM (2017). <https://doi.org/10.1145/3077136.3080834>
13. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, New York (2008)
14. Pal, A., Selvakumar, M., Sankarasubbu, M.: MAGNET: multi-label text classification using attention-based graph neural network. In: ICAART (2), pp. 494–505 (2020)
15. Pennington, J., Socher, R., Manning, C.: GloVe: global vectors for word representation. In: EMNLP, October 2014
16. Qin, K., Li, C., Pavlu, V., Aslam, J.: Adapting RNN sequence prediction model to multi-label set prediction. In: ACL, pp. 3181–3190 (2019)
17. Schapire, R.E., Singer, Y.: BoosTexter: a boosting-based system for text categorization. *Mach. Learn.* **39**(2–3), 135–168 (2000)
18. Tsoumakas, G., Katakis, I.: Multi-label classification: an overview. *Int. J. Data Warehousing Mining (IJDWM)* **3**(3), 1–13 (2007)
19. Xiao, L., Huang, X., Chen, B., Jing, L.: Label-specific document representation for multi-label text classification. In: EMNLP-IJCNLP, November 2019
20. Yang, P., Sun, X., Li, W., Ma, S., Wu, W., Wang, H.: SGM: sequence generation model for multi-label classification. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 3915–3926. Association for Computational Linguistics, Santa Fe, August 2018. <https://www.aclweb.org/anthology/C18-1330>
21. You, R., Dai, S., Zhang, Z., Mamitsuka, H., Zhu, S.: AttentionXML: extreme multi-label text classification with multi-label attention based recurrent neural networks (2018)
22. Zhang, M.L., Zhou, Z.H.: ML-KNN: a lazy learning approach to multi-label learning. *Pattern Recogn.* **40**(7), 2038–2048 (2007)
23. Zhang, W., Yan, J., Wang, X., Zha, H.: Deep extreme multi-label learning. In: Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval, pp. 100–107 (2018)
24. Zhou, J., et al.: Hierarchy-aware global model for hierarchical text classification. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 1106–1117 (2020)
25. Zong, D., Sun, S.: GNN-XML: graph neural networks for extreme multi-label text classification. arXiv preprint [arXiv:2012.05860](https://arxiv.org/abs/2012.05860) (2020)



Sentence Ordering by Context-Enhanced Pairwise Comparison

Haowei Du, Jizhi Tang, and Dongyan Zhao^(✉)

Peking University, Beijing, China

2001213236@stu.pku.edu.cn, {tangjizhi,zhaodongyan}@pku.edu.cn

Abstract. Sentence ordering is a task arranging the given unordered text into the correct order. A feasible approach is to use neural networks to predict the relative order of all sentence pairs and then organize the sentences into a coherent paragraph with topological sort. However, current methods rarely utilize the context information, which is essential for deciding the relative order of the sentence pair. Based on this observation, we propose an efficient approach context-enhanced pairwise comparison network (CPCN) that leverages both the context and sentence pair information in a post-fusion manner to order a sentence pair. To obtain the paragraph context embedding, CPCN first utilizes BERT to encode all sentences, then aggregates them using a Transformer followed by an average pooling layer. Finally, CPCN predicts the relative order of the sentence pair by the concatenation of the paragraph embedding and the sentence pair embedding. Our experiments on three benchmark datasets, SIND, NIPS and AAN show that our model outperforms all the existing models significantly and achieves a new state-of-the-art performance, which demonstrates the effectiveness of incorporating context information.

Keywords: Sentence ordering · Context information · Pairwise comparison

1 Introduction

Text coherence is essential in natural language processing(NLP) and coherence makes it much easier for us to read and understand the text. Sentence ordering is a task that aims at arranging the unordered text into the right order, which maintains the coherence of the text. Sentence ordering plays a significant role in down stream applications such as concept-to-text generation [12, 13], multi-document summarizing [1, 20], storytelling [8, 11], recipe generation [4].

Recent work uses neural network to model the coherence of text and predict the order of the text. These approaches can be classified into two kinds: sequence generating model and pairwise model. Sequence generating model uses neural network to model the probability of the next sentence depending on the previous predicted sentences [6, 9, 14, 19, 23, 25]. However, if one sentence is placed in the

wrong position, the sentences after will be placed unreasonably based on the wrong previous prediction. Pairwise models utilize local information(sentence pair information) to predict the relative order of two sentences with a classifier. Then these models organize the text with the relative order by maximizing the scores of all sentence pairs in the permutation or topological sort [5, 18, 21]. However, these models do not utilize the global information(context information) to infer the relative order of two sentences. In some time it is hard to determine the relative order without the context (Table 1).

Table 1. An example of sentences in SIND dataset and the order is on the left

Order	Sentences
(1)	We set up our chair for the softball game
(2)	They were getting ready to play. The ball was thrown
(3)	She ran to catch the ball
(4)	Our team scored a point
(5)	We enjoyed relaxing and watching the game

We take an example in SIND dataset, which is one benchmark dataset of sentence ordering task. We denote the sentence i by s_i . Without the context, we do not know the relation of s_3 and s_5 and it is hard to determine the relative order of this pair. With s_1 we can know the “game” in s_5 refers to the softball game. With s_2 we know the “ball” in s_3 refers to the ball in this softball game. With s_4 we know the action of catching the ball in s_3 results in scoring a point and scoring a point makes the audience enjoyed and relaxed in s_5 . So s_5 is the result of s_3 and the relative order is easy to decide. In this example, the global information is the information in s_1 , s_2 and s_4 . It gives the scene and the process of the event, which is essential to predict the relative order of s_3 and s_5 .

To incorporate both the global and local information of the text, we propose a new and simple approach named context-enhanced pairwise comparison network (CPCN). CPCN utilizes both global and local information through post-fusion to predict the order of a sentence pair. To obtain local information, we follow B-Tsort [21] to leverage BERT to get a single representation for each sentence pair in the text. To obtain global information, we use BERT to encode each sentence. Then we input the embeddings of all the sentences in the text to a Transformer Network. The self-attention layers in Transformer Network can help the information sharing among sentences in the paragraph. Then we fuse all the sentence embeddings by an average pooling layer to get the global information of the context. We concatenate the two levels of information as an input of the MLP to predict the relative order of sentence pair. With the relative order of all pairs of sentences, we use topological sort to organize sentences into the right order. By the experiments of three benchmarks in the field of sentence ordering, SIND caption dataset, NIPS abstract dataset and AAN abstract dataset, our model outperforms all the existing models, becoming the new state of the art.

To summarize, our contributions are in two-fold: 1) we propose a new model CPCN, which combines global and local information in a post-fusion manner. 2) we conduct extensive experiments. The results show that CPCN outperforms existing methods, and clearly demonstrates the effectiveness of incorporating context information into pairwise ordering approaches.

2 Related Work

Early work on sentence ordering uses transition probabilistic model based on linguistic features [15], content models based on Hidden Markov Models [3], and entity based approach [2]. Recent work uses neural network to model the coherence of text and predict the order of the text. We review two kinds of neural network approaches which are the most prevalent and efficient.

2.1 Sequence Generating Models

This kind of approaches uses neural network to predict sentence locations as a sequence and treat this as a seq2seq problem. Many researchers put efforts into getting a more efficient encoder and decoder of sentences and the paragraph. Gong et al. (2016) [9] use pointer network as a decoder to sequentially choose the next sentence with the embeddings produced by the encoder. To connect information in different sentences, Cui et al. (2018) [6] introduce self-attention into the paragraph encoder. To model the connection between entities and sentences and between sentences that share same entities, Yin et al. (2019) [24] refine the encoder with an entity transition graph. Kumar et al. (2020) [14] find applying BERT as a sentence encoder and feed forward neural network as a decoder can improve the performance. Moreover, Yin et al. (2020) [23] find adding supplementary loss functions during the training process is also helpful. To model the connection between sentences at different distances, Yutao Zhu et al. (2021) [25] employ multiple GNNs and fuse them with a MLP network. However, in this category of approach, the prediction of one sentence is dependent on the sentences that have been predicted.

2.2 Pairwise Models

This kind of models aims at improving the method to predict the relative order of sentence pairs. Chen, Qiu, and Huang (2016) [5] encode the sentence with CNN and LSTM. Then they infer the relative order with representations of sentences by use of feed forward neural network. They maximize the sum of the scores of all sentence pairs in the candidate order to predict the text order. Prabhumoye, Salakhutdinov, and Black (2020) [21] utilize the next sentence prediction objective of BERT to encode the sentence pair and arrange the sentences into text with topological sort. This kind of approach fails to consider the global context and the relative order of some sentence pairs can not be decided without the other sentences. In our model, information in other sentences is connected to the sentence pair by use of Transformer Network to solve this problem.

3 Task Description

Given an unordered document consisted of n sentences, $\mathcal{D} = \{s_{o_1}, \dots, s_{o_n}\}$, where the random order is $\mathbf{o} = [o_1, \dots, o_n]$. Our task aims to find the right order of the sentences $\mathbf{o}^* = [o_1^*, \dots, o_n^*]$, such that

$$P(s_{o_1^*}, \dots, s_{o_n^*} | \mathcal{D}) \geq P(s_{o_1}, \dots, s_{o_n} | \mathcal{D}) \quad \text{for any permutation of } 1, 2 \dots n \text{ } \mathbf{o} \quad (1)$$

where $P(s_{o_1}, \dots, s_{o_n} | \mathcal{D})$ means the probability of $[o_1, o_2, \dots, o_n]$ to be the right order of the sentence set \mathcal{D} .

4 Methodology

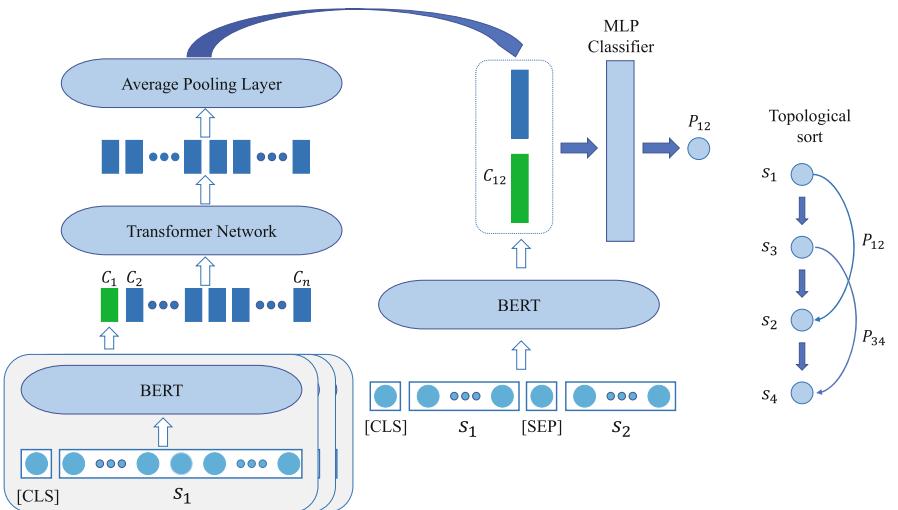


Fig. 1. Model overview

4.1 Overview

Our model includes 2 phases, where the first phase is context-enhanced pairwise ordering network and the second phase is topological sort. Context-enhanced pairwise ordering network contains 3 parts, global information encoder, local information encoder and post-fusion. The model pipeline is illustrated by Fig. 1.

4.2 Global Information Encoder

As BERT [7] is a powerful pretrained language model that can learn the token and sentence level information, we use BERT to encode each sentence in the paragraph. The input to the BERT model is a “[CLS]” token, followed by the sentence to encode and we use the hidden state corresponding to “[CLS]” as the encoding of the sentence. Next, we pass the embeddings of all sentences to a Transformer Network with self-attention layers. To handle the unordered nature of the input, we remove the position embedding layer. Based on the encoding for each sentence, we use an average pooling layer to encode the whole paragraph for the global information. The self-attention mechanism can connect different sentences which allows information sharing among sentences in a paragraph.

4.3 Local Information Encoder

For each pair of sentences in the paragraph, we leverage the next sentence prediction objective of BERT to encode the sentence pair. Sentence pair are packed together and separated by “[SEP]”, then we add “[CLS]” to the head to form the complete input sequence of the BERT model. We use the hidden state corresponding to “[CLS]” as the encoding of the sentence pair.

4.4 Post-fusion

For each sentence pair, We concatenate the encoding of the sentence pair and the paragraph as a input to a classifier to predict the relative order of the sentence pair. Let v_{ij} denote the encoding of the sentence pair and v_p denote the encoding of the paragraph that contains the sentence pair. Then

$$\mathbf{p}_{ij} = \text{MLP}([\mathbf{v}_{ij}; \mathbf{v}_p]) \quad (2)$$

where \mathbf{p}_{ij} denotes the relative order of the sentence pair.

4.5 Organizing into Text

With the relative order of the sentence pair, we can utilize different approaches to organizing the sentence into the text, such as maximizing the sum of probability of the relative order between the sentence pairs in the permutation and topological sort. In our experiment, we use topological sort. Topological sort is a standard algorithm to order vertices of directed graphs in linear time complexity. For each edge from u to v , u is placed before v in the order that topological sort produces. In our case, each sentence composes a vertex and each pair of vertices has an edge. The direction of the edge between two sentences is decided by the relative order that we have predicted before. Based on the relative order of all sentence pairs, we organize the full paragraph by use of topological sort.

5 Experiments

We conduct experiments on three standard public sentence ordering datasets, SIND dataset, NIPS dataset and AAN dataset. These datasets are commonly used in previous sentence ordering work [6, 14, 21, 25]. We compare three kinds of baseline models with our approach. In the experiment, we apply our CPCN with topological sort. Following Prabhumoye, Salakhutdinov, and Black [21], we name the whole method C-TSort(C stands for CPCN).

Table 2. The statistic of datasets

Datasets	Min	Max	Avg	Train	Dev	Test
NIPS	1	15	6	2448	409	402
AAN	1	20	5	8569	962	2626
SIND	5	5	5	40155	4990	5055

5.1 Dataset

The statistics of the datasets is shown in Table 2 where MIN, MAX, AVG mean the maximum, minimum and average number of sentences in a data instance.

NIPS [19]: This dataset contains abstracts from NeurIPS papers and the average number of sentences in a paragraph is 6. It is splitted into training set, dev set and test set by the publishing year.

AAN [19]: This dataset contains abstracts from ACL papers and the average number of sentences in a paragraph is 5. It is splitted into training set, dev set and test set by the publishing year.

SIND [10]: In this dataset, each paragraph has 5 sentences and corresponds to the caption of an image.

5.2 Baselines

Traditional Methods: Traditional methods use linguistic features or neural network to maximize coherence of text. We use Entity Grid [2] and Window Network [17] as baselines.

Pairwise Models: B-TSort [21] utilizes BERT to predict the relative order of sentence pair and organize into the text with topological sort. This is the current state-of-the-art.

Sequence Generation Models: This kind of models treats the task as a sequence generating problem. We select the typical methods below. CNN/LSTM+PtrNet [9]; Variant-LSTM+PtrNet [19]; ATTOrderNet [6]; HierarchicalATTNet [22]; SE-Graph [24]; ATTOrderNet+TwoLoss [23]; Rank-TxNet+ListMLE [14]; Constraint-Graph [25]. The last model is a competitive model this year.

5.3 Evaluation Metric

We use Kendall Tau (τ) and Perfect Match Ratio (PMR) as evaluation metrics, both being commonly used in previous work [6, 9, 14, 19, 23, 25].

Kendall Tau (τ): It computes the percentage of the sentence pair that is predicted in the wrong relative order [16]. $\tau = 1 - 2I/\binom{v_i}{2}$, where I is the number of sentence pairs which is inverted in the prediction and v_i is the number of sentences in the paragraph.

Perfect Match Ratio (PMR): It measures the percentage of instances where the predicted sentence order is totally equal to the real order [5]. That is $\text{PMR} = \frac{1}{N} \sum_{i=1}^N \mathbf{I}\{\hat{\mathbf{o}}_i = \mathbf{o}_i^*\}$, where N denotes the number of instances in the dataset. $\hat{\mathbf{o}}$ and \mathbf{o}_i^* mean the predicted and real order of instance i . It is the strictest metric.

Table 3. Results

		NIPS		AAN		SIND	
Model		τ	PMR	τ	PMR	τ	PMR
Traditional method	Entity Grid	0.09	—	0.10	—	—	—
	Window Network	0.59	—	0.65	—	0.28	—
Sequence generation model	CNN + PtrNet	0.6976	19.36	0.6700	28.75	0.4197	9.50
	LSTM + PtrNet	0.7373	20.95	0.7394	38.30	0.4833	12.96
	Variant-LSTM + PtrNet	0.7258	22.02	0.7521	40.67	0.4878	13.57
	ATTOrderNet	0.7466	21.22	0.7493	40.71	0.4823	12.27
	HierarchicalATTNet	0.7008	19.63	0.6956	30.29	0.4814	11.01
	SE-Graph	0.7370	24.63	0.7616	41.63	0.4804	12.58
	ATTOrderNet + TwoLoss	0.7357	23.63	0.7531	41.59	0.4952	14.09
	RankTxNet + ListMLE	0.7316	20.40	0.7579	36.89	0.5560	13.93
	Constraint-Graph	0.8029	32.84	0.8236	49.81	0.5856	19.07
Pairwise model	B-TSort	0.8100	32.59	0.8290	50.38	0.6000	20.32
Ours		0.8181	34.32	0.8330	51.03	0.6173	22.33

5.4 Results

By Table 3, we can see our model outperforms all the baselines on both τ and PMR on the three datasets. Our model outperforms the previous best method

B-TSort, which is also a pairwise model, by about 0.8% τ score and 1.7% PMR score in NIPS dataset, 0.4% τ score and 0.7% PMR score in AAN, 1.7% τ score and 2.0% PMR score in SIND. The results show that it is efficient to utilize the global information to help predict the relative order of the sentence pair. Comparing with the best-performing sequence generating approach Constraint-Graph, the prediction of current time step in our model is independent on the previous and it brings about 2% more score in all datasets and metrics. The most significant improvement is on SIND, where our method beats Constraint-graph by about 3% score in τ and PMR. We believe because all data instances in SIND have 5 sentences, and the context information is much more important in this situation compared to instances that only have 1 or 2 sentences.

Moreover, we divide the dataset into 3 parts on the numbers of sentences to prove the global information encoder catches the context information that is useful to decide the relative order between sentence pair. Because each paragraph in SIND dataset has equal number of sentences, we do this test on NIPS and AAN abstract. Table 4 shows that generally, our method gets higher results on all 3 segments. Moreover, if there are more the sentences in the sample, the difference of performance between ours and B-TSort is larger. This shows utilizing the global information can improve the prediction of relative sentence order as the information of context expands.

Table 4. Tau score in AAN and NIPS

Model	AAN				NIPS			
	NUM							
	All	1–3	4–8	9–15	All	1–3	4–6	7–20
B-TSort	0.8296	0.8908	0.8226	0.7224	0.8105	0.9298	0.8199	0.7865
Ours	0.8331	0.8937	0.8261	0.7276	0.8181	0.9298	0.8338	0.7875
Difference	+0.0035	+0.0029	+0.0035	+0.0052	+0.0076	0	+0.0139	+0.0010

5.5 Ablation Study

We propose 2 new models and compare them with our approach C-TSort and 2 competitive baselines to explore the impact of different modules of our model. The ablation results are illustrated by Table 5.

Ablation Model 1. To test the effect of information sharing among sentences, we propose this ablation model. In global information encoder, we remove the Transformer Network and pass the sentence embeddings to average pooling layer straightly. By Table 5, we can see C-TSort outperforms the ablation model 1 by about 1% score in all datasets and metrics. It proves information sharing in the generation of the context is essential to predict the relative order of the sentence pair.

Table 5. Ablation results

Model	NIPS		AAN	
	τ	PMR	τ	PMR
Baseline				
B-TSort	0.8100	32.59	0.8290	50.38
Constraint-Graph	0.8029	32.84	0.8236	49.81
Ours				
C-TSort	0.8181	34.32	0.8330	51.03
Ablation model 1	0.8090	33.08	0.8279	50.04
Ablation model 2	0.8054	33.33	0.8303	51.45
Pre-fusion model	0.7923	28.86	0.8127	47.52

Ablation Model 2. To compare the efficiency of different manners of information sharing among sentences through post-fusion to help predict the relative order of sentence pair. We revise the approach to generating the global encoding and propose another post-fusion based model. In the global encoder, the input of BERT is a “[CLS]” token, followed by a sequence of tokens of unordered context. We apply the first embedding corresponding to the “[CLS]” token as the paragraph embedding. Then we concatenate it with the sentence pair embedding to predict the relative order by a classifier and arrange with topological sort. By Table 5, we can see our model C-TSort outperforms ablation model 2 in most of the datasets and metrics. It proves the Transformer Network of C-TSort is more efficient in information sharing among sentences to order the sentence pair. We believe that using the whole paragraph as the input of BERT introduces noise in token level, which affects the performance on sentence pair ordering. Moreover, ablation model 2 also beats baselines in both datasets and metrics, which proves post-fusing context information helps predict the relative pairwise order.

5.6 Comparing with Pre-fusion

Pre-fusion of global and local information can connect the information directly in a token-level and is also an approach that deserves consideration. To compare the pre-fusion and post-fusion approach on incorporating global and local information to predict the relative order of sentence pair, we propose a new model based on pre-fusion of global and local information. We utilize BERT to fuse the global and local information. For each sentence pair s_1 and s_2 , the input of BERT is “[CLS] s_1 [SEP] s_2 [SEP] context”, where context is the given unordered text. Then we use the first hidden state of BERT corresponding to the “[CLS]” token to predict the relative sentence order with a MLP classifier and organize the text by topological sort. By Table 5, it shows out model C-TSort beats this pre-fusion model by over 2% score in all datasets and metrics. It proves post-fusion is more efficient in including global and local information to predict the relative order of sentence pair. We believe that as the whole paragraph usually contains

a huge amount of information, which far beyond what actually needed, the noise is inevitably introduced through intensive information sharing inside BERT and undermines the performance on the prediction (Table 6).

6 Case Study

We take several examples in SIND dataset where B-TSort is not able to predict correctly without global context while our model gets the perfect match.

Table 6. Several cases

Right order	Sentences	Ours	B-TSort
(1)	We set up our chair for the softball game	(1)	(1)
(2)	They were getting ready to play. The ball was thrown	(2)	(2)
(3)	She ran to catch the ball	(3)	(3)
(4)	Our team scored a point	(4)	(5)
(5)	We enjoyed relaxing and watching the game	(5)	(4)
(1)	Becca and Bob posed for a picture before their hike	(1)	(1)
(2)	We hiked over this bridge. It felt a little unstable	(2)	(2)
(3)	The water was really roaring fast. The water was really cold	(3)	(4)
(4)	The rocks were neat to look at. We wanted to see it all	(4)	(3)
(5)	At the end of the day we were at the top. The view was just beautiful	(5)	(5)
(1)	The Karate Championship was a big hit!	(1)	(1)
(2)	All the best fighters showed their kicks	(2)	(2)
(3)	Legs and fists were flying around	(3)	(3)
(4)	It was very fun to see all this action	(4)	(5)
(5)	Our family came away with four trophies, we were really happy	(5)	(4)

In example 1, we can not decide the relative order between s_4 and s_5 without the context information, so B-TSort does the wrong prediction on this pair. By s_1 , s_2 and s_3 we know the action “she caught the ball” leads to scoring a point, so s_4 should follow s_3 and s_5 should be a conclusion corresponding to the first sentence s_1 . The relative order between s_4 and s_5 can be determined by the context.

In example 2, the relative order of s_3 and s_4 is hard to decide without the context because both the water and rocks can appear first in the description. However, with s_2 , we know the description of river should follow the bridge and the water is cold and fast in s_3 corresponds to the feeling of unstable in s_2 . Therefore, s_3 should follow s_2 and the relative order of s_3 and s_4 is easy to determine.

In example 3, we can not know what the “action” refers to in s_4 and the connection of s_4 and s_5 without the context. B-TSort predicts the wrong relative order of s_4 and s_5 . However, by s_2 and s_3 , we can know the “action” refers to the competition of the fighters, so s_4 should follow the s_2 and s_3 . The “trophies” in s_5 corresponds to the “Karate Championship” in s_1 . s_5 gives the result of the competition and it should be the conclusion of the text. The relative order of s_4 and s_5 can be determined with the context information and our model predicts it right.

7 Conclusion

In this paper, we propose a new and simple model CPCN for sentence ordering where we combine the global and local information through post-fusion to predict the relative order of the sentence pair. This is based on the observation that the context information is essential to predict the relative order of the sentence pair. We utilize BERT to encode each sentence and pass the sentence embeddings to a Transformer Network and an average pooling layer to get the context embedding. Then we concatenate the context embedding and sentence pair embedding as the input of a MLP classifier. At last we organize into the text with topological sort. Our model achieves the state of the art on three benchmarks in the field of sentence ordering.

References

1. Barzilay, R., Elhadad, N.: Inferring strategies for sentence ordering in multidocument news summarization. *J. Artif. Intell. Res.* **17**, 35–55 (2002)
2. Barzilay, R., Lapata, M.: Modeling local coherence: an entity-based approach. *Comput. Linguistics* **34**, 1–34 (2008)
3. Barzilay, R., Lee, L.: Catching the drift: probabilistic content models, with applications to generation and summarization. arXiv preprint cs/0405039 (2004)
4. Chandu, K., Nyberg, E., Black, A.W.: Storyboarding of recipes: grounded contextual generation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 6040–6046 (2019)
5. Chen, X., Qiu, X., Huang, X.: Neural sentence ordering. arXiv preprint [arXiv:1607.06952](https://arxiv.org/abs/1607.06952) (2016)
6. Cui, B., Li, Y., Chen, M., et al.: Deep attentive sentence ordering network. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (2018)
7. Devlin, J., Chang, M.W., Lee, K., et al.: 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, vol. 1, pp. 4171–4186 (2019)
8. Fan, A., Lewis, M., Dauphin, Y.: Strategies for structuring story generation. arXiv preprint [arXiv:1902.01109](https://arxiv.org/abs/1902.01109) (2019)
9. Gong, J., Chen, X., Qiu, X., et al.: End-to-end neural sentence ordering using pointer network. arXiv preprint [arXiv:1611.04953](https://arxiv.org/abs/1611.04953) (2016)
10. Huang, T.H., Ferraro, F., Mostafazadeh, N., et al.: Visual storytelling. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1233–1239 (2016)
11. Hu, J., Cheng, Y., Gan, Z., et al.: What makes a good story? Designing composite rewards for visual storytelling. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 7969–7976 (2020)
12. Konstas, I., Lapata, M.: Concept-to-text generation via discriminative reranking. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 369–378 (2012)
13. Konstas, I., Lapata, M.: Inducing document plans for concept-to-text generation. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1503–1514 (2013)

14. Kumar, P., Brahma, D., Karnick, H., et al.: Deep attentive ranking networks for learning to order sentences. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 8115–8122 (2020)
15. Lapata, M.: Probabilistic text structuring: experiments with sentence ordering. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, pp. 545–552 (2003)
16. Lapata, M.: Automatic evaluation of information ordering: Kendall’s tau. *Comput. Linguistics* **32**, 471–484 (2006)
17. Li, J., Hovy, E.: A model of coherence based on distributed sentence representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2039–2048 (2014)
18. Li, J., Jurafsky, D.: Neural net models of open-domain discourse coherence. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 198–209 (2017)
19. Logeswaran, L., Lee, H., Radev, D.: Sentence ordering and coherence modeling using recurrent neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
20. Nallapati, R., Zhai, F., Zhou, B.: SummaRuNNer: a recurrent neural network based sequence model for extractive summarization of documents. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017)
21. Prabhumoye, S., Salakhutdinov, R., Black, A.W.: Topological sort for sentence ordering. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 2783–2792 (2020)
22. Wang, T., Wan, X.: Hierarchical attention networks for sentence ordering. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 7184–7191 (2019)
23. Yin, Y., Meng, F., Su, J., et al.: Enhancing pointer network for sentence ordering with pairwise ordering predictions. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 9482–9489 (2020)
24. Yin, Y., Song, L., Su, J., et al.: Graph-based neural sentence ordering. arXiv preprint [arXiv:1912.07225](https://arxiv.org/abs/1912.07225) (2019)
25. Zhu, Y., Zhou, K., Nie, J.Y., et al.: Neural sentence ordering based on constraint graphs. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 14656–14664 (2021)



A Dual-Attention Neural Network for Pun Location and Using Pun-Gloss Pairs for Interpretation

Shen Liu^{1(✉)}, Meirong Ma², Hao Yuan², Jianchao Zhu², Yuanbin Wu¹,
and Man Lan¹

¹ School of Computer Science and Technology, East China Normal University,
Shanghai, China

shenliu@stu.ecnu.edu.cn, {ybwu,mlan}@cs.ecnu.edu.cn

² Shanghai Transsion Co., Ltd, Shanghai, China
{meirong.ma,hao.yuan,jianchao.zhu}@transsion.com

Abstract. Pun location is to identify the punning word (usually a word or a phrase that makes the text ambiguous) in a given short text, and pun interpretation is to find out two different meanings of the punning word. Most previous studies adopt limited word senses obtained by WSD (Word Sense Disambiguation) technique or pronunciation information in isolation to address pun location. For the task of pun interpretation, related work pays attention to various WSD algorithms. In this paper, a model called DANN (**D**ual-**A**ttentive **N**eural **N**etwork) is proposed for pun location, effectively integrates word senses and pronunciation with context information to address two kinds of pun at the same time. Furthermore, we treat pun interpretation as a classification task and construct pun-gloss pairs as processing data to solve this task. Experiments on the two benchmark datasets show that our proposed methods achieve new state-of-the-art results. Our source code is available in the public code repository (<https://github.com/LawsonAbs/pun>).

Keywords: Pun location · Pun interpretation · Pronunciation ·
Pun-gloss pairs · Word Sense Disambiguation

1 Introduction

Puns where the two meanings share the same pronunciation are known as homophonic (i.e., homographic puns), while those relying on similar but not identical-sounding signs are known as heterophonic (i.e., heterographic puns). Figure 1 shows two examples. Pun location aims to find the word appearing in the text that implies more than one meaning and pun interpretation is an attempt to give the two word senses of the punning word.

Pun location and interpretation have a wide range of applications [11, 12]. Sequence labeling is a general framework to solve pun location [2, 20, 21]. Cai *et al.* [2] proposed Sense-Aware Neural Model (SAM) which is built on the WSD

- Ex1: (homophonic) I used to be a banker but I lose **interest**.
- interest%1:21:00::* a fixed charge for borrowing money; usually a percentage of the amount borrowed
- interest%1:09:00::* a sense of concern with and curiosity about someone or something
- Ex2: (heterophonic) The boating store had its best **sail** ever.
- sail%1:06:00::* a large piece of fabric (usually canvas fabric) by means of which wind is used to propel a sailing vessel
- sale%1:04:00::* a particular instance of selling

Fig. 1. Two samples drawn from two different types puns and their corresponding punning words with the glosses (the definition of word senses) from WordNet (<https://wordnet.princeton.edu/>).

(Word Sense Ambiguation) algorithms. It suffers from the bias because of the following reasons: (1) It is inadequate to identify the punning word by using two distinct word senses; (2) The results produced by the WSD algorithms are not always correct, so the error propagation can not be ignored. Moreover, they fail to address the heterographic puns task. Therefore, Zou *et al.* [20] add a pronunciation module to the model which is named PCPR(**P**ronunciation-attentive **C**ontextualized **P**un **R**ecognition) to solve the heterographic puns. However, only utilizing the contextual and pronunciation information, PCPR omits the word senses which are the most important elements in natural language. According to the categories of puns, it is intuitive to assume that both word senses and pronunciation are the key points in pun location. So to resolve this problem, we propose a model called DANN(**D**ual-**A**ttentive **N**eural **N**etwork) to capture the rich semantic and pronunciation information simultaneously. In DANN, the sense-aware and pronunciation-aware modules employ the word meanings and phoneme information respectively. Firstly, unlike SAM, we capture semantic information by paying attention to all meanings of the word automatically rather than selecting several word senses by WSD algorithms in advance. Secondly, we consolidate word senses, context, and pronunciation information to deal with all kinds of puns.

For pun interpretation, Duluth [14] and BuzzSaw [13] both use the WSD algorithm to choose the most probable meaning for the punning word. Specifically, Duluth uses 19 different configurations to create a set of candidate target senses and choose the two most frequent senses from them as the final predicted value. However, one limitation of this approach is the uncertain level of accuracy of the WSD algorithms, which vary from word to word and domain to domain [14]. Different from Duluth and BuzzSaw, we treat pun interpretation as a sentence pair matching task, that is, we use a pre-training model (e.g., BERT) to select the best matching pun and paraphrase pairs. Concatenating the pun and the gloss of the punning word to one whole sentence, we classify it as yes or no to identify the word sense is correct or not.

In summary, our contributions are as follows: (1) We take full advantage of semantic and phonetic features to conduct the pun location. By the dual-attentive module, both of them can be taken into account.

(2) We further explore which meanings of words can lead to rhetorical effects, which is essential for understanding puns. Compared with the simple WSD algorithms, an innovative method through pun-gloss pairs to solve the pun interpretation greatly improves the experimental result.

(3) Both models achieve state-of-the-art performance in the benchmark dataset.

2 Related Work

2.1 Pun Location

Fixed patterns or characteristics are proposed to solve pun location [8, 12, 14]. Yang *et al.* [19] creatively designed a set of features from four aspects as follows: (a) Incongruity; (b) Ambiguity; (c) Interpersonal Effect; (d) Phonetic Style. Based on the characteristics of manual design, Duluth [14] proposed approaches that relied on WSD and measures of semantic correlation. Using some feature components, Vechtomova *et al.* [17] ranked words in the pun by a score calculated as the sum of values of eleven features. Indurthi *et al.* [8] select the latter word as a punning word from the maximum similarity word pair. A computational model of humor in puns based on entropy was proposed in [9]. PMI (Pointwise Mutual Information) [3] to measure the association between words is used in [15]. Doogan *et al.* [5] proposed a probabilistic model to produce candidate words. Feng *et al.* [6] first collect 10 kinds of features for this task, then they use logistic regression to find out which word is punning and use the weight of different features to explain why a punning word is detected.

Based on neural network, some methods are proposed to solve pun location [2, 10, 20]. Mao *et al.* [10] proposed CSN-ML (Compositional Semantics Network with Task Learning) to capture the rich semantic information of punning words in a sentence. Cai *et al.* [2] proposed SAM (Sense-Aware Neural Model) which is built on limited WSD results. Their main idea is modeling multiple sequences of word senses corresponding to different WSD results, which were obtained by various WSD algorithms. Zhou *et al.* [20] proposed a model named PCPR (Pronunciation-attentive Contextualized Pun Recognition) with current best effectiveness.

Different from these work, we incorporate both semantic and phonetic information into the model and solve pun location perfectly.

2.2 Pun Interpretation

Duluth [14] use a WSD algorithm on different configurations and then take the MFS(Most Frequent Senses) strategy to predict the appropriate meaning for punning word and get the current best performance. However, the MFS strategy is too fixed to address the problem of selecting word senses. Instead of using the

WSD algorithm directly, we get the meanings from top-2 pun-gloss pairs with the highest probability as the final results for each target word.

BuzzSaw [13] hypothesize that a pun can be divided into two parts, each containing information about the two distinct senses of the pun, can be exploited for pun interpretation, then they use the method that loosely based on the Lesk algorithm to get the meaning for each polysemous word. Due to error propagation, the pipelined way do not get the best performance in this problem. Therefore, we use pun-gloss pairs to fuse the pun and the gloss of the punning word to one sentence and reduce the process directly. The corresponding experiment shows that our model outperforms all other models.

3 Methodology

Figure 2 shows our model architecture for pun location. Our model is a sequence labeling system, which is based on the adaptation of the BIO annotation ['O', 'P'], where P stands for the punning word tokens and O stands for other tokens. With this tagging scheme, each word in a sentence will be assigned a label.

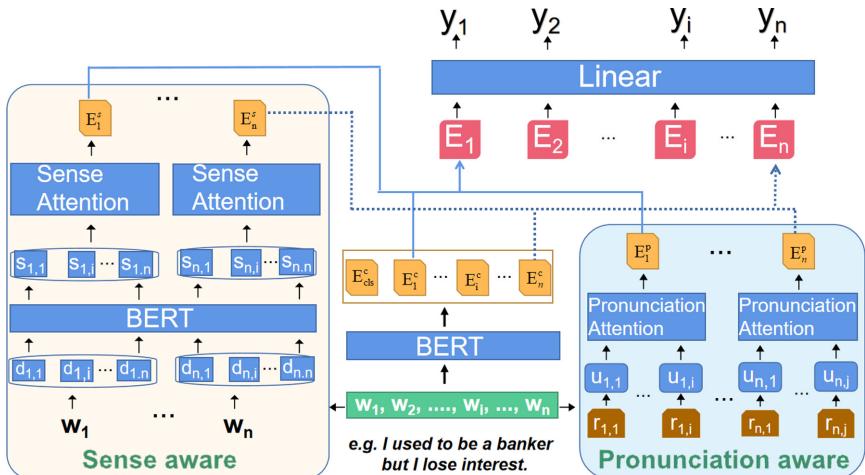


Fig. 2. The model architecture of Dual-Attentive Neural Network for Pun Location. We use a dual-attentive module to focus on crucial word senses and pronunciation.

Table 1 shows the main construction of training data to solve pun interpretation. Inspired by the GlossBERT [7], we use the pun-gloss pairs to capture the correlation between the pun and the gloss of target word. Conventional WSD methods usually return the sense with the highest score. Similarly, we can choose the best and second-best word meanings according to the maximum and sub-maximum probability values returned in the classification process.

Table 1. The sample was taken from SemEval-2017 task 7 dataset to explain the construction methods that concatenating the pun and the gloss. The ellipsis “...” indicates the remainder of the sentence.

Homographic pun:			
I used to be a banker but i lose <u>interest</u>			
Pun-Gloss Pairs of the punning word		Label	Sense Key
[CLS] I used to be a ...[SEP]	a sense of concern ... [SEP]	Yes	interest%1:09:00::
[CLS] I used to be a ...[SEP]	a reason for wanting ... [SEP]	No	interest%1:07:01::
[CLS] I used to be a ...[SEP]	excite the curiosity of ... [SEP]	No	interest%2:37:00::
[CLS] I used to be a ...[SEP]	a fixed charge for ...[SEP]	Yes	interest%1:21:00::

3.1 Pun Location

Sense-Aware Module. The highlight of our model is using the sense-attention module to focus on pertinent word senses automatically.

As shown in the lower left corner of the Fig. 2. Firstly, we get all definitions of the word senses from WordNet for each content word in a pun and denote them as $\{d_{1,1}, \dots, d_{1,i}, \dots, d_{n,n}\}$. Secondly, we use BERT [4] to process each definition and use its [CLS] token embedding as the representation and denoted them as $\{s_{1,1}, \dots, s_{1,i}, \dots, s_{1,n}\}$. For each word sense embedding $s_{i,j}$ of the word w_i , we project $s_{i,j}$ to a trainable vector $s'_{i,j}$ to represent its meaning properties. Based on the word sense embeddings, we apply the attention mechanism [16] to simultaneously identify important meanings and derive the composite word sense embedding E_i^S .

Specifically, the embedding of word senses are transformed by a fully-connected hidden layer (i.e., $F_S(\cdot)$), and then multiplying by the query vector (i.e., q) to measure the importance scores $\alpha_{i,j}$ of word sense embeddings as follows:

$$v_{i,j} = F_S(s_{i,j}) \quad (1)$$

$$\alpha_{i,j} = \frac{v_{i,j} \cdot q}{\sum_k v_{i,k} \cdot q} \quad (2)$$

Finally, the synthetical sense embedding E_i^S can be generated by the weighted combination of various embeddings as follows:

$$E_i^S = \sum_j \alpha_{i,j} \cdot s'_{i,j} \quad (3)$$

We select context-sensitive paraphrases to help determine whether a word is the target word through using the attention mechanism. Nevertheless, not all words in the input sentence W_1, W_1, \dots, W_n have the same number of meanings, so this is a hyperparameter, which will be described in Sect. 4.2. After that, a synthetic representation vector (i.e., E_i^S) of the various meanings of each word will be got.

Pronunciation-Aware Module. It is well-known that pronunciation plays an essential role in the language. Inspired by the PCPR, we also introduce a pronunciation-aware module into DANN to solve the heterographic puns. By projecting pronunciation to the embedding space, words that sound alike are nearby to each other [1]. Each word is divided into phonemes (i.e., $\{r_{1,1}, \dots, r_{1,i}, \dots, r_{n,j}\}$) which represent the characteristics in pronunciation. Each phoneme is projected to a phoneme embedding space (i.e., $\{u_{1,1}, \dots, u_{1,i}, \dots, u_{n,j}\}$). Pronunciation vector (i.e., E_i^P) can be obtained with the attention mechanism. Through the pronunciation component, we can join words with the same sound together.

Implementation Details. In our work, we use BERT to get all word embeddings for the whole input sentence. So we can get E^c , E^s , E^p to present contextual embedding, word sense, and pronunciation embedding of the word respectively. Then our model concatenates these embeddings and converts them to a project layer, we can get every word's predicted value y_i .

Specifically, first, the BERT model processes the input then gets every word's contextual embedding, we denote them as E^c . Second, we use every word's pronunciation embedding, and after the attention process, we get embedding E^p to denote the important pronunciation. Third, word sense embedding serves as the input of the sense-attention module to get the compounded representation of the word, we denote it as embedding E^s . Last, all embedding parts are concatenated to get the final expression (i.e., E_i) of i -th word.

$$E_i = E_i^s \oplus E_i^s \oplus E_i^p$$

E_i will be transferred to a project layer to determine whether the i -th word is a punning word.

3.2 Pun Interpretation

Framework Overview. BERT uses a “next sentence prediction” task to train text-pair representations, so it can explicitly model the relationship of a pair of texts, which has shown to be beneficial to many pair-wise natural language understanding tasks [4]. To fully leverage gloss information, we construct pun-gloss pairs over puns and all possible senses of the punning word in WordNet, thus treating the WSD task as a sentence-pair classification problem.

Table 1 shows the main construction process of training examples. The sentence containing the punning word is denoted as a *pun* sentence. For punning words, we extract glosses of all senses from WordNet. An example in homographic pun gives a detailed introduction of the construction method (See Table 1). *Interest* is a punning word. [SEP] mark is added to the pun-gloss pairs to separate pun from paraphrasing. Each target word has a set of pun-gloss pair training instances with label $\in \{yes, no\}$.

The pun-gloss pairs will serve as inputs to the BERT, and the output of the model is *yes* or *no*. The “*yes*” represents the gloss following the pun is the sense definition of the punning word, the “*no*” stands for the contrary meaning.

For clarity and convenience, we use the sense key from WordNet to stand for concrete definition.

Implementation Details. We use BERT as our pre-training approach. In training, we get the whole sentence and use BERT to get the [CLS] token embedding, then a linear layer is used to obtain the classification results. Cross-entropy loss is used when adjusting model weights during training. When testing, we output the sense key of the punning word with the two maximum probabilities for each pun-gloss pair.

4 Experiment Settings

4.1 Dataset and Evaluation Metrics

We evaluate our models on the SemEval-2017 shared task 7 dataset¹. Homographic puns and heterographic puns have 1607 and 1271 samples respectively. Due to the limited data and keep the equity of evaluation, we perform ten cross-validation as the same as PCPR and SAM, then use the average of the evaluation result as the final score. Meanwhile, we use the same metrics with them.

4.2 Baselines

Pun Location. We compare our model with the following baselines. (1) Olga [17]. (2) Idiom Savant [5]. (3) Fermi [8]. (4) ECNU [18]. (5) BERT [4]. (6) LRegression [6]. (7) SAM [2]. (8) JDL [21]. (9) PCPR [20]. We directly quote the experimental results of these baselines except BERT.

Pun Interpretation. The top-3 competition models in SemEval-2017 task-7 would be used as the baselines.

4.3 Hyperparameters

Different words have different numbers of meanings, so the number of word senses that should be obtained in the model is a hyperparameter which is denoted as d_s . In our work, we use 50 different meanings of a word, and if the word does not have 50 meanings, then it will be initialized to zero embeddings.

5 Experimental Results and Analysis

5.1 Pun Location

Table 2 shows the specific experimental results. Compared to PCPR, DANN achieves the highest performance with 1.5% and 0.6% improvements of $F1$ for

¹ <https://alt.qcri.org/semeval2017/task7/>.

Table 2. Results of DANN and strong baselines on Semeval-2017 task 7 for pun location. * means that the experiments are reproduced in our work.

System	Homographic			Hetergraphic		
	P	R	F1	P	R	F1
Olga	0.652	0.652	0.652	0.797	0.795	0.796
Idiom Savant	0.663	0.663	0.663	0.684	0.684	0.684
Fermi	0.521	0.521	0.521	–	–	–
ECNU	0.337	0.337	0.337	0.568	0.568	0.568
BERT*	0.884	0.870	0.877	0.924	0.925	0.924
LRegression	0.762	0.762	0.762	0.849	0.849	0.849
SAM	0.815	0.747	0.780	–	–	–
JDL	0.835	0.771	0.802	0.814	0.775	0.794
PCPR	0.904	0.875	0.889	0.942	0.904	0.922
DANN	0.895	0.914	0.904	0.918	0.939	0.928

the homographic and heterographic datasets respectively. By applying the sense-attention module, we pick out the most valuable meanings to conduct the detecting punning word task. Our model outperforms all baseline models, which indicates that the sense-aware module plays a crucial role, especially in homographic puns.

5.2 Pun Interpretation

Table 3 shows that our model achieves the highest performance with 9.16% improvements of $F1$ against the best among the baselines (i.e. Duluth) for the homographic puns. We posit the reason is that our model makes a good connection between the pun and the gloss of the punning word. So it is possible to see if a relevant definition matches the pun.

Table 3. Results of our model and baselines on Semeval-2017 task 7 for pun interpretation.

System	Homographic		
	P	R	F1
Duluth	0.144	0.168	0.155
BuzzSaw	0.152	0.156	0.154
Ours	0.247	0.247	0.247

Figure 3 shows two examples of the explanation in homographic puns. In the first example, all displaying senses are nouns, *relief%1:12:00::* and

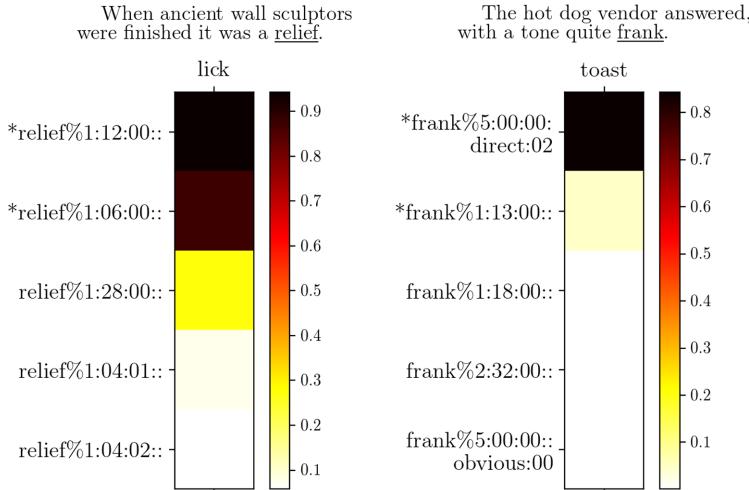


Fig. 3. The sense key with top-5 probability for each target word in two samples. The punning words are underlined, and an asterisk indicates the meaning of the word that causes a pun.

relief%1:06:00:: have a higher score because they are closely related to the context. In the second example, although *frank%5:00:00:direct:02* (adjective) and *frank%1:13:00::* (noun) have different parts of speech, they could also get relatively higher attention scores in this process. We assume that the possible reasons are as follows: (1) It is easy to find out the primary meaning of *frank*, so the probability of *frank%5:00:00:direct:02* is the greatest. (2) The synonyms of *frank%1:13:00::* include *hot_dog%1:13:01::*. The gloss (i.e., a smooth-textured sausage of minced beef or pork usually smoked; often served on a bread roll) of *frank%1:13:00::* have a correlation with *hot dog*, so it has the second highest probability score.

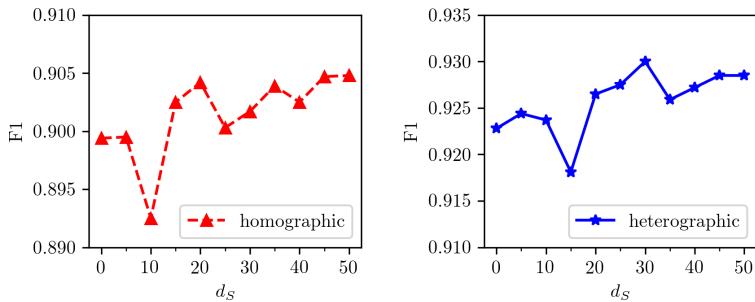
5.3 Analysis

Case Study. Table 4 shows the experimental results on several cases between PCPR and DANN. It is obvious to find a significant difference in homographic puns. The valid reason is that the rich semantic information is captured by DANN but forgotten by PCPR. In the first case, *patent* is predicted by the former but *lies* by the latter. We can infer that only considering pronunciation will introduce bias to the model, but the DANN could correct this bias caused by insufficient information through introducing word senses. Except for words with more meanings like *get*, our model got the correct answer on almost every sample. Because these words have so many meanings, it is not a simple matter to find out exactly one definition of them.

Table 4. The cases of homographic puns (shown in bold) identified by PCPR and DANN models.

Sentence	PCPR	DANN
He stole an invention and then told patent lies	lies	patent
A thief who stole a calendar got twelve months	-	months
Finding area is an integral part of calculus	calculus	integral

Effect of Number of Word Senses. Figure 4 shows the diverse results of the model with a different number of meanings. There is no doubt that the more word senses you use, the higher the $F1$ score you will get. Meanwhile, to keep fair comparison, the hyperparameters we use are exactly the same as in the PCPR, such as phoneme embedding size d_p and attention size d_A .

**Fig. 4.** Performance over different word sense number in homographic and heterographic puns.

6 Conclusions and Future Work

In this paper, we propose a novel SOTA model named DANN, which leverages word senses and pronunciation to solve pun location. Empirically, it outperforms previous methods that rely heavily on handcrafted features or another single characteristic. Moreover, we formulate pun interpretation as a classification task and construct pun-gloss pairs to solve it. The experiments show that this method achieves the new best performance with nearly 9.2% improvement in homographic puns. In the future, we plan to focus on exploring more effective ways to pun interpretation. Furthermore, due to the rich emotional information in puns, we want to incorporate it into sentiment analysis and text generation to make the machine look smarter.

Acknowledgements. We thank the anonymous reviewers for their thoughtful comments. This work has been supported by Shanghai Transsion Co., Ltd.

References

1. Bengio, S., Heigold, G.: Word embeddings for speech recognition (2014)
2. Cai, Y., Li, Y., Wan, X.: Sense-aware neural models for pun location in texts. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia (Volume 2: Short Papers), pp. 546–551. Association for Computational Linguistics, July 2018. <https://doi.org/10.18653/v1/P18-2087> <https://aclanthology.org/P18-2087>
3. Church, K., Hanks, P.: Word association norms, mutual information, and lexicography. *Comput. Linguist.* **16**(1), 22–29 (1990)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) (2018)
5. Doogan, S., Ghosh, A., Chen, H., Veale, T.: Idiom savant at SemEval-2017 task 7: detection and interpretation of English puns. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 103–108 (2017)
6. Feng, J., Sevgili, Ö., Remus, S., Ruppert, E., Biemann, C.: Supervised pun detection and location with feature engineering and logistic regression. In: SwissText/KONVENTS (2020)
7. Huang, L., Sun, C., Qiu, X., Huang, X.: Glossbert: bert for word sense disambiguation with gloss knowledge. arXiv preprint [arXiv:1908.07245](https://arxiv.org/abs/1908.07245) (2019)
8. Indurthi, V., Oota, S.R.: Fermi at semeval-2017 task 7: detection and interpretation of homographic puns in English language. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 457–460 (2017)
9. Kao, J.T., Levy, R., Goodman, N.D.: The funny thing about incongruity: a computational model of humor in puns. In: Proceedings of the Annual Meeting of the Cognitive Science Society, vol. 35 (2013)
10. Mao, J., Wang, R., Huang, X., Chen, Z.: Compositional semantics network with multi-task learning for pun location. *IEEE Access* **8**, 44976–44982 (2020)
11. Miller, T., Gurevych, I.: Automatic disambiguation of English puns. 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 (Volume 1: Long Papers), pp. 719–729. Association for Computational Linguistics (2015)
12. Miller, T., Hempelmann, C., Gurevych, I.: SemEval-2017 task 7: Detection and interpretation of English puns. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Vancouver, Canada, pp. 58–68. Association for Computational Linguistics, August 2017. <https://doi.org/10.18653/v1/S17-2005> <https://aclanthology.org/S17-2005>
13. Oele, D., Evang, K.: Buzzsaw at SemEval-2017 task 7: global vs. local context for interpreting and locating homographic English puns with sense embeddings. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 444–448 (2017)
14. Pedersen, T.: Duluth at SemEval-2017 task 7: puns upon a midnight dreary, lexical semantics for the weak and weary. arXiv preprint [arXiv:1704.08388](https://arxiv.org/abs/1704.08388) (2017)
15. Sevgili, Ö., Ghotbi, N., Tekir, S.: N-hance at SemEval-2017 task 7: a computational approach using word association for puns. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 436–439 (2017)
16. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)

17. Vechtomova, O.: Uwaterloo at SemEval-2017 task 7: locating the pun using syntactic characteristics and corpus-based metrics. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 421–425 (2017)
18. Xiu, Y., Lan, M., Wu, Y.: ECNU at SemEval-2017 task 7: using supervised and unsupervised methods to detect and locate English puns. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 453–456 (2017)
19. Yang, D., Lavie, A., Dyer, C., Hovy, E.: Humor recognition and humor anchor extraction. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 2367–2376 (2015)
20. Zhou, Y., Jiang, J.Y., Zhao, J., Chang, K.W., Wang, W.: “The boating store had its best sail ever”: pronunciation-attentive contextualized pun recognition. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 813–822. Association for Computational Linguistics, Online, July 2020. <https://doi.org/10.18653/v1/2020.acl-main.75>. <https://aclanthology.org/2020.acl-main.75>
21. Zou, Y., Lu, W.: Joint detection and location of English puns (2019)



A Simple Baseline for Cross-Domain Few-Shot Text Classification

Chen Zhang and Dawei Song^(✉)

Beijing Institute of Technology, Beijing, China
{czhang,dwsong}@bit.edu.cn

Abstract. Few-shot text classification has been largely explored due to its remarkable few-shot generalization ability to in-domain novel classes. Yet, the generalization ability of existing models to cross-domain novel classes has seldom be studied. To fill the gap, we investigate a new task, called cross-domain few-shot text classification (XFew) and present a simple baseline that witnesses an appealing cross-domain generalization capability while retains a nice in-domain generalization capability. Experiments are conducted on two datasets under both in-domain and cross-domain settings. The results show that current few-shot text classification models lack a mechanism to account for potential domain shift in the XFEW task. In contrast, our proposed simple baseline achieves surprisingly superior results in comparison with other models in cross-domain scenarios, confirming the need of further research in the XFEW task and providing insights for possible directions. (The code and datasets are available at <https://github.com/GeneZC/XFew>).

Keywords: Cross-domain setting · Few-shot learning · Text classification

1 Introduction

Few-shot text classification aims at learning a text classifier for low-resource classes (a.k.a. novel classes, with only few labeled examples in each class) with the aid of high-resource classes (a.k.a. base classes, with abundant labeled examples in each class). In light of advancements in few-shot learning, the approaches to few-shot text classification can be categorized into three streams: metric-based [2, 9, 17–20], optimization-based [7], and model-based [15] ones. Despite of their differences in formulation, these methods are all used to transfer the ability of learning within few shots on labeled data, from base classes to novel classes. This formulation is also referred to as learning-to-learn or meta-learning. It is noteworthy that transductive transfer learning and inductive transfer learning are not up to the task’s concern, as the former requires large amounts of unlabeled data and cannot handle disparate label set, while the latter suffers from insufficient labels.

Whereas having achieved compelling performance, existing few-shot text classification models are fundamentally limited by the implicit demand of few-shot learning for a similar data distribution between the novel classes and base classes. Thus, the models facing a gap between base classes and novel classes can produce sub-optimal results in two aspects. First, task discrepancy between base classes and novel classes can be harmful for models that are employed in real-word applications. For instance, if a model is trained in a few-shot fashion on base classes which are leveraged for news topic categorization, then the model cannot be directly adapted to sentiment classification appropriately. Second, there may exist domain shift between base classes and novel classes. An illustrative example could be that, base classes are to classify the intents of utterances related to banking while novel classes are aimed to discriminate the intents of medical questions. While the task discrepancy is to some extent addressed by [20], in which diverse metrics are proposed and properly combined to boost the performance of metric learning, the issue of domain shift lying in few-shot text classification is still challenging yet to be tackled.

To this end, we investigate a new task, namely cross-domain few-shot text classification (henceforth XFEW). To our best knowledge, our work is the first investigation on the cross-domain few-shot learning task in the area of text classification. To facilitate the understanding of XFEW, we present a simple baseline for XFEW besides examining the cross-domain generalization ability of prior models. The baseline is inspired by [5], where a model is pre-trained on all base classes in a supervised manner and fine-tuned on novel classes, with only the classification layer² being tunable. Further, based on recent observations in few-shot learning that the pre-trained model without the learned classifier, but alternatively with an instantly induced classifier, is strong enough for various few-shot image classification benchmarks [6, 10], we replace the fine-tuning stage with an induction stage in our baseline. Therefore, after pre-training, we estimate labels of unlabeled examples by instead applying the induced classifier.

A set of in-domain and cross-domain experiments are conducted with two datasets, consisting of queries from home domain and banking domain. The differences between in-domain and cross-domain results reveal that a range of state-of-the-art few-shot text classification approaches fall short in cross-domain generalization, indicating that XFEW is a challenging task that should be further studied. Surprisingly, our simple baseline achieves superior results on cross-domain settings while preserves competitive results on in-domain settings.

2 Related Work

A variety of methods have been developed for few-shot text classification, which can be folded into two main groups according to the core ideas of these algorithms, i.e., metric-based and optimization-based ones.

Metric-based methods basically learn metrics to measure the proximity between samples in query set and those in support set, therefore determining the

² The layer is a fully connected layer armed with softmax.

labels of query samples based on the labels of nearest support samples. To list a few, the prototypical network [17] takes the mean of feature vectors from support samples for each class as a class centroid, and conducts proximity measurement between query samples and class centroids based on their euclidean distance. The relation network [18] directly learns a metric by exploring the relations between each pair of individual query sample and support sample, i.e., if a concerned pair belong to the same class, then the similarity of the pair should be 1, and 0 otherwise. The induction network [9] leverages dynamic routing algorithm to replace the simple averaging strategy used in the prototypical network.

Different from metric-based approaches, optimization-based models target at finding an effective initialization so that they could adapt to any task with only few labels available. Model-agnostic Meta-Learning, in short MAML [7], is in each episode fast adapted on the support samples, then updated with second-order gradients³ back-propagated from loss computed on the query samples.

While both metric-based and optimization-based models have achieved competitive results in few-shot text classification, they may fail in the *XFew* task when the base classes and novel classes are from dissimilar domains. On the one hand, metrics may vary from one domain to another domain. On the other hand, a good initialization in a domain can be inferior for other domains.

Note that potential domain shift within few-shot learning framework has been studied in image classification scenarios [5, 10]. [5] presents an experimental setting for evaluating the cross-domain generalization ability for few-shot image classification algorithms. And [10] establishes a standard benchmark for cross-domain few-shot learning in the context of images. Likewise, [8] imposes the domain adaptation challenge for few-shot relation extraction. However, how existing systems perform on cross-domain few-shot text classification is under-explored.

3 Background

Suppose we have a set of base classes $\mathcal{C}^b = \{\mathcal{C}_i^b\}_{i=1}^{|\mathcal{C}^b|}$ with plentiful examples in each class, and a set of novel classes $\mathcal{C}^n = \{\mathcal{C}_i^n\}_{i=1}^{|\mathcal{C}^n|}$ with few examples in each class. Here, $|\cdot|$ denotes the size of a set. Either a base class or a novel class \mathcal{C} can be decomposed into examples $\{(x_i, y_i)\}_{i=1}^{|\mathcal{C}|}$, where x_i is a text and y_i is the corresponding label. The goal of few-shot text classification is to train a text classifier on base classes that can adapt to novel classes efficiently.

To achieve the goal, an episode-based training scheme is adopted in previous work. Basically, the episode-based scheme is organized in an n -way k -shot paradigm. Episodes (a.k.a. tasks) are sampled from \mathcal{C}^b for training. For each episode, there is a support set (for training in an episode) containing n classes with k examples in each class. Furthermore, additional q examples, which do not overlap with those in the support set, are sampled in each way as a query set (for testing in an episode). In doing so, models trained on training episodes

³ MAML can be simplified with first-order gradients, though.

are expected to obtain a remarkable performance when adapted to unseen novel classes since training episodes are made to mimic the low-resource situation in novel classes. The whole training procedure is given in Algorithm 1.

Algorithm 1. Episode-based training.

```

1: for each episode do
2:   Sample  $n$  classes  $\{\mathcal{C}_j\}_{j=1}^n$  from  $\mathcal{C}^b$ .  $\triangleleft$  label is re-mapped herein.
3:   for each sampled class  $\mathcal{C}_j$  do
4:     Sample  $k$  examples to form  $\mathcal{S}_j$ .
5:     Sample  $q$  examples to form  $\mathcal{Q}_j$  that are mutually exclusive with  $\mathcal{S}_j$ .
6:   end for
7:   Adapt model with support set  $\{\mathcal{S}_j\}_{j=1}^n$ , and update parameters of the model
   with query set  $\{\mathcal{Q}_j\}_{j=1}^n$ .
8: end for

```

Cross-domain few-shot text classification (XFEW) typically falls into the framework of few-shot text classification. However, the base classes and novel classes in XFEW are distinct in term of domain distributions. The current formalization posits that the data distribution of base classes and novel classes should be akin to each other. Therefore it is difficult to generalize the conventional few-shot text classification approaches to cross-domain scenarios. As a result, an investigation in XFEW is needed.

4 A Simple Baseline for XFEW

We present a simple baseline without any tricky magics. An overview of our proposed baseline is given in Fig. 1, which essentially contains two stages, that is, pre-training stage and induction stage. At pre-training stage, the baseline pre-trains the encoder and the classifier on all base classes in a supervised manner. At induction stage, the baseline adaptively induces a classifier with a few offer instances.

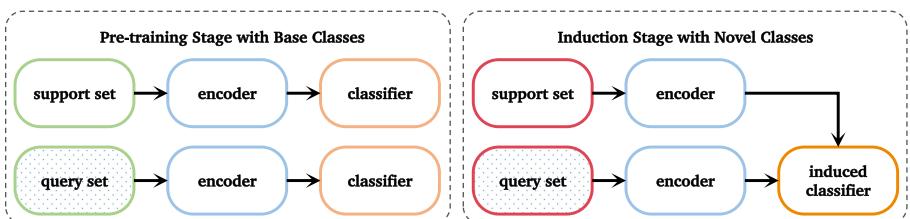


Fig. 1. An overview of our simple baseline for XFEW.

4.1 Pre-training Stage

Motivated by pre-train-then-fine-tune pipeline for few-shot learning [5], our model is pre-trained on all base classes in a supervised manner. Concretely, the model is composed of an arbitrary encoder f_θ and a classifier g_ϕ (typically a fully connected layer followed by a softmax layer). Output of the model is thereby:

$$\mathbf{y}_i = g_\phi \cdot f_\theta(x_i). \quad (1)$$

where $g_\phi \cdot f_\theta$ is a composited function.

We pre-train the model via a $|\mathcal{C}^b|$ -way cross-entropy objective, that is:

$$\mathcal{L} = - \sum_{\mathcal{C}_j \in \mathcal{C}^b} \sum_{x_i \in \mathcal{C}_j} \log(\mathbb{1}(\mathbf{y}_i, y_i)) \quad (2)$$

where $\mathbb{1}(\mathbf{y}, i)$ is a function that returns i -th element of \mathbf{y} .

4.2 Induction Stage

Recent studies [6, 10] uncover the fact that the encoder in a pre-trained model, i.e., f_θ , can be considered as a feature extractor and straightforwardly applied to find novel class centroids by averaging feature vectors belonging to the same class:

$$\mathbf{w}_j = \sum_{x_i \in \mathcal{S}_j} f_\theta(x_i)/k \quad (3)$$

where \mathbf{w}_j is the class centroid accounting for the j -th class.

These class centroids can be regarded as weights of classifier, which replaces the function of the learned classifier, thereby waiving the fine-tuning stage as required by pre-train-then-fine-tune paradigm. When additionally armed with a constant distance metric, e.g., cosine distance, the induced classifier can be used to determine labels of unlabeled data:

$$\mathbf{y}_{i,j} = \text{softmax}(\alpha \cdot \mathbf{w}_j^\top f_\theta(x_i) / \|\mathbf{w}_j\| \|f_\theta(x_i)\|), \quad x_i \in \mathcal{Q} \quad (4)$$

where $\mathbf{y}_{i,j}$ means the j -th element of \mathbf{y}_i , and $\|\cdot\|$ represents 2-norm. α here is a term to enlarge cosine values so that they can be treated differently by the softmax function.

In addition, without the fine-tuning stage, the potential over-fitting and negative transfer issues that may hinder effectiveness of pre-trained models are alleviated. Hereafter, we call our proposed model Pre-trained Network (in short PtNet).

5 Experiment

5.1 Datasets and Evaluation Metrics

In order to evaluate cross-domain generalization ability of state-of-the-art few-shot text classification systems, we introduce two data collections for XFEW.

These instances are thus from two domains. The former collection contain queries from the `Home` domain [13]⁴, while the latter one is from `Banking` domain [4]. Based on the two domains, we construct cross-domain pairs by combining above two domains. We first regard all classes in `Home` domain as the base classes (or source) and all classes in `Banking` domain as the novel classes (or target). We name the cross-domain pair as `Home2Banking`. Conversely, we can have `Banking2Home` in a similar way. Moreover, base classes are further randomly divided into classes for training and validation. Statistics of all datasets are given in Table 1.

Table 1. Statistics of all datasets.

	Home	Banking	Home2Banking	Banking2Home
# (base) classes for training	39	49	56	70
# (base) classes for validation	6	7	7	7
# (novel) classes for test	18	21	77	63

As we can observe, the numbers of base classes are generally larger than those of novel classes. We can hence generate more non-overlapped episodes for training. Meanwhile, the numbers of novel classes are not very small. This means we could produce enough testing episodes under a n -way k -shot paradigm and take the averaged results for a more robust evaluation.

We evaluate existing few-shot text classification models and our proposed baseline on these datasets. Experiments are carried out under 5-way 1-shot, 5-shot, and 10-shot settings. The numbers for training episodes, validation episodes, and testing episodes are 4000, 200, and 4000, respectively. Averaged accuracy over all testing episodes is adopted as the evaluation metric, with a 0.95 confidence level under the one-tailed hypothesis test.

5.2 Baselines

Here we list baselines for comparison as below:

- InductNet [9] uses dynamic routing to induce prototypes for classification.
- RelationNet [18] alternatively learns a metric for classification with instance-wise measurement.
- MAML [7] proposes an optimization-based algorithm to achieve meta-learning with high-order gradient descent.
- ProtoNet [17] employs averaging representations as prototypes and carry out measurement with a constant metric.

⁴ Some literature regards different scenarios in the dataset as separate domains. However, we think the domain shifts among them are not sufficiently large, so that in this work we do not consider them as different domains.

5.3 Implementation Details

In all of our experiments, we utilize 300-dimensional GloVe word vectors [16] to initialize the embeddings [3]. Except that InductNet in accordance with its original setting uses a bidirectional LSTM [11] coupled with attention mechanism [1, 14] as the encoder (i.e., f_θ), all other models benefit from the CNN structure as suggested in [12]. Specifically, the CNN architecture is composed of three independent one-dimensional convolutional layers and the filter widths of these layers are respectively 3, 4, and 5. Considering overall parameter scale and efficacy, the number of filters in each convolution for RelationNet is set to be 50, while 100 for ProtoNet, MAML, and PtNet.

Furthermore, throughout the experiments, the size of query set in an episode is 15. We employ Adam optimizer with a learning rate 0.001 and a regularization coefficient 0.00001 for training. All trainable parameters are initialized with uniform distribution. The patience epoch number is set to 5 for early stopping. The α is 20.

Table 2. In-domain comparison results (%) under 5-way 1-shot, 5-shot, and 10-shot settings. Results in **bold** are the best performing ones under each setting.

Model	Home			Banking		
	1-shot	5-shot	10-shot	1-shot	5-shot	10-shot
InductNet	63.19 ± 0.41	71.67 ± 0.31	74.90 ± 0.29	76.72 ± 0.38	85.00 ± 0.27	85.41 ± 0.25
RelationNet	63.38 ± 0.41	74.81 ± 0.33	73.19 ± 0.34	81.31 ± 0.35	88.00 ± 0.26	89.57 ± 0.23
MAML	58.58 ± 0.38	68.44 ± 0.35	71.01 ± 0.32	69.51 ± 0.39	81.58 ± 0.29	84.03 ± 0.26
ProtoNet	67.91 ± 0.39	82.92 ± 0.26	86.15 ± 0.22	82.59 ± 0.31	92.20 ± 0.17	93.44 ± 0.14
PtNet	63.82 ± 0.38	83.32 ± 0.23	86.63 ± 0.20	75.83 ± 0.34	89.80 ± 0.20	92.08 ± 0.16

Table 3. Cross-domain comparison results (%) under 5-way 1-shot, 5-shot, and 10-shot settings. Results in **bold** are the best performing ones under each setting.

Model	Home2Banking			Banking2Home		
	1-shot	5-shot	10-shot	1-shot	5-shot	10-shot
InductNet	46.15 ± 0.36	54.64 ± 0.33	54.52 ± 0.31	44.54 ± 0.34	57.78 ± 0.32	64.98 ± 0.31
RelationNet	43.55 ± 0.35	56.89 ± 0.32	55.26 ± 0.31	42.10 ± 0.34	55.00 ± 0.32	57.19 ± 0.30
MAML	44.42 ± 0.34	52.07 ± 0.36	35.90 ± 0.31	37.53 ± 0.30	46.31 ± 0.31	44.51 ± 0.33
ProtoNet	56.90 ± 0.34	79.23 ± 0.28	82.90 ± 0.24	54.62 ± 0.35	78.32 ± 0.27	82.02 ± 0.24
PtNet	50.78 ± 0.34	77.14 ± 0.27	84.35 ± 0.21	58.87 ± 0.36	81.34 ± 0.26	84.95 ± 0.22

5.4 Comparison Results

Table 2 displays the results of in-domain comparison while the cross-domain comparison results are presented in Table 3. A key observation is that models performing competitively on two in-domain datasets perform less encouragingly on cross-domain datasets, implying that XFEW is a challenging task that needs to be further studied.

PtNet with forthright learning regime could achieve comparable results with rather complicated models on in-domain datasets, urging the necessity of revisiting current few-shot text classification approaches. PtNet can outperform other models by large margins on cross-domain datasets. This phenomenon demonstrates that PtNet has better cross-domain generalization ability than the existing models. When the shot number is small, PtNet can be sub-optimal, however, the margin is alleviated when the shot number becomes larger.

We conjecture the reason why PtNet performs better is that episode-based training scheme may limit the discriminative capacity on unseen classes. Particularly, PtNet is trained to discriminate more classes than other episode-based models (with only 5-way at one episode). Thus, when a model is directly applied to novel classes, it is easier for PtNet to make class centroids far from each other.

6 Conclusion and Future Work

In this paper, we investigate a new NLP task namely cross-domain few-shot text classification (XFEW). We find that although existing systems could reach good results on in-domain datasets, they struggle to yield competitive results on cross-domain datasets. Therefore, we propose a pre-training based simple baseline for the XFEW task, which achieves largely better results than other models on cross-domain datasets.

Based on the empirical results, we believe there are two promising directions needed to be explored. The first is to take a new perspective for few-shot learning which goes beyond episode-base training scheme. The second is to improve cross-domain generalization ability of few-shot text classification from the perspective of low-resource domain adaptation.

Acknowledgement. This work is supported by the National Key Research and Development Program of China (grant No. 2018YFC0831704) and Natural Science Foundation of China (grant No. U1636203).

References

- Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: ICLR (2015)
- Bao, Y., Wu, M., Chang, S., Barzilay, R.: Few-shot text classification with distributional signatures. In: ICLR (2019)
- Bengio, Y., Ducharme, R., Vincent, P.: A neural probabilistic language model. In: NeurIPS, pp. 932–938 (2000)
- Casanueva, I., Temclin, T., Gerz, D., Henderson, M., Vulic, I.: Efficient intent detection with dual sentence encoders. arXiv [arXiv:2003.04807](https://arxiv.org/abs/2003.04807) (2020)
- Chen, W.Y., Liu, Y.C., Kira, Z., Wang, Y.C.F., Huang, J.B.: A closer look at few-shot classification. In: ICLR (2019)
- Chen, Y., Wang, X., Liu, Z., Xu, H., Darrell, T.: A new meta-baseline for few-shot learning. arXiv [arXiv:2003.04390](https://arxiv.org/abs/2003.04390) (2020)

7. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML, vol. 70, pp. 1126–1135 (2017)
8. Gao, T., Han, X., Zhu, H., Liu, Z., Li, P., Sun, M., Zhou, J.: FewRel 2.0: towards more challenging few-shot relation classification. In: EMNLP/IJCNLP, no. 1, pp. 6249–6254 (2019)
9. Geng, R., Li, B., Li, Y., Zhu, X., Jian, P., Sun, J.: Induction networks for few-shot text classification. In: EMNLP-IJCNLP, no. 1, pp. 3902–3911 (2019)
10. Guo, Y., Codella, N.C.F., Karlinsky, L., Smith, J.R., Rosing, T., Feris, R.S.: A new benchmark for evaluation of cross-domain few-shot learning. arXiv [arXiv:1912.07200](https://arxiv.org/abs/1912.07200) (2019)
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
12. Kim, Y.: Convolutional neural networks for sentence classification. In: EMNLP, pp. 1746–1751 (2014)
13. Liu, X., Eshghi, A., Swietojanski, P., Rieser, V.: Benchmarking natural language understanding services for building conversational agents. arXiv [arXiv:1903.05566](https://arxiv.org/abs/1903.05566) (2019)
14. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: EMNLP, pp. 1412–1421 (2015)
15. Mishra, N., Rohaninejad, M., Chen, X., Abbeel, P.: A simple neural attentive meta-learner. In: ICLR (2018)
16. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: EMNLP, vol. 14, pp. 1532–1543 (2014)
17. Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning. In: NeurIPS, pp. 4077–4087 (2017)
18. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H.S., Hospedales, T.M.: Learning to compare: relation network for few-shot learning. In: CVPR, pp. 1199–1208 (2018)
19. Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., Wierstra, D.: Matching networks for one shot learning. In: NeurIPS, pp. 3630–3638 (2016)
20. Yu, M., et al.: Diverse few-shot text classification with multiple metrics. In: NAACL-HLT, pp. 1206–1215 (2018)



Shared Component Cross Punctuation Clauses Recognition in Chinese

Xiang Liu, Ruifang Han, Shuxin Li, Yujiao Han, Mingming Zhang,
Zhilin Zhao, and Zhiyong Luo^(✉)

School of Information Science, Beijing Language and Culture University,
Beijing 100083, China
luo_zy@blcu.edu.cn

Abstract. NT (Naming-telling) Clause Complex Framework defines the clause complex structures through component sharing and logic-semantic relationships. In this paper, we formalize component sharing recognition as a multi-span extraction problem in machine learning. And we propose a model with mask strategy to recognize the shared components of cross punctuation clauses based on pre-training models. Furthermore, we present a Chinese Long-distance Shared Component Recognition Dataset (LSCR) with four domains, including 43k texts and 156k shared components that need to be predicted. Experimental results and analysis show that our model outperforms previous methods in large margin. All the codes and dataset are available at <https://github.com/smiletm/LSCR>.

Keywords: Deep learning · Shared component recognition · Dataset · Pre-training

1 Introduction

The phenomenon of cross punctuation clauses component sharing often occurs in Chinese text processing. This long-distance shared component often plagues many Natural Language Processing (NLP) downstream tasks [6]. Although we blindly collect a large amount of data and increase the width and depth of the pre-training model today, there are hardly very effective solutions to this difficulty.

Table 1. Three punctuation clauses

①他找了个女朋友，②特别漂亮，③心里很开心。

He finds a girlfriend who is very beautiful, and he is very happy.

As shown in Table 1, there are three punctuation clauses. If the last two punctuation clauses are analyzed independently, it can be found that they are incomplete. This situation of missing shared component is very common in Chinese texts. It also causes a lot of difficulties for many NLP tasks. For example,

in Chinese-English machine translation, the phenomenon of long-distance component sharing often causes the existing translation systems to fail to accurately identify the correct target of the utterance (Naming or Telling), which affects the result of the translation. We have conducted a survey in which we input this simple sentence 他找了个女朋友，特别漂亮，心里很开心。(He finds a girlfriend who is very beautiful, and he is very happy.) to the current mainstream Chinese-English translation engines, such as Google, Baidu, and Bing etc. The translation results are shown in the upper part of Table 2. From the perspective of human cognition, it is easy to correspond 特别漂亮, (very beautiful) with 女朋友 (girlfriend), and 心里很开心。 (very happy) with 他 (he). However, the translations of these engines are too blunt to express the corresponding Naming-Telling relationship in the source text. We have been thinking about how to combine the rules of Chinese language with the deep learning model to improve this phenomenon, instead of blindly increasing the complexity of the model and collecting more corpus without knowledge. Assuming that we can fill in the missing components (Naming or Telling) of every small punctuation sentence, can this phenomenon be alleviated? Therefore, we fill in the missing components of each small punctuation sentence in the previous sentence, and then input the whole sentence into the translation engine, and the result is shown in the lower part of Table 2. Obviously, it can be found that most of the current translation engines can carry out good translations results for the completed Chinese text. This process might look a bit like turning a long sentence into short sentence. At least, compared with the previous studies, it alleviates the problem of ambiguity caused by long-distance sharing components, but there is still a certain distance from achieving the truly authentic English.

Table 2. The upper part is the translation result of the original input. The lower part is the translation result after filling shared components.(2021.6.1)

Sentence1: 他找了个女朋友，特别漂亮，心里很开心。 (He finds a girlfriend who is very beautiful, and he is very happy.)	
Google	He found a girlfriend, who was very beautiful and very happy.
Baidu	He found a girlfriend, especially beautiful, very happy.
Bing	He found a girlfriend, especially beautiful, very happy.
Youdao	He found a girl friend who was very beautiful and very happy.
Sentence2: 他找了个女朋友，女朋友特别漂亮，他心里很开心。 (He finds a girlfriend who is very beautiful, and he is very happy.)	
Google	He found a girlfriend, the girlfriend is very beautiful, he is very happy.
Baidu	He found a girlfriend, who was very beautiful, and he was very happy.
Bing	He found a girlfriend, girlfriend is particularly beautiful, he is very happy.
Youdao	He found a girl friend, the girl friend is very beautiful, he is very happy.

Based on the problem of the sharing of long-distance components in Chinese, the concept of Naming and Telling was proposed in [15] to define the concept of Naming Structure and formed NT clause theory [14,16]. As show in Table 3, the

NT clause theory contains four main pattern: Stack Pattern, New branch Pattern, Postposition Pattern and Influx Pattern. The theory of NT clause reveals the organization form of Chinese text in micro topic level, and proves its high coverage and operability in a number of corpus [5,11,12].

Naming and Telling. The concepts of Naming and Telling belong to the pragmatic category. A Naming is the starting point of an utterance, while a Telling is the description of the Naming. A Naming is usually a word or a phrase in a punctuation clause or the whole punctuation sentence. Refer to the top of every pattern in Table 3 for more details.

NT Clause. The combination of a Naming and one of its Tellings is called a NT clause. In many punctuation clauses, the Naming is missing or Telling is incomplete, which will result in the incompleteness of the punctuation clause semantics. Based on the context, the NT clause completes the missing component of the punctuation clause, and restores the original semantics as much as possible. Refer to the bottom of every pattern in Table 3.

In the previous work, we constructed a Chinese Clause Complex Bank (CCCB) based on the theory of Chinese clause complexes. It is a two-dimensional tagging format with newline indentation, which covers news, novels, government work reports, and encyclopedic, and contains more than 50w characters and more than 1.4w NT-Clauses.

In this work, we constructed a LSCR dataset for learning and research. We manually cleaned the CCCB corpus, and formed a dataset of about 5w by concatenating multiple consecutive clause complexes in context. Then we present an end-to-end model to identify shared components.

Our contributions can be summarized as below:

- We build a LSCR dataset based on the original manually labeled CCCB, which can be used for researches on Chinese-English machine translation, Chinese information extraction, etc., and served as a benchmark for studying sentence level Co-reference Resolution. The LSCR dataset will continue to be maintained with the CCCB labeled dataset expanding.
- We present an end-to-end model for the long-distance shared component recognition. Moreover, we conducted a series of experiments on the current mainstream pre-training models.

2 Related Work

So far, many researchers have studied the long-distance component sharing relationship based on the Chinese Clause Complex theory. These approaches can be mainly divided into two categories: one employs traditional machine learning [6-9] and the other deep learning [17]. But these methods just based on a single pattern (Stack Pattern). For example, [17] proposed a neural network model based on Attention and LSTM to recognize the sharing components of Stack Pattern. These methods are too limited to be universal. In order to solve this

Table 3. Four main Chinese Clause Complex patterns and corresponding Chinese NT clauses

Stack Pattern (堆棧模式) 吴之荣又碰了一鼻子灰。 Zhirong Wu was rejected again, □□□眼见回家已无盘缠， soon had no money to go home, □□□□势将流落街头。 would wander outside.	New branch Pattern (新支模式) 杰克盖好了房子。 Jack built the house, □□里面放着糖， sugar in it, □□ 很放心。 was relieved.
NT Clause (NT小句) 吴之荣又碰了一鼻子灰。 Zhirong Wu was rejected again, 吴之荣眼见回家已无盘缠， Zhirong Wu soon had no money to go home, 吴之荣眼见势将流落街头。 Zhirong Wu soon would wander outside.	NT Clause (NT小句) 杰克盖好了房子。 Jack built the house, 房子里面放着糖， sugar in the house , 杰克很放心。 Jack was relieved.
Postposition Pattern (后置模式) □□户口不迁来， did not transfer residence, □□再没有个娃娃， has no baby, 女人迟早回家。 the woman will go home sooner.	Influx Pattern (汇流模式) 白起利用赵括只善纸上谈兵□□□， Qi Bai use that Kuo Zhao only good at empty talk, □□□□□缺乏作战经验的弱点。 the weakness of lacking combat experience.
NT Clause (NT小句) 女人户口不迁来， the woman did not transfer residence， 女人再没有个娃娃， the woman has no baby， 女人迟早回家。 the woman will go home sooner.	NT Clause (NT小句) 白起利用赵括只善纸上谈兵的弱点， Qi Bai use the weakness Kuo Zhao only good at empty talk， 白起利用赵括缺乏作战经验的弱点。 Qi Bai use the Kuo Zhao's weak- ness of lacking combat experience.

problem, we rethink the Chinese Clause Complex theory and propose a more general method (see Subsect. 4).

In recent years, pretrained language models, such as BERT [3] and GPT [13], have been successful in many natural language processing tasks. This approach is very effective, because these models are able to effectively encode contextual information through their attention mechanism and adapt to a variety of NLP tasks with a large amount of training data. And subsequent models, such as UniLM [4], Big Bird [19] and so on, have further improved the attention mechanism for different tasks on the basis of BERT. This work combines the pre-trained language models to research the long-distance component sharing relationship.

3 Dataset

We did a lot of cleaning and proofreading work on CCCB and finally got the LSCR dataset. First, we convert the CCCB manually labeled into a form that can be computed through its labeled rules. Then, in the process of cleaning, we used its context relationship to concat previous or next sentences to expand sample, and finally got the LSCR dataset. The dataset is all in simplified Chinese format, which covers the four major areas of novels, news, encyclopedia entries, and government work reports. It has 43944 texts, 156816 shared component predictions. The details of LSCR are as shown in Table 4.

Table 4. The details of dataset in different splits of LSCR. The Count is the number of text. The Total is the total of all types (Stack, New branch, Postposition, Influx) of shared components that need to be predicted.

Split	Count	Stack	New branch	Postposition	Influx	Total
Train	29493	92640	6143	2347	1389	102519
Vail	8037	29118	1769	714	344	31945
Test	6414	20312	1188	499	353	22352
Total	43944	142070	9100	3560	2086	156816

3.1 Details

Table 5. The X is the input text. The P represents the position. In this example, $P = 1$ represents that there is missing shared component near 通(tong). S represents the missing shared component at the P position. In here, It's missing 韩国 (South Korea). The start and end positions of the 韩国 are 13 and 14 respectively in this text.

$X:$ “通过这些努力，可喜的是，韩国对这些国家的出口额增加了30%，” (Through these efforts, it is gratifying that South Korea’s exports to these countries have increased by 30%.) $Y:\{P=1, S=\{\text{start}=13, \text{end}=14, \text{seg}=\text{“韩国”}\}\}$
--

According to our analysis of CCCB and Chinese text, it's easily found that the missing contents often span multiple punctuation sentences. This may be like a multi-hop extractive task. In order to hold the information of the original sentence as much as possible, this problem is simplified in LSCR, and we only need to predict the nearest shared component segment. In addition, we also found that most of the missing positions of shared components are at the beginning or end of punctuation sentences. Therefore, the missing positions of shared components given in LSCR dataset are all at the beginning or end of punctuation sentences. The position in the middle of the sentence is not covered. We simply made a formal definition for the LSCR dataset, which defined as $X = \{T\}$, $Y = \{P, S\}$ in Table 5. Where T means text, P means the position of the missing shared component, S means the missing shared component at the P position.

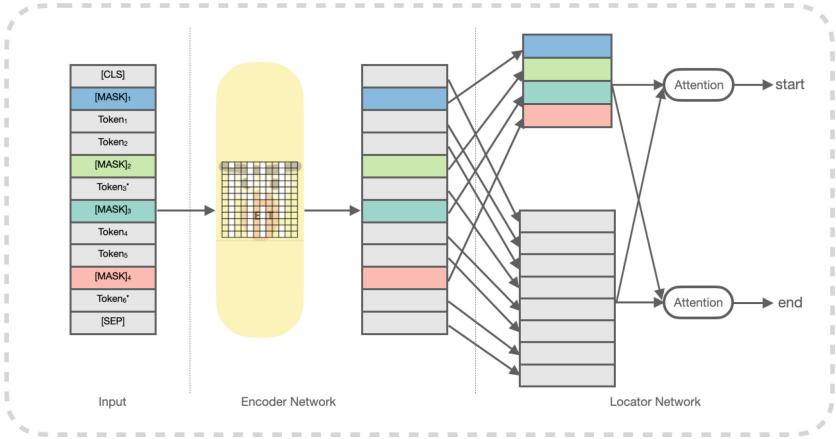


Fig. 1. The overall architecture of our model. The “*” represents this token may be a punctuation mark.

4 Methodology

4.1 Overall Architecture

In this section, we will describe our model in detail. We have proposed a novel neural network architecture to recognize sharing components. The overall architecture of our model is shown in Fig. 1, which consists of encoder network and locator network. The encoder network aims to obtain contextual information, which is based on BERT. The locator network aims to find positions of the missing contents. We will detail each component below. At the end of this section, we will describe the model learning.

Encoder Network. The encoder network is a gathered contextual informations model based on BERT. The input is the sequence of tokens embeddings $E = (e_{m1}, e_{t1}, e_{t2}, e_{m2}, \dots, e_{tn}, e_{mk})$, where e_{tn} denotes the embedding of token tn , e_{mk} denotes the embedding of mask mk , which is the sum of word embedding, position embedding, and segment embedding of the character, as in BERT. The output is the output of last Transformer encoder block of BERT, $E' = (e'_{m1}, e'_{t1}, e'_{t2}, e'_{m2}, \dots, e'_{tn}, e'_{mk})$. Then the output will be passed into the Locator Network.

BERT consists of a stack of 12 Transformer encoder blocks taking the entire sequence as input. Each block contains a Multi-Head Attention mechanism, which consists of several attention layers running in parallel. In our model, The fully connected attention matrix of BERT is replaced by a sparse attention matrix, which called Fence-Attention. Then output followed by a feed-forward network. These are defined as:

$$\begin{aligned} & \text{MultiHead}(Q, K, V) \\ &= \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \end{aligned} \quad (1)$$

$$\text{head}_i = \text{FenceAttention}(QW_i^Q, KW_i^K, VW_i^V) \quad (2)$$

$$FNN(X) = \max(0, XW_1 + b_1)W_2 + b_2 \quad (3)$$

where Q, K, V are the same matrices representing the input sequence or the output of the previous block, MultiHead, FenceAttention, and FNN denote multi-head self-attention, self-attention, and feed-forward network respectively, W_i^Q, W_i^K, W_i^V are parameters of every head_i , W^O, W_1, W_2, b_1, b_2 are parameters of linear transformation.

Our calculation of attention function is a bit different from that in BERT. We add Fence-MASK in it as follows:

$$\begin{aligned} & \text{FenceAttention}(Q, K, V) \\ &= \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + \text{Fence-MASK}\right)V \end{aligned} \quad (4)$$

where d_k is the dimension of keys and acts as scaling factor, Q, K, V are the products of different linear transformations of the same matrices.

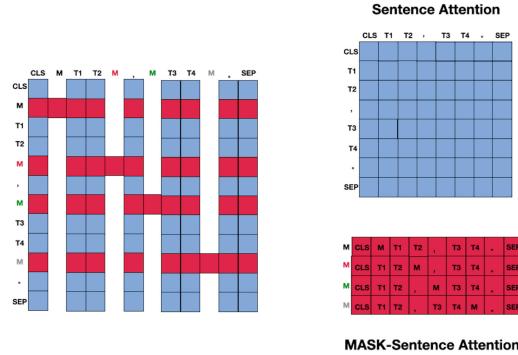


Fig. 2. FenceAttention. The left is Fence-MASK. The Global Attention is like being cut into two parts, Sentence Attention and MASK-Sentence Attention.

FenceAttention. This attention mechanism is shaped a bit like Fence, so we call it as Fence-Attention which is a local attention mechanism used to remove unnecessary informations. In the left of the Fig. 1, we insert some [MASK] in the sentence as the input. Intuitively, if we insert too many [MASK] in the input, this will add some noises that produced by inserted [MASK], and interfere with the model learning useful information from context. The Fence-Attention is as shown in Fig. 2.

The Fence-MASK is as shown on the left of the Fig. 2, and we let the inserted [MASK] just focuses on itself and the context, instead of other inserted [MASK], and other parts only focus on the context. In this way, we split the Global attention into the Sentence attention and Mask-Sentence attention through the Fence-MASK, thus reducing unnecessary noise from inserting extra [MASK]. The FenceAttention is constructed by the Fence-MASK.

Locator Network. The locator network is an attention span network. The input is the output of the encoder network $E' = (e'_{m1}, e'_{t1}, e'_{t2}, e'_{m2}, \dots, e'_{tn}, e'_{mk})$, where e'_{tn} and e'_{mk} respectively denote the embedding of the token n and mask mk through the encoder network. Then the embedding of token and mask are extracted separately, defined as:

$$H_t^s = \text{extract}(E'W_s + b_s, Pos_t) \quad (5)$$

$$H_m^s = \text{extract}(E'W_s + b_s, Pos_m) \quad (6)$$

$$H_t^e = \text{extract}(E'W_e + b_e, Pos_t) \quad (7)$$

$$H_m^e = \text{extract}(E'W_e + b_e, Pos_m) \quad (8)$$

where E' is the output of the encoder network, Pos_t and Pos_m respectively represents the positions of tokens and masks, W_s , W_e , b_e , b_e are parameters, the script s and e respectively represent the start and the end, extract is a gather function that based positions to collect vectors.

Then the H_m and H_t of start and end respectively input to the attention and softmax to get the position probabilities of start and end of each mask missing content, which is defined as

$$P_c(m_i^s = s_i | X) = \text{softmax}(H_m^s \cdot H_t^{sT})[s_i] \quad (9)$$

$$P_c(m_i^e = e_i | X) = \text{softmax}(H_m^e \cdot H_t^{eT})[e_i] \quad (10)$$

where $P_c(m_i^s = s_i | X)$ and $P_c(m_i^e = e_i | X)$ is the conditional probability that the position of the missing contents m_i is (s_i, e_i) in the input X , softmax is the softmax function.

4.2 Learning

The learning of our model is conducted end-to-end, provided that BERT is pre-trained and training data is the same input format.

The learning process is divided into two parts to optimize, which correspond to the start position and the end position recognition of the missing contents.

$$L = -\frac{1}{2} \left(\sum_{i=1}^n \log P_c(m_i^s = s_i | X) + \sum_{i=1}^n \log P_c(m_i^e = e_i | X) \right) \quad (11)$$

5 Experimental Results

5.1 Experiment Setting

Many recent studies show that pre-trained language models (PLMs) have become a powerful encoder in many downstream task. We also use PLMs (bert-base-chinese¹, chinese-bert-wwm², chinese-roberta-wwm³ [2]) as the backbone encoder in our model and initialize with the corresponding pre-trained cased weights. In all of our examples, we train 4 epochs of train set, then evaluate on the dev set to select the best model and evaluate it on the test set. The hidden size is 768, the max input length is 512, the number of layers and heads is 12 and the input batch is 16. Models are implemented by Tensorflow2.X framework [1] and Huggingface transformers [18]. All models are optimized by AdamW [10] with the learning rate of 5e-5. Moreover, according to our experiments, we found that the model can obtain better results when the weight of the loss at the missing component and the loss at the non-missing component is 5 : 1. We thus set $w = 5$ in all experiments ($loss = loss_{non} + w * loss_{miss}$). All the experiments are conducted with CUDA on one NVIDIA 24GB-TITAN RTX.

We use Sentence Accuracy(S), Precision(P), Recall(R) and F1-score($F1$) as metrics to evaluate the model performance in LSCR. Where S is targeted for the whole sentence, and only when all missing components in the whole sentence are predicted correctly can it be positive, while $P, R, F1$ is targeted for each missing component in the sentence.

5.2 Main Results

We have evaluated several current mainstream pre-trained models on LSCR, and the results are shown in the Table 6. In the horizontal comparison, RoBERTa-wwm-ext+FA (Fence Attention) got the best results. (Where +FA means that Fence-Attention is added to the original model.) But the highest in S is only 51.36%, and the highest in $F1$ is only 73.00%. As the previous contents introduced, the task of identifying shared components is very difficult. On the one hand, we need to find the position of predicting, on the other hand we need the model to predict the missing content.

5.3 Ablation Study

We carried out ablation study on the way of inserting [MASK] to predict and adding FA. Table 7 shows the results on LSCR test set. In the table, -m represents not inserting [MASK] in input text to make predictions, thus the attention mask mechanism is the same as BERT, just using the Chinese characters at the beginning and end of each punctuation sentence to make predictions. From comparison of BERT-base -m vs BERT-base, BERT-wwm-ext-base -m vs BERT-wwm-ext-base and RoBERTa-wwm-ext -m vs RoBERTa-wwm-ext, we find the way of

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

² <https://huggingface.co/hfl/chinese-bert-wwm-ext>.

³ <https://huggingface.co/hfl/chinese-roberta-wwm-ext>.

Table 6. Results of models on LSCR test dataset.

Models	S	P	R	F1
BERT-base+FA	50.09	73.63	69.42	71.46
BERT-wwm-ext-base+FA	50.14	73.11	69.00	70.99
RoBERTa-wwm-ext+FA	51.36	74.18	71.85	73.00

Table 7. Ablation study results of models on LSCR test dataset.

Models	S	P	R	F1
BERT-base -m	49.39	72.43	69.01	70.68
BERT-base	49.95	72.63	69.32	70.93
BERT-base+FA	50.09	73.63	69.42	71.46
BERT-wwm-ext-base -m	50.39	73.40	69.68	71.49
BERT-wwm-ext-base	51.51	73.85	70.28	72.02
BERT-wwm-ext-base+FA	50.14	73.11	69.00	70.99
RoBERTa-wwm-ext -m	51.90	73.88	71.82	72.84
RoBERTa-wwm-ext	51.70	73.95	71.60	72.76
RoBERTa-wwm-ext+FA	51.36	74.18	71.85	73.00

inserting [MASK] in input text can improve precision of model. Adding FA can further improve F1 of BERT-base and RoBERTa-wwm-ext. Moreover, after adding FA, the RoBERTa-wwm-ext got the best results on the LSCR test set.

5.4 Discussions

We observed that our method is able to make effective use of global context information. For example, there is a Chinese Clause Complex “我们以为校长和课本都是在胡说八道。发现了煤块中的松香，才明白校长没有骗我们，课本也没有骗我们。”

(We thought the principal and the textbooks were talking nonsense. After discovering the rosin in the coal, I realized that the principal did not lie to us, nor did the textbook lie to us.). Each missing components, in the sentence, is shown in Table 8. The punctuation sentences “发现了煤块中的松香，” (discovered the rosin in the coal,), “才明白校长没有骗我们，” (realized that the principal did not lie to us) and “课本也没有骗我们。” (nor did the textbook lie to us.) are semantically incomplete sentences, which all miss some contents. The first two missing subject “我们” (we). The last misses subject, adverb and verb “我们才明白” (Then we realized). We analyze attention matrix of the last transformer encoder block in Encoder(Bert-base), as show in Fig. 3, and find that the addition of the Fence-MASK enables the model to concentrate useful information on the [MASK] more effectively.

Table 8. Each missing components of the sentence

我们以为校长和课本都是在胡说八道。
 才发现了煤块中的松香，
 才明白校长没有骗我们，
 课本也没有骗我们。

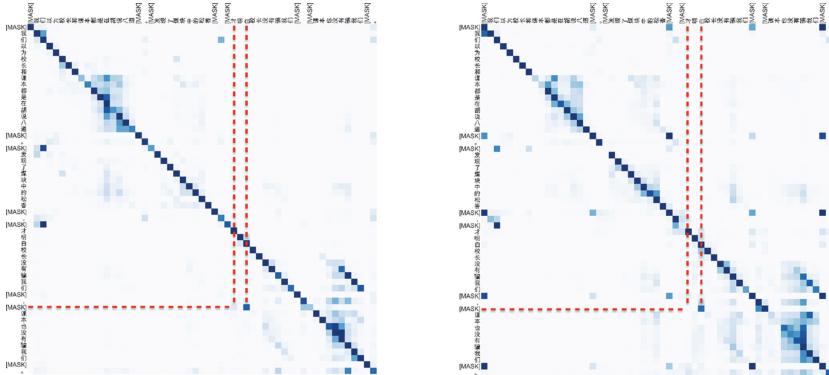


Fig. 3. The attention visual analysis of the last transformer encoder block in the Encoder (Bert-base). The [MASK] of the left figure can pay more attention to 才 than the right. The darker the color, the greater the weight there. (**Left**) with the Fence-MASK. (**Right**) without the Fence-MASK.

6 Conclusion

In this paper, we propose LSCR, a shared component recognition dataset. This is the first Chinese sentence-level shared component recognition dataset. There are few datasets on Chinese sentence-level shared component recognition. This dataset can be used to research in this area. By using mainstream PLMs as backbone encoder, we carry out a series of experiments on LSCR, demonstrating that sentence-level shared component recognition remains a challenging problem and worth exploring. The LSCR will continue to be updated with the CCCB expanding. We hope these works can promote the development of many NLP downstream tasks, such as Reading Comprehension, Machine Translation, Information Extraction, Search, and so on.

Acknowledgements. We would like to thank the anonymous reviewers for their valuable comments. Thanks to all the members who participated in this project, especially the annotators of CCCB. This work is supported by the National Natural Science Foundation of China (No. 62076037).

References

1. Abadi, M., et al.: TensorFlow: a system for large-scale machine learning (2016)
2. Cui, Y., Che, W., Liu, T., Qin, B., Wang, S., Hu, G.: Revisiting pre-trained models for Chinese natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pp. 657–668. Association for Computational Linguistics (2020)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of Deep Bidirectional Transformers for Language Understanding (2019)
4. Dong, L., et al.: Unified language model pre-training for natural language understanding and generation (2019)
5. Ge, S., Song, R.: The naming sharing structure and its cognitive meaning in chinese and english. In: Proceedings of the 2nd Workshop on Semantics-Driven Machine Translation (SedMT 2016) (2016)
6. Jiang, Y., Song, R.: Topic clause identification based on generalized topic theory. *J. Chin. Inf. Process.* **26**, 114–119 (2012)
7. Jiang, Y., Song, R.: Topic structure identification of pclause sequence based on generalized topic theory. In: Zhou, M., Zhou, G., Zhao, D., Liu, Q., Zou, L. (eds.) NLPCC 2012. CCIS, vol. 333, pp. 85–96. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34456-5_9
8. Jiang, Y., Song, R.: Optimization of candidate topic clause evaluation function in topic clause identification (2014)
9. Jiang, Y., Song, R.: Topic clause identification method based on specific features. *J. Comput. Appl.* **34**, 1345–1349 (2014)
10. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization (2019)
11. Lu, D.: A study of syntactic constituent and semantic role of new branch topic. In: Proceedings of the 19th Chinese National Conference on Computational Linguistics. Chinese Information Processing Society of China (2020)
12. Lu, D., Song, R., Shang, Y.: Cognitive complexity of topic in chinese text based on generalized topic structure theory (2014)
13. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training, p. 12 (2018)
14. Song, R.: Stream model of generalized topic structure in chinese text (2013)
15. Song, R.: Chinese clause complex and the naming structure (2018)
16. Song, R., Jiang, Y., Wang, J.: On generalized-topic-based chinese discourse structure (2010)
17. Mao, T., Zhang, Y., Jiang, Y., Zhang, Y.: Research on construction method of chinese nt clause based on attention-LSTM. In: Zhang, M., Ng, V., Zhao, D., Li, S., Zan, H. (eds.) NLPCC 2018. LNCS (LNAI), vol. 11109, pp. 340–350. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99501-4_30
18. Wolf, T., et al.: Transformers: state-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (2020)
19. Zaheer, M., et al.: Big bird: transformers for longer sequences (2020)



BERT-KG: A Short Text Classification Model Based on Knowledge Graph and Deep Semantics

Yuyanzhen Zhong¹(✉), Zhiyang Zhang², Weiqi Zhang², and Juyi Zhu²

¹ Shenzhen University Webank Institute of FinTech, Shenzhen, China

² University of Electronic Science and Technology of China, Chengdu, China

Abstract. Chinese short text classification is one of the increasingly significant tasks in Natural Language Processing (NLP). Different from documents and paragraphs, short text faces the problems of shortness, sparseness, non-standardization, etc., which brings enormous challenges for traditional classification methods. In this paper, we propose a novel model named BERT-KG, which can classify Chinese short text promptly and accurately and overcome the difficulty of short text classification. BERT-KG enriches short text features by obtaining background knowledge from the knowledge graph and further embeds the three-tuple information of the target entity into a BERT-based model. Then we fuse the dynamic word vector with the knowledge of the short text to form a feature vector for short text. And finally, the learned feature vector is input into the Softmax classifier to obtain a target label for short text. Extensive experiments conducted on two real-world datasets demonstrate that BERT-KG significantly improves the classification performance compared with state-of-the-art baselines.

CCS Concepts: Computer methodologies · Artificial intelligence · Natural language processing · Lexical semantics

Keywords: Short text classification · Knowledge graph · BERT-based model

1 Introduction

In the last decade, the rapid development of Internet technology accelerates the development of mobile social network platforms, such as Weibo, Twitter, etc. These platforms generate more and more Chinese short text messages, e.g., online news, instant messaging, and user comments. Understanding these short texts are useful in a wide range of applications, e.g., information retrieval, text recommendation, and relation extraction. As the fundamental task in understanding short texts, short text classification has attracted significant attention in both academia and industry. Compare with documents and paragraphs, short text faces the difficulty of shortness, sparseness, non-standardization, and noises [1]. How to effectively extract text features and choose an appropriate classification model to classify short text has become a difficult problem in current research.

To address the aforementioned limitations of short text, most existing short text classification methods use pre-trained static word vectors to represent short text features,

e.g., Word2Vec and GloVe. However, these methods based on static word vectors ignore the current context and have limited ability to understand non-standardization of short text. To solve this problem, Peter et al. [4] proposes Embeddings from Language Models, ELMo which uses large-scale corpus to train a two-way language model, and then combines the hidden layer features to get the dynamic word vector. This model can not only express the grammatical and semantic features of words, but also change according to the semantics of context. Furthermore, Alec et al. [5] proposes a generative pre-training (GPT) model, capturing more text semantic information with the cost of computing speed decreases compared with ELMo. To solve the sparseness problem, existing approaches enrich the short text from external resources, such as *Wikipedia* and *knowledge bases*. These approaches, however, rely on a large number of external data, which cannot easily extend to some specific domains or languages.

In this paper, we propose a novel model named BERT-KG, which alleviates the sparseness and non-standardized problems by introducing the knowledge graph and BERT. Specifically, The main contributions of our BERT-KG are summarized as follows:

- BERT-KG enriches the features of short text by obtaining background knowledge from the knowledge graph and further embeds the three-tuple information of the target entity into a BERT-based model.
- By improving the input layer of the BERT model, we input short text and its background knowledge into the BERT model for short text classification.
- We designed a Whole Word Masking Transformer model for short text classification for the BERT model, so as to capture the boundary relationship between words from the given short text and background knowledge, and eliminate the influence of invisible words in the visible matrix at the same time.
- We evaluate our proposed approach based on two public real-world datasets. The results show that our method outperforms some common short text classification baselines in literature.

2 Related Work

In recent years, deep learning has made remarkable progress in many subfields of artificial intelligence. In the aspect of natural language processing, deep learning models are often used to explore and obtain better solutions to sparsity problems. In 2014, Kim [7] uses a convolutional neural network on sentence-level text classification task for the first time and proposes TextCNN model. The model uses the pre-trained word vector model as input, and the convolutional neural network to automatically learn text sentence-level features for text classification. Then, Zhang et al. [8] proposes CharCNN, a character-level convolutional neural network text classification model, which uses character vector as model input, and performs text classification based on the character-level features of the text extracted by the convolutional neural network. Convolutional neural network-based text classification model can quickly extract important features of text, but it ignores the word order and context information of text, resulting in information loss. In order to solve this problem, Lai et al. [9] proposes RCNN model, which constructs the context

of a word through a bidirectional RNN model, then splices with the word vector of the word, and convolutional neural network extracts text features for text classification. Lee et al. [10] applies RCNN to short text classification, considering the context of words in short text, and achieves good experimental results. Some scholars regard the word frequency in short text as features [11], and construct feature engineering to distinguish the semantics of feature words in short text to solve the non-standard problem of short text. However, this kind of method ignores the context, and has defects in capturing and understanding deep semantic information.

Jacob et al. [12] proposes the pre-training model of BERT, which also adopts a two-stage process. Moreover, BERT increases the data scale of the language model while adopting the bi-directional language model, and then fine-tunes the pre-training language model generated by BERT based on the dataset of downstream natural language processing tasks. Zhou Y et al. [13] proposes a Chinese short text classification method based on BERT, which uses BERT and training models to extract sentence-level feature vectors of short texts, and then applies them to short text classification.

In order to solve the problem of sparse features, Google [14] has introduced the concept of knowledge graph based on the Semantic Web, which is widely used in intelligent search and intelligent question answering. Subsequently, knowledge graphs such as freebase [15], DBpedia [16], Yago [17], and probase [11] have emerged in academic and industrial circles. By incorporating the semantic web structure, knowledge graph organizes knowledge in the form of a directed graph, which improves the way of its storage and acquisition. In addition, high-quality knowledge can be extracted from knowledge graph, which can be used to expand the features of short text and effectively solve the difficulty brought from sparse features of short text.

3 BERT-KG Model

3.1 Framework

We first present the general framework of the proposed BERT-KG with the basis of the knowledge graph and BERT. It aims to embed both context-aware information in a given short text and implicit information obtained from the knowledge graph into a unified representation, such a dense representation will be the input of BERT for the short text classification task.

As shown in Fig. 1, BERT-KG contains four components: (1) feature extraction layer, (2) knowledge extraction layer, (3) hybrid coding layer and (4) BERT model layer. According to these four parts, the short text and its implicit knowledge will be effectively integrated and embedded. Then, the BERT model is used to learn and absorb this information for short text classification. The detailed description of BERT-KG is as follows:

- (1) Feature extraction layer: We first make segmentation for short text and remove stop words. Then, we obtain a text sequence with a fixed length of n , which is recorded as $W = (w_1, w_2, \dots, w_n)$, and be used as the input representation of the subsequent BERT model. Next, the keywords of short text are extracted by TKE

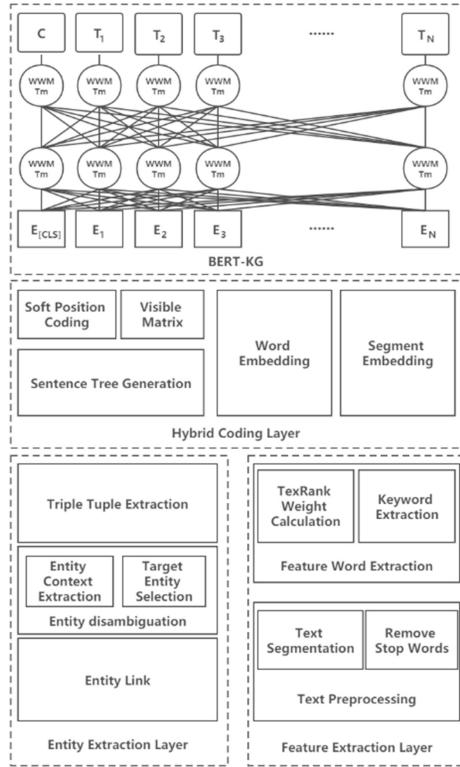


Fig. 1. Framework of BERT-KG

algorithm [18], and get the set of short text keywords set of length of H , which is denoted as $KW = (kw_1, kw_2, \dots, kw_n)$. In particular, it will be used for obtaining the background knowledge of the short text in the following knowledge extraction layer.

- (2) Knowledge extraction layer: BERT-KG enriches the characteristics of short text by exploiting the implicit knowledge of the short text from the knowledge graph. Specifically, we link the keywords in the short text entity, and then take the linked results as feature expansion items.
- (3) Hybrid coding layer: The hybrid coding layer is the core part of the model. We improve the input layer of the BERT model to input short text and its background knowledge into the BERT model for short text classification. Specifically, we first build a sentence tree inspired by Liu et al. [9] to generate an input sequence where the background knowledge is included in that tree and the sentence tree is transformed into a sequence containing entity relationships. Then, the position coding in the original model is replaced by the soft position coding, so that the knowledge in the sequence can be input into the BERT model. Besides, the visible matrix is built for avoiding the appearance of noise during knowledge input.
- (4) BERT model layer: BERT model layer can receive dense representation from our improved input layer. We improves the transformer model in the Bert model and

replaces it with Whole Word Masking Transformer model [6], so that the boundary relationship between words can be captured from the given short text and background knowledge, and the influence of invisible words in the visible matrix can be eliminated.

Since the feature extraction layer and knowledge extraction layer are implemented by TKE algorithm and CSEL algorithm, respectively, we will not repeat them here. Next, we will introduce in detail the sentence tree generation, soft position coding, visible matrix in the hybrid coding layer, and the Whole Word Masking Transformer model in the BERT layer.

3.2 Sentence Tree Generation

From the feature extraction layer, we can get the word sequence $W = (w_1, w_2, \dots, w_n)$ and the keyword set $KW = (kw_1, kw_2, \dots, kw_n)$ of the short text, where the keyword set is a subset of the corresponding set of the word sequence. Assuming that the index of the keyword kw_i in the original short text segmentation result is i , the corresponding target entity with w_i as the entity reference and its entity relationship triple can be obtained from the knowledge extraction layer, denoted as (w_i, r_j, e_k) , where w_i and e_k are the name of the entity and r_j is the relationship between the two entities. (w_i stands for multiple meanings).

According to the short text word sequence and keyword entity relation triple tuple, the process of generating the sentence tree is as follows: the entity e_k in the triple tuple is taken as the leaf node, and the relation entity corresponding to the keyword is taken as the subtree, which is connected with the keyword node. For the keywords with multiple triples, the depth of the one triplet subtree cannot exceed one. In this paper, sentence tree is a concrete representation of short text keywords, entities and the relationship between them: keywords are used as root nodes, entities are used as leaf nodes, and they are connected by relationships. This makes the input data attached with a priori feature information. Combined with the soft position coding operation later, it can improve the classification results. The sentence tree can be written as:

$$s_{tree} = \{w_1, w_2\{(r_{21}, e_{21}), (r_{22}, e_{22}), (r_{23}, e_{23})\}, \dots, w_i\{(r_{i1}, e_{i1}), (r_{i2}, e_{i2}), \dots, (r_{im}, e_{im})\}, \dots, w_n\}$$

E.g., given a short text sequence:

(苹果下调第一季度销售预期) (Apple Decrease First Quarter Sales Expectation), after adding the involved background knowledge, the sequence's sentence tree will be:

(苹果{(类型, 公司), (首席执行官, 蒂姆 库克),} 下调 第一季度 销售 预期) (Apple {(type, company), (CEO, Tim Cook), ...} Decrease First Quarter Sales Expectations). (Format problem).

Converting the above sequence directly into the character-level Token input of the BERT model will cause various noises. To this end, the input layer in the original BERT must be improved so that it can correctly fuse the short text structure information with the introduced external knowledge. The specific strategy after improving is to use soft location coding and involve the visible matrix.

3.3 Extended Short Text Location Coding

In the BERT model, location coding is used to supplement sentence structure information that cannot be captured by the self-attention mechanism. However, the original position coding method can not fully or even incorrectly obtain the sentence tree after introducing external knowledge. When Liu et al. [18] Input complex sentences, soft position coding has been proved to be able to input the sentence information into the model completely through a two-dimensional matrix. For this reason, BERT-KG uses soft location coding to encode the original short text Token sequence continuously by using the structure of sentence tree. For the knowledge introduced by keywords, BERT-KG will encode the knowledge according to the keyword Token encoding. For instance, given the sequence:

(苹, 果, 类, 型, 公, 司, 首, 席, 执, 行, 官, 蒂, 姆, 库, 克, ……, 下, 调, 第, 一, 季, 度, 销, 售, 预, 期).

A comparison diagram of the soft position coding and the original position coding is shown in Fig. 2:

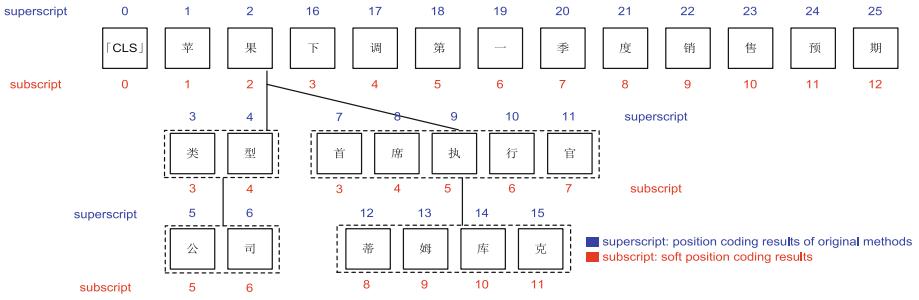


Fig. 2. Comparison of soft location coding and original position coding.

For the subscript digit encoding section in Fig. 2, using soft location coding can restore the correct syntax structure information of short text, while it also brings a new problem: there may be more than one token corresponding to the same location encoding. E.g., token with location code “3” has (下), (类), and (首). Similarly, Token with location code “4”, “5”, “6” … “11” also has many Tokens. If only soft locations are used, the model will produce token problems after (下) that may be (调), (席), or (型), affecting the correct understanding of semantics. Therefore, it is very necessary for us to further solve the contextual multi-Token correspondence problem of soft location coding by introducing a visible matrix. The detailed strategy will be described in detail below.

One difficulty in soft position coding is that position coding generates multiple corresponding tokens, which prevents the model from understanding the correct semantics. Take the sequence shown in Fig. 2 as an example, there are three Tokens coded as “3”, and the Token that ultimately reflects the correct semantics should be (下), not (类) or (首) in the introduced external knowledge, while the Token appearing after (下) should be (调), not (型) or (席) in the introduced external knowledge. In summary, the disturbing factors to the model understanding of Token itself and its context come from the introduced external knowledge. Furthermore, according to the sentence tree structure, the token of external knowledge is all in the branch of the sentence tree, thereby Token that introduces external knowledge can be shielded when understanding the original short text context. In

addition, Token that is originally short text can be shielded when understanding external knowledge. As such, we can tackle the above problem by preventing incorrect semantic understanding.

Visible matrix makes the embedding of a Token come only from the context of the same branch in the sentence tree according to the structure of the sentence tree, shielding the influence of Token between different branches. Visible matrices can be formalized as:

$$M_{ij} = \begin{cases} 0, & w_i \text{ and } w_j \text{ on the same branch of the sentence tree} \\ -\infty, & w_i \text{ and } w_j \text{ on the different branch of the sentence tree} \end{cases}$$

where i and j are the result of encoding the Token of w_i and w_j by using the original method.

3.4 Whole Word Masking Transformer Model

After using soft position coding for short texts containing background knowledge and generating a visible matrix for them, we input them into the BERT model for short text classification. However, the structure of the benchmark BERT model cannot receive visible matrix as input directly. Meanwhile, the Token input for Chinese is on character level. If it is not processed well, the word breaking information of short text will be lost. To solve the above two problems, this paper proposes to develop the Transformer model in the original BERT model structure and obscures the whole word masking Transformer model (Whole Word Masking, WWM). The process of improving the Transformer model is divided into the following two steps:

1. Enhance the structure of the Transformer model in the BERT model. The self-attention mechanism in the structure of the Transformer model will be modified, and the visible matrix is added to its *Softmax* function to reduce the noise introduced by the invisible Token. Then, adding the visible matrix M will achieve the following equation:

$$\text{Attention}(Q', K', V') = \text{Softmax}\left(\frac{Q'K'^T + M}{\sqrt{d_k}}\right)V'$$

After the visible matrix M is added, where two Tokens belong to the same branch of the sentence tree, the result of attention calculation is not affected. On the other hand, the two Tokens correspond to take the value $-\infty$ in the visible matrix, which makes the value of attention obtained by the *Softmax* function approach to 0.

2. Improve the masking method in MLM models. We use the result of text segmentation where the Token of each character belonging to a word is masked so that the pre-trained model predicts every masked Token in the same word, that is, the whole word masking. Taking sequence (苹, 果, 类, 型, 公, 司, 首, 席, 执, 行, 官, 蒂, 姆, 库, 克, ……, 下, 调, 第, 一, 季, 度, 销, 售, 预, 期) as an example, in the feature extraction layer, the segmentation of the sequence has been obtained, and background knowledge can be united to further segmentation. If the character-level masking method of the original BERT model is used, the result of the masking sequence is:

([MASK] 果类型公司首席执行官蒂姆库克 下调第一季度销 [MASK] 预期)

Using the whole-word masking method, the masking sequence results are as follows:

([MASK] [MASK] 类型公司首席执行官蒂姆库克 下调第一季度 [MASK] [MASK] 预期)

Thus, by using whole word masking to learn word boundaries, we enable the basic BERT model to have a certain understanding of word granular context semantics.

3.5 Model Training

The short text classification model training BERT-KG algorithm is described as follows:

ALGORITHM NAME: BERT-KG

INPUT: Short text corpus training set $C = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ where x_i represents a short text and y_i is the classification category of short text x_i

OUTPUT: Short text classification model M

1. Preprocess input short text for word division, remove stop words, etc.
 2. $KW = TKE(x_i)$ // Using TKE algorithm to extract keywords from short text
 3. $E = CSEL(KW)$ // Using CSEL algorithm for keyword entity linking
 4. Expand short text feature short text sequence to $W = (w_1, (w_2, e_2), \dots, (w_i, e_i), \dots, w_N)$ where e_i is the target entity corresponding to the keyword w_i
 5. Extract the entity-relationship tuple information of the target entity, and generate a short text sentence tree as
 $s_{\text{tree}} = \{w_1, w_2\{(r_{21}, e_{21}), (r_{22}, e_{22}), (r_{23}, e_{23})\}, \dots, w_i\{(r_{i1}, e_{i1}), (r_{i2}, e_{i2}), \dots, (r_{im}, e_{im})\}, \dots, w_n\}$
 6. Tile out the sentence tree to get a text sequence of M characters
 7. Using Word2Vec to get short text character level Token embedded as $\text{Token}(x) = (t_1, t_2, \dots, t_n)$ where t_i is the embedding corresponding to the first character in the text sequence
 8. Soft-position coding of each character Token in the text sequence yields the result of encoding is $\text{SoftPosition}(x)$. Soft-position coding of each character Token in the text sequence yields the result of encoding is $\text{SoftPosition}(x) = (p_1, p_2, \dots, p_M)$ where p_i represents the i th Token, and generates a visible matrix based on the position encoding of the original short text and the sentence tree structure is $\text{VisibleMatrix}(x) = (m_{ij})$. If the i th Token and the j th Token are in the same branch of the sentence tree, $m_{ij} = 0$, otherwise $m_{ij} = -\infty$
 9. Embedding each Token in the extended text sequence into its own clause yields a segment embedding result of $\text{Segment}(x) = (sg_1, sg_2, \dots, sg_M)$, where sg_i represents the index of the i th clause.
 10. Adding Token embedding, soft location coding and segment embedding, the input representation of the model is $\text{InputRepresentation}(x) = \text{Token}(x) + \text{SoftPosition}(x) + \text{Segment}(x)$
 11. Add visible matrices to the self-attention mechanism in the Transformer model:

$$\text{Attention}(Q', K', V') = \text{Softmax}\left(\frac{Q'K'^T + M}{\sqrt{d_k}}\right)V'$$
 12. Replace character masking with whole word masking in the MLM pre-trained model of BERT, that is, according to the word segmentation results of the text, all tokens belonging to the same word are masked with [MASK]
 13. $\text{InputRepresentation}(x)$ is input into the improved BERT model to get a short text representation vector V which combines knowledge map with deep semantics.
 14. After fine-tuning the model, the short text feature representation is input into the Softmax classifier to get the probability distribution of the short text for each classification category, which is compared with the actual classification label, and the cross-entropy is used as the target function, that is $L = -\sum_{i=1}^n \sum_{j=1}^M y_{ij} \log p(\hat{y}_{ij})$ where y_{ij} is the probability distribution of the actual classification label, and \hat{y}_{ij} is the probability distribution of the classification labels of the model output.
 15. Adjust model parameters, minimize objective function L , get short text classification model M , output model M
-
- END

4 Experiments and Results Analysis

4.1 Experimental Setting

In this paper, we choose two datasets, namely DS1 and DS2, to evaluate our proposed model where external knowledge was derived from the CN-DBpedia knowledge map [16]. In detail, DS1 is the headline data set for Toutiao. This experiment uses 21,000 data for seven categories including technology (news_Tech), Finance (news_Finance), entertainment (news_Entertainment), international (news_World), automobiles (news_Car), culture (news_Culture) and sports (news_Sports). Test results are evaluated by 5-fold cross-validation. In addition, DS2 is a multi-topic categorized dataset of Weibo. This experiment uses 21,000 data from seven categories: (小米, 同桌的你, 房价, 恒大, 公务员, 韩剧, 贪官,) and also uses five-fold cross-validation to gain the test results.

4.2 Experimental Metrics

The results of short text classification are also evaluated by recall rate, accuracy rate, F1 score and Macro-F1 score. F1 score is used to comprehensively measure the recall rate and accuracy rate of two categories, while F1 macro average score is used to expand the evaluation index to multi category problems.

Recall (R): recall rate, indicates the proportion of samples that are correctly predicted to be positive to the samples that are actually positive.

$$R = \frac{TP}{TP + FN}$$

Precision (P): refers to the proportion of correctly predicted positive samples to all predicted positive samples.

$$P = \frac{TP}{TP + FP}$$

F1 score (F1): is a metric to comprehensively measure the recall and accuracy of text classification results.

$$F1 = \frac{2 \times R \times P}{R + P}$$

Macro-F1: is the average F1 score of each category, which is used to comprehensively evaluate the results of multi-classification problems.

4.3 Comparison Model Selection

In order to verify the effectiveness of the proposed BERT-KG model for short text classification task, this paper compares the short text classification results of Word2Vec+KG, benchmark BERT model, BERT-KG model based on character granularity masking and BERT-KG model based on whole word masking. Here we set up the following comparative test:

1. The feature expansion method based on knowledge graph and TextCNN classification model is used to classify short text. The target entity and its context information of short text keywords are obtained from knowledge graph to enrich the context features of short text, and then input into TextCNN model for short text classification.
2. Short text classification based on benchmark BERT model (BERT-BASE): Short text representation is performed without introducing external knowledge, and then the short text is classified.
3. Short text classification (BERT-KG(CM)) based on the BERT-KG model of character granularity masking: merges external knowledge into the benchmark Bert model to enrich the functions of short texts, and fine-tunes them according to the text classification task after obtaining the BERT-KG model to achieve short text classification.
4. Short text classification (BERT-KG(WWM)) based on the best-kg model of whole word masking: On the basis of BERT-KG, the whole word masking method is used to train the BERT model. According to the text classification task, the model is fine-tuned for short text classification.

The classification results of each model on datasets DS1 and DS2 are shown in Table 1 and Table 2, respectively.

Table 1. Short text classification results of DS1

Model	Index	Category							
		Tech	Finance	Entertainment	World	Car	Culture	Sports	Average
Word2Vec + KG	R	0.756	0.773	0.783	0.754	0.785	0.764	0.789	0.772
	P	0.775	0.761	0.772	0.736	0.764	0.749	0.772	0.761
	F1	0.766	0.767	0.778	0.745	0.774	0.756	0.781	0.767
BERT-BASE	R	0.772	0.798	0.810	0.773	0.814	0.787	0.819	0.796
	P	0.789	0.778	0.797	0.753	0.793	0.772	0.792	0.782
	F1	0.780	0.788	0.803	0.763	0.803	0.779	0.805	0.789
BERT-KG (CM)	R	0.788	0.810	0.814	0.779	0.822	0.802	0.824	0.806
	P	0.800	0.788	0.797	0.758	0.797	0.779	0.800	0.788
	F1	0.794	0.799	0.805	0.768	0.809	0.790	0.812	0.797
BERT-KG (WWM)	R	0.791	0.810	0.818	0.785	0.826	0.807	0.830	0.810
	P	0.804	0.791	0.800	0.759	0.797	0.784	0.799	0.791
	F1	0.798	0.801	0.809	0.772	0.811	0.795	0.814	0.800

According to Table 1, After extracting the deep semantics of short text based on BERT-KG model on DS1, we can observe that: **(O1)** Compared with the method which only uses the shallow semantics in Sect. 3, the F1 score of short text classification results is up to 3.3%, which proves that... **(O2)** Compared with the benchmark BERT model, the F1 score of the short text classification results of the BERT-KG model is the highest, which is increased by 1.1%, indicating that the incorporation of external knowledge to

expand the short text features is effective. **(O3)** By comparing the results of BERT-KG (CM) and BERT-KG (WWM) models on DS1, it can be concluded that the short text is a news headline, and the words are relatively strict and the word boundaries are relatively clear. The F1 score of classification results based on the BERT-KG (WWM) model is 0.3% higher than that of the BERT-KG (CM) model, which indicates that the BERT model trained by the method of whole word masking can learn more semantic features of short texts to a certain extent.

Table 2. Short text classification results of DS2

Model	Index	Category							
		Millet	My old classmate	House price	Evergrande	Civil servant	South Korean TV soaps	Corrupt officials	Average
Word2Vec + KG	R	0.797	0.747	0.789	0.771	0.798	0.775	0.789	0.779
	P	0.778	0.757	0.768	0.761	0.810	0.763	0.785	0.775
	F1	0.788	0.772	0.778	0.766	0.804	0.769	0.782	0.777
BERT-BASE	R	0.817	0.763	0.804	0.786	0.812	0.793	0.796	0.796
	P	0.794	0.776	0.784	0.773	0.826	0.780	0.806	0.791
	F1	0.805	0.770	0.794	0.779	0.819	0.786	0.801	0.793
BERT-KG (CM)	R	0.824	0.770	0.807	0.789	0.820	0.799	0.799	0.801
	P	0.796	0.782	0.788	0.780	0.829	0.785	0.811	0.796
	F1	0.810	0.776	0.798	0.785	0.824	0.792	0.805	0.799
BERT-KG (WWM)	R	0.828	0.773	0.812	0.794	0.822	0.806	0.805	0.806
	P	0.805	0.778	0.788	0.779	0.831	0.781	0.811	0.796
	F1	0.816	0.775	0.800	0.786	0.827	0.793	0.808	0.801

According to Table 2, after extracting the deep semantics of short text based on the BERT-KG model on DS2, we can observe that: **(O1)** Compared with the method which only uses the shallow semantics in Sect. 3, the F1 score of short text classification results is up to 2.4%, and the accuracy of each classification category is significantly improved. Combined with the characteristics of DS2, it can be concluded that the BERT-KG model can effectively extract the deep semantics of the short text and reduce the noise impact of the nonstandard part of the short text. **(O2)** Compared with the benchmark BERT model, the F1 score of the short text classification result of the BERT-KG model is the highest, which is increased by 0.8%, indicating that the introduction of external knowledge is effective to expand the features of the short text with non-standard. **(O3)** In DS2, short text with relatively random words and fuzzy word boundaries is a microblog published by users. We observe that the F1 score of classification results based on the BERT-KG (WWM) model is 0.2% higher than that of the BERT-KG (CM) model, which is slightly lower than that on data set DS1. However, it is also clear that the method of whole word masking helps to improve the model's ability to understand the semantics of short text.

5 Conclusions

In this paper, we propose a short text classification model, i.e., BERT-KG, which combines knowledge graph and deep semantics. In this model, sentence tree and soft position coding are used to embed the feature expansion term into the input representation of the Bert model. Then, the structure of the original BERT model is developed, and the visible matrix and full word masking are introduced to make the model contain a certain word granularity learning ability. In the end, the BERT model is fine-tuned to get the short text classification results. Through the experiments on two sets of datasets, we conduct the experiments on both our proposed BERT-KG and benchmarks, and experimental results show that our BERT-KG model can achieve the best short text classification results when tackling the issue of irregularities for short texts.

References

1. Zhao, H., Liu, H.: Classification algorithm of Chinese short texts based on wikipedia. *Libr. Inf. Serv.* **57**(11), 120–124 (2013)
2. Kilimci, Z.H., Omurca, S.I.: Extended feature spaces based classifier ensembles for sentiment analysis of short texts. *Inf. Technol. Control* **47**(3), 457–470 (2018)
3. Bollegala, D., Mastsuo, Y., Lshizuka, M.: Measuring semantic similarity between words using web search engines. In: Proceedings of the 2nd ACM International Conference on World Wide Web, pp. 757–766. ACM (2007)
4. Peters, M.E., Neumann, M., Iyyer, M., et al.: Deep contextualized word representations. In: Proceedings of NAACL-HLT, pp. 2227–2237 (2018)
5. Radford, A., Narasimhan, K., Salimans, T., et al.: Improving language understanding by generative pre-training[EB/OL] (2018). https://s3-us-west2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/languageunderstanding_paper.pdf
6. Cui, Y., Che, W., Liu, T., et al.: Pre-training with whole word masking for chinese bert (2019). <https://arxiv.org/abs/1906.08101>
7. Kim, Y.: Convolutional neural networks for sentence classification. In: EMNLP, pp. 1746–1751 (2014)
8. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Advances in Neural Information Processing Systems, pp. 649–657 (2015)
9. Lai, S., Xu, L., Liu, K., et al.: Recurrent convolutional neural networks for text classification. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
10. Lee, J.Y., Dernoncourt, F.: Sequential short-text classification with recurrent and convolutional neural networks (2016). <https://arxiv.org/abs/1603.03827>
11. Wu, W., Li, H., Wang, H., et al.: Probbase: a probabilistic taxonomy for text understanding. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, pp. 481–492 (2012)
12. Devlin, J., Chang, M.W., Lee, K., et al.: Bert: pre-training of deep bidirectional transformers for language understanding (2018). <https://arxiv.org/abs/1810.04805>
13. Zhou, Y., Jiaming, X., Cao, J., Bo, X., Li, C., Bo, X.: Hybrid attention networks for chinese short text classification. *Comput. Sist.* **21**(4), 759–769 (2018)
14. Amit, S.: Introducing the Knowledge Graph. Official Blog of Google, America (2012)
15. Bollacker, K., Cook, R., Tufts, P.: Freebase: a shared database of structured general human knowledge. In: Proceedings of the 22nd AAAI Conf on Artificial Intelligence, Menlo Park, CA, pp. 1962–1963. AAAI (2007)

16. Bizer, C., Lehmann, J., Kobilarov, G., et al.: DBpedia-a crystallization point for the web of data. *J. Web Semant.* **7**(3), 154–165 (2009)
17. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a large ontology from wikipedia and wordnet. *J. Web Semant.* **6**(3), 203–217 (2008)
18. Liu, W., Zhou, P., Zhao, Z., et al.: K-bert: enabling language representation with knowledge graph (2019). <https://arxiv.org/abs/1909.07606>



Uncertainty-Aware Self-paced Learning for Grammatical Error Correction

Kai Dang, Jiaying Xie, Jie Liu^(✉), and Shaowei Chen

College of Artificial Intelligence, Nankai University, Tianjin, China
 {dangkai,ying,shaoweichen}@mail.nankai.edu.cn, jliu@nankai.edu.cn

Abstract. Recently, pre-trained language models have gained dramatic progress on grammatical error correction (GEC) task by fine-tuning on a small amount of annotated data. However, the current approaches ignore two problems. On the one hand, the GEC datasets suffer from annotation errors which may impair the performance of the model. On the other hand, the correction difficulty varies across sentences and the generating difficulty of each token within a sentence is inconsistent as well. Therefore, hard and easy samples in GEC task should be treated differently. To address these issues, we propose an uncertainty-aware self-paced learning framework for GEC task. We leverage Bayesian deep learning to mine and filter noisy samples in the training set. Besides, we design a confidence-based self-paced learning strategy to dynamically adjust the loss weights of hard and easy samples. Specifically, we measure the confidence score of the model on the samples at the token-level and the sentence-level, and schedule the training procedure according to the confidence scores. Extensive experiments demonstrate that the proposed approach surpasses the baseline model by 2.0+ point of $F_{0.5}$ scores on several GEC datasets and proves the effectiveness of our approach.

Keywords: Grammatical error correction · Bayesian neural network · Self-paced learning

1 Introduction

Grammatical error correction (GEC) is the task of automatically detecting and correcting grammatical errors in sentences. Previous studies treated the GEC task as a translation task, where the erroneous sentence and the correct sentence corresponded to the source sentence and the target sentence, respectively [33]. Based on this concept, statistical machine translation (SMT) and neural machine translation (NMT) models have been utilized to implement the GEC task with remarkable results [5, 12]. Many researchers have concentrated attention on expanding the GEC dataset and proposed multiple data augmentation methods [16, 34]. In recent years, higher grades have emerged due to pre-trained language model [15]. By fine-tuning on a small number of annotated data, the pre-trained language model is capable of achieving excellent performance.

However, the current approaches suffer from two problems. Common GEC datasets contain inappropriately corrected sentence pairs, which will disrupt the model from learning GEC task. [22] investigated this problem in detail and demonstrated that removing noisy data can facilitate the performance of the model. Apart from that, existing methods neglect a characteristic of the GEC task, where the correction difficulty varies across sentences and the generating difficulty of each token within a sentence is inconsistent as well. Treating these samples equally ignores the diverse complexities of data and the learning status of the current model, which enhances the challenge of model learning. Therefore, we should assign different weights to easy and hard samples according to the current capability of the model.

To address the above issues, we develop an uncertainty-aware self-paced learning framework for GEC task. We first filter noisy samples from the dataset to reduce their impact on the model. With the help of a pre-trained model, we estimate the uncertainty of each sample using Bayesian deep learning so as to mine the noisy samples in the dataset. Furthermore, we design a confidence-based self-paced learning strategy to dynamically adjust the loss weights of hard and easy samples. Concretely, we measure the model's confidence scores on both token-level and sentence-level, whereby the losses are re-weighted in training. The model will simulate the human learning process where it learns simple sentences and easy parts of sentences first followed by complex sentences or difficult parts. To corroborate the effectiveness of our framework, we conducted extensive experiments based on the bart [18] model. The experimental results demonstrate that the proposed framework surpasses the baseline model by 2.0+ point of $F_{0.5}$ scores on several GEC datasets, proving that our framework can bring a boost to the pre-trained model. The contributions of our paper are as follows:

- We develop an uncertainty-aware self-paced learning framework for GEC task.
- We leverage Bayesian active learning to automatically mine and filter the noisy samples in the dataset.
- We design a confidence-based self-paced learning strategy to dynamically adjust the loss weights of hard and easy samples.
- Experiments demonstrate that our framework can enhance the behavior of pre-trained language models on GEC task.

2 Background

2.1 Grammatical Error Correction

Assume that $D = \{x_i, y_i\}$ is a GEC dataset consisting of $|D|$ error-corrected sentence pairs, in which x_i is the i -th source sentence and y_i is its corresponding corrected sentence. Each x_i is a sequence of m tokens: $x_i = \{x_1^i, x_2^i, \dots, x_m^i\}$, and each y_i is a sequence of n tokens: $y_i = \{y_1^i, y_2^i, \dots, y_n^i\}$. Suppose that the NMT framework is utilized as the GEC model [6], so the model need to produce the following conditional probabilities:

$$p(y_i|x_i; \theta) = \prod_{t=1}^n p(y_t^i|y_{<t}^i, x_i; \theta) \quad (1)$$

where θ is model parameters, and $y_{<t}^i = \{y_1^i, y_2^i, \dots, y_{t-1}^i\}$ represents the previous generated tokens. Usually, the model is optimized by using maximum likelihood estimation (MLE), which is equivalent to minimizing the negative log-likelihood (NLL) loss:

$$\mathcal{L}_i = -\frac{1}{n} \sum_{t=1}^n \log p(y_t^i | y_{<t}^i, x_i; \theta) \quad (2)$$

2.2 Bayesian Neural Network

Bayesian neural network [8] assumes that a prior probability distribution can be represented by a set of model weight parameters. For the case of the classification task, assume there is a likelihood model:

$$p(y = c|x, \theta) = \text{softmax}(f^\theta(x)) \quad (3)$$

where $f^\theta(x)$ is the model output with parameters θ .

Bayesian inference attempts to estimate the prior distribution $p(y = c|x, D)$ based on the true posterior distribution $p(\theta|D)$. However, this posterior distribution requires finding all possible model weights, which is infeasible in practice. To implement approximate variational inference, [8] proposed to approximate the posterior distribution $q_\omega(\theta)$ through Monte Carlo dropout (MC dropout), which performs random dropout before every weight layer during the training and test phase. MC dropout can minimize the Kullback-Leibler (KL) divergence between the approximating distribution and the true posterior $p(\theta|D)$:

$$\begin{aligned} p(y = c|x, D) &= \int_{\theta} p(y = c|x, \theta)p(\theta|D)d\theta \\ &\approx \int_{\theta} p(y = c|x, \theta)q_{\theta}(\theta)d\theta \\ &\approx \frac{1}{M} \sum_{m=1}^M p(y = c|x, \tilde{\theta}_m) \\ &= \frac{1}{M} \sum_{m=1}^M \text{softmax}(f^{\tilde{\theta}_m}(x)) \end{aligned} \quad (4)$$

where M denotes the sample times, $\tilde{\theta}_m \sim q_{\omega}(\theta)$, and $q_{\omega}(\theta)$ is the Dropout distribution [8, 26]. In other words, MC dropout is equivalent to performing M forward passes through the network and averaging the outputs of the softmax.

3 Methodology

In this paper, we employ a pre-trained sequence-to-sequence model (like BART) as the backbone, in this way to ensure that the model has the good ability to generate sentences and evaluate the uncertainty of sentences. Our framework

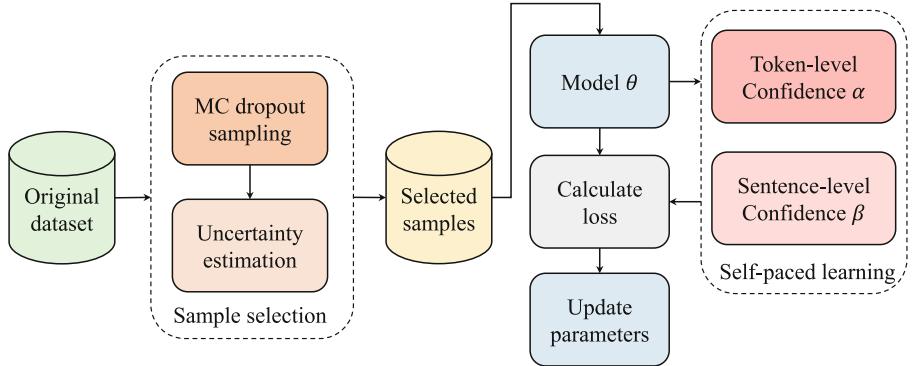


Fig. 1. Illustrations of uncertainty-aware self-paced learning framework.

is derived by extending the standard training procedure with two components, as shown in the Fig. 1, namely: (i) Sample selection: Inspired by deep Bayesian active learning, we estimate each sample’s uncertainty, and then remove the extremely confusing samples, which tend to be the noise data in the dataset, i.e., inappropriately corrected sentence pair. (ii) Confidence-based self-paced learning: We measure the confidence scores of the model on two granularities, namely the token level and the sentence level. Then we utilize the confidence scores to re-weight the loss during training, which adjusts the emphasis on hard and easy samples. In the following, we will explain these two components in detail.

3.1 Sample Selection

We adopt uncertainty-aware methods as our sample selection strategy. Specifically, the difficulty of the sample is measured by Bayesian Active Learning by Disagreement (BALD) [11]. BALD aims to select samples that maximize the information gain about the model parameters, i.e. maximize the mutual information between predictions and model posterior:

$$\mathbb{I}[y_i, \theta | x_i, D] = \mathbb{H}[y_i | x_i, D] - \mathbb{E}_{p(\theta|D)}[\mathbb{H}[y_i | x_i, \theta]] \quad (5)$$

where $\mathbb{H}[y_i | x_i, \theta]$ denotes the entropy of y_i given x_i and model weights θ . [9] approximated the above function with the approximate distribution $q_\omega(\theta)$ as follows:

$$\mathbb{I}[y_i, \theta | x_i, D] = - \sum_c \left(\frac{1}{M} \sum_m \hat{p}_c^m \right) \log \left(\frac{1}{M} \sum_m \hat{p}_c^m \right) + \frac{1}{M} \sum_{m,c} \hat{p}_c^m \log(\hat{p}_c^m) \quad (6)$$

where $\hat{p}_c^m = \text{softmax}(f^{\tilde{\theta}_m}(x))$. A high value of $\mathbb{I}[y_i, \theta | x_i, D]$ indicates the model is highly confused about the expected output of the instance x_i . We will mine noisy samples based on this uncertainty.

In the prior works [9], this acquisition function was used for mining valuable samples to speed up the training procedure. The purpose of this paper is

the opposite of that. Within the pre-trained model context, samples with large information gain are more likely to be noisy samples, so we need to remove these extremely confusing samples. We sort all samples by the value of $\mathbb{I}[y_i, \theta|x_i, D]$ from largest to smallest, and filter the top $p\%$ of samples. The remaining samples are considered clean samples and served as the training set.

3.2 Confidence-Based Self-paced Learning

Token-level confidence The difficulty of generating each token within the same sentence is different. Intuitively, the common tokens and copied tokens are easy to generate, while rare tokens and modified tokens are difficult to generate. We perform M Monte Carlo dropout sampling on the model to obtain M conditional probabilities. We utilize the variance of the probabilities to measure the prediction uncertainty of each token. In the early period of training, the model has a higher prediction probability mean and variance for easy tokens, but a lower prediction probability mean and variance for hard tokens. As the model capability grows, the model gets higher probability means and lower variances for easy tokens, while both probability means and variances increase for hard tokens. Thus, easy samples have higher uncertainty in the early stages and harder samples have higher uncertainty in the later stages. The uncertainty reflects the model's confidence in the prediction results. The token-level confidence scores are calculated as follows:

$$\hat{\alpha}_t^i = \text{Var}\{p(y_t^i|y_{<t}^i, x_i; \tilde{\theta}_m)\}_{m=1}^M \quad (7)$$

where $\hat{\alpha}_t^i$ denotes the confidence score of the token y_t^i , and $\text{Var}\{p(y_t^i|y_{<t}^i, x_i; \tilde{\theta}_m)\}$ denotes the corresponding probability variance. In order to smooth the training process and maintain the loss scale, we normalize the confidence scores by *softmax* function:

$$\alpha_t^i = \frac{\exp(\hat{\alpha}_t^i/\tau)}{\sum_{j=1}^n \exp(\hat{\alpha}_j^i/\tau)} \quad (8)$$

where τ indicates a temperature. A higher temperature will smooth the weight distribution, so we can adjust the extent of discrimination between hard and easy samples by varying the temperature.

Sentence-level confidence In addition to token-level confidence, we also focus on sentence-level confidence. As the sentence length, the number of rare tokens, and the number of grammatical errors increase, the difficulty of correcting the sentences increases as well. Similar to the token-level confidence, the confidence score $\hat{\beta}_i$ of the sample (x_i, y_i) is measured by the variance of the average predicted probability of the sentence:

$$\hat{\beta}_i = \text{Var}\{p(y_i|x_i; \tilde{\theta}_m)\}_{m=1}^M \quad (9)$$

where $\text{Var}\{p(y_i|x_i; \tilde{\theta}_m)\}$ indicates the variance of predicted probability with respect to y_i . We also normalize the confidence scores as:

$$\beta_i = \frac{\exp(\hat{\beta}_i/\tau)}{\sum_{j=1}^N \exp(\hat{\beta}_j/\tau)} \quad (10)$$

where N indicates the mini-batch size.

Training Strategy. To govern the learning schedule automatically, we leverage the confidence scores as factors to re-weight the loss during training. As described earlier, the model has a higher uncertainty for easy samples in the early stage, and the model focuses on easy samples. As training continues, the uncertainty of hard samples gradually increases and the model focuses more on hard samples. Thus, for each sentence y_i , the log-likelihood can be calculated as:

$$\mathcal{L}_i = \frac{1}{n} \sum_{t=1}^n \alpha_t^i \log p(y_t^i|y_{<t}^i, x_i; \theta) \quad (11)$$

We use the mini-batch approach for training, so the loss of a batch is:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \beta_i \mathcal{L}_i \quad (12)$$

4 Experiments

4.1 Datasets and Evaluation Metrics

Datasets. We use the BEA-2019 shared task [3] data as our training and development sets. Concretely, we train our models on NUCLE [7], FCE-train [32], Lang-8 [27], and W&I+LOCNESS [31] datasets, and we use W&I-dev as the development set. In the evaluation phase, we evaluate our model on CoNLL-2014 test set, FCE test set and BEA-2019 test set, respectively.

Metrics. For CoNLL-2014 test set and FCE test set, we report scores measured by the M^2 scorer. For BEA-2019 test set, we use the ERRANT scores for evaluation. All our results are average of four distinct trials using four different random seeds.

4.2 Experimental Setting

We conduct our experiments using the BART-large model implemented by fairseq¹ toolkit. Following [24], we apply two stages of fine-tuning to the model. We set $p = 10$, i.e., filter 10% of the samples. We set the number of MC dropout

¹ <https://github.com/pytorch/fairseq>.

Table 1. Comparison with existing models. A **bold** value indicates the highest score within the column. [†]Trained on additional data.

Models	CoNLL-2014			BEA-2019			FCE		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$
This work									
BART	70.8	44.4	63.3	69.4	59.3	67.1	69.4	40.7	60.8
+ sample selection	70.9	46.3	64.0	70.4	63.0	68.8	74.2	45.8	66.0
+ self-paced learning	72.6	46.6	65.3	72.3	60.7	69.6	76.1	48.4	68.2
Recent GEC Systems									
Mita et al. [22]	63.8	52.4	61.1	59.9	66.9	61.2	—	—	—
Lichtarge et al. [19]	69.4	43.9	62.1	67.6	62.5	66.5	—	—	—
Kiyono et al. [16]	67.9	44.1	61.3	65.5	59.4	64.2	—	—	—
Omelianchuk et al. [24] [†]	77.5	40.1	65.3	79.2	53.9	72.4	—	—	—
Kaneko et al. [14]	69.2	45.6	62.6	67.1	60.1	65.6	59.8	46.9	59.7

samples to 5 and the temperature τ to 0.1. The dropout ratio and label smoothing factor are both set to 0.1. We utilize the AdamW optimizer with a learning rate $5e - 5$ and a dynamic batch of 2,000 tokens. The warmup updates is 500 and the total update steps is 5,000. The accumulation steps is set to 4. During the inference phase, we use greedy decoding to correct the sentence. Note that we don't use any additional corpus and pre-processing and post-processing operations, such as ensembling models and re-ranking outputs.

4.3 Main Results

We compare our model with several previous state-of-the-art GEC systems. We choose the single-model GEC system of the same scale as the baselines for a fair comparison. [22] proposed a self-refinement strategy to denoise the dataset and reduce the impact of noisy data on the model. [19] incorporated delta-log-perplexity into a training schedule for GEC. [16] applied back translation to generate pseudo-data to enhance the model. [24] designed a GEC sequence tagger and produced promising results on the basis of pre-trained models. [14] integrated a pre-trained masked language model into an encoder-decoder model for GEC to improve the model's performance.

As shown in Table 1, our approach outperforms most existing GEC systems without the help of additional data and assistive technologies. Compared to simply fine-tuning BART, our approach is proven to produce a significant improvement to the model. For the BEA-2019 data, sample selection brings a 1.7 point improvement in $F_{0.5}$ score over the baseline. On this basis, self-paced learning delivers a boost of 0.8 point to the model. Sample selection reduces the effect of noisy samples and enhances the recall of the model. Self-paced learning gradually emphasizes the focus on hard samples, which brings an increase in precision for the model.

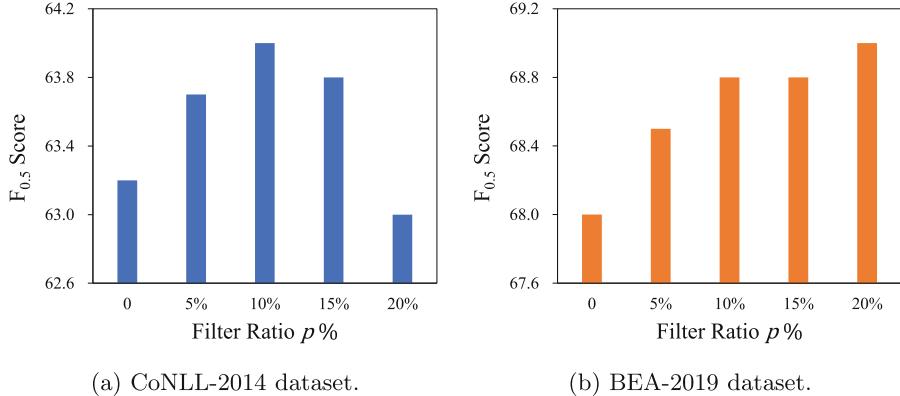


Fig. 2. Affects of filter ratio p on CoNLL-2014 and BEA-2019 test set.

4.4 Analysis of Filter Ratio p

In this section, we explore the effect of different filtering ratios p under the premise of label smoothing cross-entropy loss. We conduct experiments by varying only the value of p on CoNLL-2014 and BEA-2019 test sets. Figure 2 shows that the performance of the model is significantly improved on both the datasets when the filter ratio increases from 0 to 10%. As the filter ratio continues to rise, the model declines in performance on CoNLL-2014 and has a slight enhancement on W&I. It indicates that filtering a small number of noisy samples is beneficial for improving the model's performance because noisy data in the training set can disrupt the model from learning error correction. However, filtering too many samples prevents the model from taking full advantage of the annotated data, which can degrade the generalization ability of the model. So we need to find a trade-off when filtering the noisy samples. We choose 10% as the filtering ratio in the following experiments.

4.5 Analysis of Temperature τ

As aforementioned, temperature τ is utilized to adjust the extent of discrimination between hard and easy samples. Under the premise of filtering 10% of the data, we perform experiments to evaluate the effect of τ . We also compare our approach with label smoothing cross entropy-loss and focal loss [20]. As shown in Fig. 3, confidence learning generally outperforms label smoothing cross-entropy loss, indicating the necessity to treat hard and easy samples differently. The model achieves the best performance at τ of 0.1. This is because a larger τ smooths the weight distribution and discriminates less between hard and easy samples. GEC task entails greater differentiation of the samples so a smaller temperature should be chosen. We also conduct experiments with focal loss [20] under the same conditions. The experimental results show that focal loss degrades the performance of the model. The reason for this phenomenon is

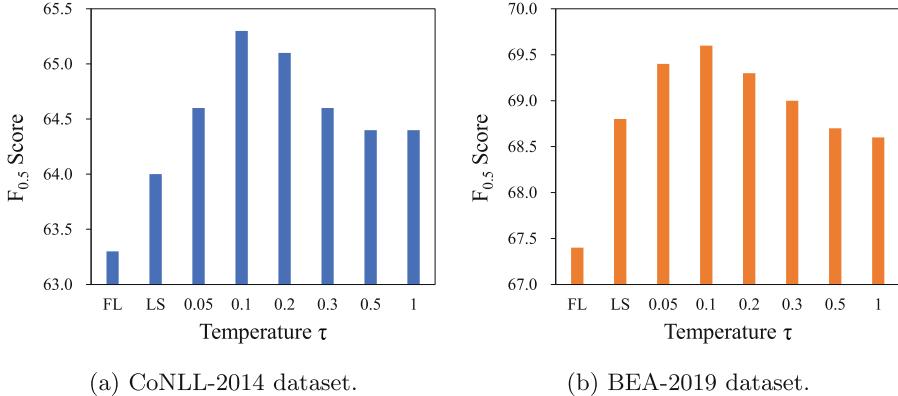


Fig. 3. Affects of temperature τ on CoNLL-2014 and BEA-2019 test set. LS denotes label smoothing cross-entropy loss. FL denotes focal loss.

that focal loss overemphasizes the importance of hard samples and causes the model to make errors on easy samples. This also suggests that it is ineffective for GEC task to increase the weights of hard samples merely. Dynamically adjusting the hard and easy sample weights is favorable for the model to learn GEC task.

4.6 Analysis of Self-paced Learning

To explore how self-paced learning adjusts the learning procedure, we visualized the token-level weight distribution in Fig. 4. We consider the first epoch as the early training period and the fourth epoch as the later training period. We observe that the model focuses more on common tokens or the copied tokens in the early period. As the training proceeds, the model gradually emphasizes rare tokens or corrected tokens. This phenomenon validates our idea that self-paced learning allows the model to learn generating sentences from easy to complex.

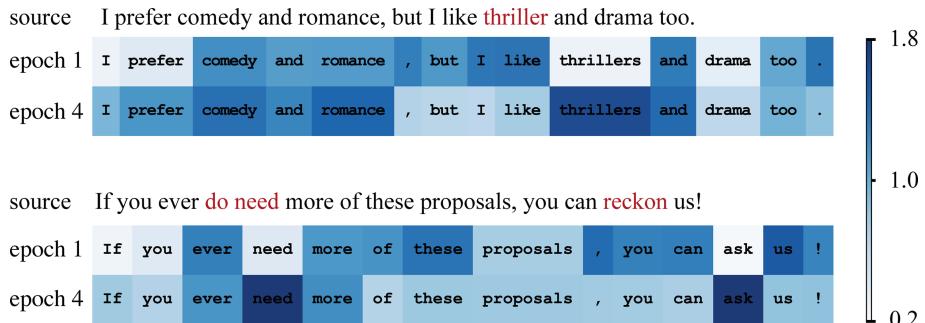


Fig. 4. Two visualization examples of token-level weight distributions.

5 Related Work

5.1 Grammatical Error Correction

Neural network-based models emerge as the dominant approach to grammar correction. Researchers treated the GEC task as a machine translation task and applied NMT models [5, 6, 13] to obtain remarkable results. Recently, benefiting from the development of pre-trained language models, several edit-based models have been proposed for GEC task. [1, 21, 24] designed different editing modes and gained promising results in terms of both accuracy and inference speed. Moreover, some researchers have focused on enlarging the training corpus and proposed various data augmentation methods, such as noise inject [34, 35], back translation [16], round-trip translation [10] and constructing adversarial examples [29]. In addition, there exist some explorations on GEC datasets. [22] investigated noisy data problem in GEC and proposed a self-refinement strategy to address it. [19] adopted delta-log-perplexity to reweight the GEC data.

5.2 Sample Selection

The philosophy of curriculum learning [2] is that neural networks learn the easier aspects of the task first followed by the more complex ones. It is a challenge to automatically distinguish between simple and hard samples. Prior researchers proposed self-paced learning [17] to select easy samples based on model confidence or lower loss during training. [25] leveraged hard sample mining (anti-curriculum learning) to deal with the hard and easy sample imbalance problem. [23] incorporated active learning with self-training to filter and utilize samples with high confidence. [30] applied meta-learning to the self-training framework to adaptively select and re-weight the samples.

5.3 Uncertainty-Aware Learning

DL research has mainly considered aleatoric uncertainty (AU) and epistemic uncertainty (EU). [8] proposed Bayesian neural networks to estimate the uncertainty of models. Based on the uncertainty, a large number of research works have emerged, such as sample selection [23], learning scheduling [28], mining difficult samples [4], etc.

6 Conclusion

In this work, we develop an uncertainty-aware self-paced learning framework for GEC task. We address the noisy data problem by automatic sample selection. Furthermore, we propose a confidence-based self-paced learning strategy to dynamically adjust the hard and easy sample loss weights. The experimental results illustrate that our approach can boost the pre-trained language model, obtaining a 2.0+ point gain on several datasets compared to the baseline model. In the future, our approach can be extended to semi-supervised learning in combination with data augmentation methods to improve GEC performance.

Acknowledgement. This research is supported by the National Natural Science Foundation of China under the grant No. 61976119 and the Natural Science Foundation of Tianjin under the grant No. 18ZXZNGX00310.

References

1. Awasthi, A., Sarawagi, S., Goyal, R., Ghosh, S., Piratla, V.: Parallel iterative edit models for local sequence transduction. *ACL* (2019)
2. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: *ICML* (2009)
3. Bryant, C., Felice, M., Andersen, Ø.E.: The BEA-2019 Shared Task on Grammatical Error Correction. *ACL* (2019)
4. Chang, H., Learned-Miller, E.G., McCallum, A.: Active bias: training more accurate neural networks by emphasizing high variance samples. In: *NIPS* (2017)
5. Chollampatt, S., Ng, H.T.: A multilayer convolutional encoder-decoder neural network for grammatical error correction, pp. 5755–5762. AAAI Press (2018)
6. Chollampatt, S., Taghipour, K., Ng, H.T.: Neural network translation models for grammatical error correction, pp. 2768–2774. IJCAI/AAAI Press (2016)
7. Dahlmeier, D., Ng, H.T., Wu, S.M.: Building a large annotated corpus of learner English: the NUS corpus of learner English. In: *BEA@NAACL-HLT* (2013)
8. Gal, Y., Ghahramani, Z.: Dropout as a Bayesian approximation: representing model uncertainty in deep learning (2016). [JMLR.org](https://jmlr.org)
9. Gal, Y., Islam, R., Ghahramani, Z.: Deep Bayesian active learning with image data, pp. 1183–1192. PMLR (2017)
10. Ge, T., Wei, F., Zhou, M.: Fluency boost learning and inference for neural grammatical error correction. Association for Computational Linguistics (2018)
11. Houlsby, N., Huszar, F., Ghahramani, Z., Lengyel, M.: Bayesian active learning for classification and preference learning. CoRR (2011)
12. Junczys-Dowmunt, M., Grundkiewicz, R.: Phrase-based machine translation is state-of-the-art for automatic grammatical error correction. *ACL* (2016)
13. Junczys-Dowmunt, M., Grundkiewicz, R., Guha, S., Heafield, K.: Approaching neural grammatical error correction as a low-resource machine translation task. *ACL* (2018)
14. Kaneko, M., Mita, M., Kiyono, S., Suzuki, J., Inui, K.: Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction, pp. 4248–4254. Association for Computational Linguistics (2020)
15. Katsumata, S., Komachi, M.: Stronger baselines for grammatical error correction using a pretrained encoder-decoder model. In: *AACL/IJCNLP 2020* (2020)
16. Kiyono, S., Suzuki, J., Mita, M., Mizumoto, T., Inui, K.: An empirical study of incorporating pseudo data into grammatical error correction. *ACL* (2019)
17. Kumar, M.P., Packer, B., Koller, D.: Self-paced learning for latent variable models. In: *NIPS 2010*, pp. 1189–1197 (2010)
18. Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, pp. 7871–7880 (2020)
19. Lichtarge, J., Alberti, C., Kumar, S.: Data weighted training strategies for grammatical error correction. *Trans. Assoc. Comput. Linguist.* **8**, 634–646 (2020)
20. Lin, T., Goyal, P., Girshick, R.B., He, K., Dollár, P.: Focal loss for dense object detection. In: *ICCV 2017*, pp. 2999–3007 (2017)
21. Malmi, E., Krause, S., Rothe, S., Mirylenka, D., Severyn, A.: Encode, tag, realize: high-precision text editing. In: *EMNLP-IJCNLP 2019*, pp. 5053–5064 (2019)

22. Mita, M., Kiyono, S., Kaneko, M., Suzuki, J., Inui, K.: A self-refinement strategy for noise reduction in grammatical error correction, pp. 267–280. ACL (2020)
23. Mukherjee, S., Awadallah, A.H.: Uncertainty-aware self-training for few-shot text classification. In: NeurIPS (2020)
24. Omelianchuk, K., Atrasevych, V., Chernodub, A.N., Skurzhanskyi, O.: Gector - grammatical error correction: Tag, not rewrite. In: BEA@ACL 2020. ACL (2020)
25. Shrivastava, A., Gupta, A., Girshick, R.B.: Training region-based object detectors with online hard example mining. In: CVPR (2016)
26. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting (2014)
27. Tajiri, T., Komachi, M., Matsumoto, Y.: Tense and aspect error correction for ESL learners using global context. The Association for Computer Linguistics (2012)
28. Wan, Y., et al.: Self-paced learning for neural machine translation. In: EMNLP 2020, pp. 1074–1080
29. Wang, L., Zheng, X.: Improving grammatical error correction models with purpose-built adversarial examples. In: EMNLP 2020, pp. 2858–2869 (2020)
30. Wang, Y., et al.: Adaptive self-training for few-shot neural sequence labeling. CoRR (2020)
31. Yannakoudakis, H., Andersen, Ø.E., Geranpayeh, A., Briscoe, T., Nicholls, D.: Developing an automated writing placement system for ESL learners. Appl. Measure. Educ. **31**(3), 251–267 (2018)
32. Yannakoudakis, H., Briscoe, T., Medlock, B.: A new dataset and method for automatically grading ESOL texts. ACL (2011)
33. Yuan, Z., Briscoe, T.: Grammatical error correction using neural machine translation. The Association for Computational Linguistics (2016)
34. Zhao, W., Wang, L., Shen, K., Jia, R., Liu, J.: Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data, pp. 156–165. NAACL (2019)
35. Zhao, Z., Wang, H.: MaskGEC: improving neural grammatical error correction via dynamic masking, pp. 1226–1233. AAAI Press (2020)



Metaphor Recognition and Analysis via Data Augmentation

Liang Yang^(✉), Jingjie Zeng, Shuqun Li, Zhexu Shen, Yansong Sun,
and Hongfei Lin

Dalian University of Technology, Dalian 116024, China

{liang,hflin}@dlut.edu.cn,

{jjwind,shuqunli,szx,bettersun}@mail.dlut.edu.cn

Abstract. Metaphoric expression is widespread and frequently used to convey emotions. When it comes to metaphor recognition and analysis, there are still not enough samples for these tasks. In this study, we target on recognizing verb metaphors and analyzing their emotions via data augmentation. To this end, we firstly propose a sentence reconstruction method to prune the dependency parsing tree, and thus alleviates the disturbances caused by the noise information. Then, the data augmentation strategies are proposed based on Seq2Seq model and the reconstructed sentence, which generate sufficient candidate samples after an effective quality evaluation. Finally, a proposed model is trained with the extended dataset, and it achieves the recognition and emotion analysis for metaphors. Experiments are conducted on Chinese and English metaphor corpus respectively, and results show that our proposed model has the best performance compared with the baseline methods.

Keywords: Metaphor recognition · Data augmentation · Sentence reconstruction · Dependency parsing tree

1 Introduction

In our daily life, individuals are inspired by the objects of real world. We frequently use metaphors to illustrate latent ideas, perceptions and emotions, which promotes the enrichment and evolution of language. In the book “The Metaphor We Live By”, [13] believe that metaphor is ubiquitous and inevitable, which is not only the polysemy of words but also an indispensable rhetorical device. According to some related statistics studies [1, 14], about one-third of sentences are generated in the form of metaphor expression, and this proves that metaphor is inseparable from human language and cognition. Hence, in many studies of linguistics, metaphor mostly adopts certain words to express another concept rather than taking their literal meaning. For instance, in the metaphor sentence: “你就是她感情的备胎” (You are a just-in-case), “just-in-case” emphasizes the unstable relationship, its metaphorical meaning is “the second substitute”, and the author expresses a negative emotion. However, since there is no

L. Yang and J. Zeng—Both authors contributed equally to this research.

emotion vocabulary occurred in this sentence, literal translation cannot predict the implicit emotion. Furthermore, some brain and cognitive activates more emotions in the human brain than the literal language in the same context [2]. Thus, the emotion analysis is an essential part of metaphor analysis. Take this sentence as a example: ““音乐凝固了这个小镇”” “(The town is frozen by the music)”, the verb “frozen” is used to embody the relationship between music and architecture. From above examples, we can find that metaphorical rhetoric acts as a bridge connecting source and target domains conceptually.

To investigate the underlying mechanisms in metaphors and analyze the implicated emotion automatically, many studies on sentiment analysis of metaphorical texts have been proposed. [19] constructs an emotion lexicon based model for this problem. Strzalkowski [18] proposes an approach, which recognizes the emotion associated with metaphorical language automatically.

Although existing research provides some foundation, there are still some urgent work to be carried out, such as work at the data level. The metaphor data set can be regarded as Low-Resource. Due to data annotation costs a lot, there is a lack of metaphorical data. Most of existing works tent to make the best use of limited semantic resources, the me which utilize the whole metaphor sentences as input units and make some progress. However, the interpretability of these work is not satisfied. These work lacks an analysis of the reason of metaphor, such as the role and function of different words in a special metaphor.

In fact, metaphorical information is only implied among a few special words in metaphors. The rest words might be unimportant, and they will introduce some noise, and affect the results of metaphor recognition and emotion classification tasks. As shown in Table 1, we list important and non important words in a metaphorical sentence. If the important words are obscured, it will completely change the original meaning of the sentence. But if the insignificant words are obscured, it will not affect the sentence understanding. On the contrary, this will make the metaphor sentence more concisely and easy to read. The ““亲尝”” (tasted) in the first example plays a key role in the metaphorical sentence, ““亲尝人生”” (tasting life) refers to carefully think about the life. In the second example, ““重筑”” (rebuilding) also plays a key role in this metaphorical sentence, it means giving the second chance in one’s life.

According to the phenomena above, we try to explore a kind of an effective method to reconstruct the metaphor and get rid of noise, which will make the metaphor sentences more focused on the core words, and also help metaphor understanding, such as judging whether a sentence is a metaphor or not. Furthermore, based on the core meaning of these metaphors, we introduce a data augmentation approach to expand the metaphor dataset, which will provide more sufficient and workable training data, leading to better effect of model training.

The contributions of this paper are summed as the following three-folds:

- We propose a sentence reconstruction method for obtaining important information of metaphors, and it will reduce the influence of some noises in metaphorical sentences.

Table 1. The effect of masking different words.

Metaphor Examples:
人生需要亲历亲尝才算是真正活过。 Life needs to be experienced, tasted and lived for real.
它在重筑我人生过程中起到了重要的作用。 It plays an important role in rebuilding my life.
Masking Important Word:
人生需要亲历<mask>才算是真正活过。 Life needs to be experienced, <mask> and lived for real. 它在<mask>我人生过程中起到了重要的作用。 It plays an important role in <mask> my life.
Masking Unimportant Words:
人生需要亲历亲尝才算是<mask>活过。 Life needs to be experienced, tasted and lived <mask>. 它在重筑我人生过程中起到了<mask>作用。 It played a <mask> role in rebuilding my life.

- Data augmentation strategies based on the sentence reconstruction has been proposed, which utilize the core information of metaphors to generate sufficient candidate samples.
- We conduct detailed experiments on both Chinese and English metaphor corpus respectively, and the results show that our model achieves the best performance in metaphor recognition and emotion analysis.

2 Related Work

2.1 Metaphor Recognition

Recently, there are many different types of NLP models have been used for metaphor recognition task, and they provide important references for this task. Shutova [16] tends to map the concepts between source and target domains, and this theory is originally from conceptual metaphor theory [13]. Also, Many models primarily rely on contextual information to predict whether a targeted phrase is metaphoric [5] or not.

Verb metaphor recognition is an important task of metaphor recognition. Hongyan [9] uses conditional random fields model and maximum entropy model to recognize verb metaphor and points out that there is no mature syntactic and semantic tool for metaphor analysis in Chinese. Klebanov [11] investigates the effectiveness of semantic generalizations and classifications for capturing the regularities of the verbs' behavior, and tries to mining their metaphoricity from orthographic word unigrams. For noun metaphor recognition, Fu [6] develops hierarchical clustering for Chinese noun phrases in order to recognize metaphorical phrases. However, due to the lack of semantic information, their models can only cover a small part of Chinese nouns.

Although these works have achieved some results, they did not analyzed the role of different words in metaphorical sentences, especially the core words. Based on the above analysis, the accuracy of metaphor recognition results will be improved to a large extent by distinguishing them properly.

2.2 Data Augmentation

Data augmentation has been proved to be an effective way to expand dataset and improve the results. Unfortunately, in metaphor recognition and emotion analysis tasks, there is no related research that specifically adopts data augmentation strategy to improve their results. Previous works usually have adopted Generative Adversarial Networks [7] to directly generate augmented data.

In these years, Cubuk et al. [3] proposes automatic data augmentation, which uses a hypernetwork to train the target model. There are also some template-based operations, like Wei et al. [20], and they presents a four text operations to augment data. Since random replacement used in template-based method might replace the core words in metaphors, in this paper, we introduce the model based on Seq2seq to achieve data augmentation.

3 Data Augmentation Based on Sentence Reconstruction

In this section, we aim to provide a data augmentation method based on sentence reconstruction (DASR), which is used to solve the problem of metaphor recognition and metaphor emotion analysis tasks.

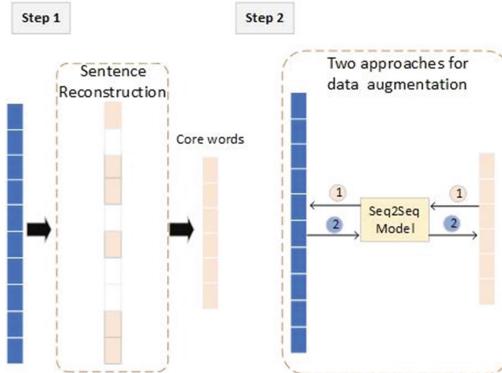


Fig. 1. The framework of the entire approach. Step 1 represents metaphors with sentence reconstruction. Step 2 shows two methods of data augmentation (S2L, L2S).

To achieve this goal, our method is divided into two steps. First, we extract the core meaning in a metaphorical sentence through dependency parsing tree. And then, we train a Seq2Seq model for data augmentation. The ends of this

Seq2Seq model are the original metaphorical sentence and the reconstructed sentence. The reconstructed sentence is used as the input or output of Seq2Seq model, which will generate different data augmentation methods. Specific information can be seen in Subsect. 3.2. In the end, we use the generated data to assist us in metaphor recognition and metaphor emotion classification. The specific frame diagram is shown in Fig. 1.

3.1 Sentence Reconstruction

Based on the observation of metaphors, we find that metaphors are only implied between a few specific words, and these words are the core words of a metaphor. While some unimportant words might introduce noise for metaphor recognition or emotion classification tasks. Hence, we decide to reconstruct the sentence based on the relationship of dependency parsing tree. Dependent parsing tree can correctly reflect the relationship between words in sentences.

We use the StanfordNLP developed by the Stanford University Natural Language Processing Group to generate a dependency parsing tree [15]. We assume that the “ROOT” of a sentence is the most important sentence in a metaphorical sentence. Where “ROOT” is defined that The root grammatical relation points to the root of the sentence. The example of the dependency parsing tree is shown in Fig. 2. The metaphor sentence: *I dream tomorrow, and this country will stand up*. In this tree, its root, *dream*, is indeed the core of this sample. We selectively retain and delete words according to the universal dependency relations provided by Stanford University [4], which gives 37 universal syntactic relations. We only keep the following groups of relationships as follows, for these relations can keep the SVO structure of the metaphor sentence to the a large extent:

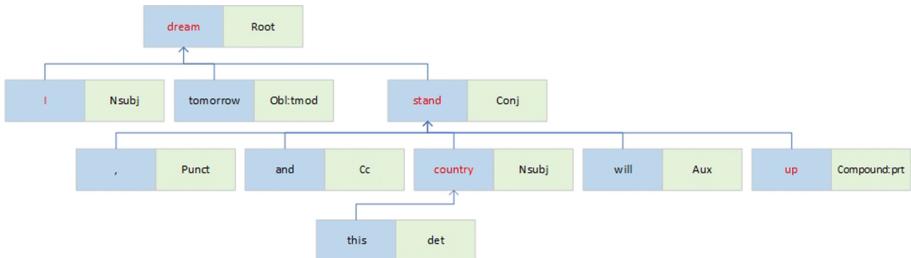


Fig. 2. Dependency parsing tree generated by StanfordNLP. Where “dream” is the tree “ROOT”, and it means the root of the grammatical sentence. In this tree, each word has a “relationship” in addition to its own meaning, and it represents its relationship with other words in the dependency parsing tree. The red font represents the core meaning of this metaphor (Color figure online)

root: the root grammatical relation points to the root of the sentence. **nsubj:** a nominal subject is a nominal which is the syntactic subject and the proto-agent of a clause. **obj:** the object of a verb is the second most core argument of a verb after the subject. **csubj:** a clausal syntactic subject of a clause. **ccomp:** the main

component of the complement clause. **xcomp**: an open clausal complement of a verb or an adjective is a predicative or clausal complement without its own subject. **compound:prt**: a particle verbs. (Provided that the verb to which it belongs is *root* or *conj*.) **conj**: a conjunct is a relation between two elements that are coordinated.

The red part in Fig. 2 indicates dependency parsing tree after reconstruction with the specific example. The new generated sentence is: *I dream country stand up*. Note that these relationships may not exist together in a sentence, but as long as the relationship appears, it will be retained to generate a new sentence. Obviously, the word retained already contains the metaphorical information of the original sentence, and the unimportant information is removed. Furthermore, it is more concise, easier to be analyzed, and the removed words can be replaced with any other related word. Based on this sentence reconstruction mechanism, we reconstruct all the metaphor sentences for the next step, and apply them as the input for the data augmentation.

3.2 Data Augmentation

The sentence reconstruction above can be seen as condensing the core content of metaphorical sentences through linguistic rules. Through the core content of the obtained metaphor, we consider the following two ways to expand the data. Both methods are based on the Seq2Seq model, but their input and output are different. We define two approaches as: Long2Short(L2S), Short2Long(S2L). The example is presented in Table 2.

Table 2. Sentences reconstructed by different approaches. The red part is the core word in the metaphorical sentence.

Metaphor Sentence:
我喜欢你的眸子，仿佛宝石一样。 I like your jewel-like eyes.
Long2Short:
我难受的像凋谢的笑容。 I feel like a withered smile.
Short2Long:
我沐浴着和煦的春风，似静水一般，向我命途中妙龄美的阳光一样，我被雨熏染得黑里透红的情，要被你洒在妙龄。 I am bathed in the warm spring breeze, like calm water, and like the sunshine of my youth in the way I am caught in the rain and darkened by the rain.

Generating Short Sentences from Long Sentences. We use the original metaphor sentence as input (source sentence) for the encode end, and the short sentence reconstructed based on linguistic rules as output (target sentence) at the decode end. We can find that from the perspective of sentence structure, although great changes have taken place in semantics, there are still metaphorical elements implied in the generated sentence shown on the row Long2Short of Table 2. We can think that the generated short sentence imitates the previous

noun metaphorical sentence pattern and generates a similar noun metaphorical sentence pattern.

Generating Long Sentences from Short Sentences. Contrary to the aforementioned strategy, we also use short sentences reconstructed based on the linguistic rules as the input of the encode end (source sentence), with the original metaphor sentence as the output of the decode end (target sentence) to conduct sentence reconstruction. As shown in the row Short2Long of Table 2, following this strategy, it still maintains the metaphorical elements with the great semantic and structure changes.

Based on these two methods, we can improve the results of metaphor recognition or metaphor emotion analysis with limited metaphor resources.

4 Experiments

4.1 Datasets

In order to explore the generality of metaphors between different languages, we conduct experiments on Chinese metaphor dataset (CMRSA) and English metaphor (VUA) dataset. Relevant statistics can be seen in Table 3.

Table 3. VUA and CMRSA Datasets information

Datasets	Data		Classes
	Train set	Test set	
CMRSA-Task 1	4394	1100	3
VUA	7529	2694	2

Table 4. Emotion analysis dataset information on CMRSA-Task 2

	Joy	Love	Angry	Sadness	Fear	Disgust	Surprise	Total
Train	362	1662	124	524	171	705	82	3630
Test	111	407	34	135	41	158	23	909

CMRSA. Chinese Metaphor Recognition and Sentiment Analysis(CMRSA) is the largest publicly available metaphor dataset, which consist of two tasks – metaphor recognition and emotion analysis [21] for metaphors. The corpus is used by the CCL-2018¹ Metaphor Task. The Chinese verb metaphor recognition task aims to recognize whether a sentence is a verb metaphor, a noun metaphor or a non-metaphor sentence. While the emotion analysis of Chinese metaphor is to predict the emotion categories of a metaphor in terms of 7 classes (joy, love, angry, sadness, fear, disgust, and surprise), the details can be seen in Table 4. We

¹ <http://www.cips-cl.org/static/CCL2018/call-evaluation.html>.

test our methods on both metaphor recognition and metaphor emotion analysis tasks.

VUA. VU Amsterdam Metaphor Corpus² (VUA) is the authoritative data set of the English metaphor [17]. Every word in the corpus is labeled, guided by MIP. Metaphor recognition has been treated as a sequence tagging task by the NAACL-2018 Metaphor Shared Task. We apply a classification method to achieve verb tracks. After de-duplicating the data set, as long as the verb has a metaphor label in this sentence, we think that this sentence belongs to verb metaphor, and then this dataset can be used in verb metaphor recognition task for comparison.

4.2 Baselines

We have selected several strong baselines to verify that our method achieve good results in both Chinese and English metaphor datasets. All the following-mentioned data augmentation methods are expanded to the training set, and BERT is used for classification.

- **Character Augment.** Augmenting data in character level. For English data, we directly replace characters randomly according to the keyboard distance. For Chinese data, we used two methods for character-level enhancement: 1. Randomly disassemble the Chinese characters into radicals 2. Randomly replace the pinyin of the Chinese characters according to the keyboard distance, and then spell out the Chinese characters. 30% of word will be augmented.
- **Word Embeddings Augment.** Besides character augmentation, word level is important as well. We make use of word embeddings to find most similar group of words to replace original word, like word2vec. For English data, we used Google-News word vectors. For the Chinese data, we used the Tencent-AILab-ChineseEmbedding word vectors. 30% of word will be augmented.
- **Contextual Word Embeddings Augment.** To benefit from the development of transformer models, we can use language models to predict possible target word, like BERT. We use of bert-base-chinese pretrain model for Chinese dataset, bert-base-uncased pretrain model for English dataset. 30% of word will be augmented.
- **Sentence Augment.** Using the creativity of the model, we hope to continue writing based on metaphorical sentences, which can be seen as an explanation or description of metaphorical sentences, such as the GPT model.
- **RNN-MHCA** [8]. This is a result of the current state of the art on the word-level level of the VUA metaphor dataset. We use this method as a powerful baseline for two sentence-level metaphor tasks.

4.3 Experimental Results and Analysis

Metaphor Recognition. In order to verify the effectiveness of our method, we use F1-score as the main measurement for method performance. We test our

² <http://ota.ahds.ac.uk/headers/2541.xml>.

Table 5. The performance on two metaphor analysis tasks.

	Metaphor recognition		Metaphor emotion analysis
	VUA	CMRSA-Task 1	CMRAS-Task 2
Bert baseline	78.11	85.09	46.83
Character augment	Replace characters radicals	77.90 -	84.88 85.15
	Word embeddings augment	78.69	85.77
Contextual word embeddings augment	79.13	86.02	47.15
Sentence augment	78.91	86.23	47.33
RNN-MHCA	79.30	85.51	46.45
DASR (Bert sentence pair)	80.19	86.32	47.27
DASR (Supplement data)	80.65	86.44	47.72

method on the English metaphor dataset VUA and Chinese metaphor dataset CMRSA in sentence-level metaphor recognition. The results is shown in Table 5. All data augmentation processes the training set once by default. For the model performances of VUA corpus results are worse than CMRSA corpus, it is because the source of VUA corpus coming from some more formal written text, while most of the data in CMRSA corpus comes from the Internet, they are different from sources and languages. Also, the CMRSA dataset obtained from Internet is more informal, they are easy to be reconstructed.

Our data augmentation based on sentence reconstruction (DASR) model start from analyzing the dependency grammar tree, and also consider the grammatical information and hierarchical relationships of metaphorical sentences to some extent. It shows the effectiveness of hierarchical analysis.

After data augmentation, more data is used for training, thus the result is improved. According to the results, supplementing the generated pseudo-data as training data can improve the metaphor recognition tasks in Chinese and English to a certain extent, and achieve the best results compared with the baseline.

We used two methods to use this data. The first way is to directly expand the training set, so that the model can get more training samples, which has always improved the accuracy of emotion classification. The second one is to generate the pseudo-data as a sentence pair with the original sentence, and it lets the model find the corresponding semantic meaning from these two sentences. For example, is it a verb metaphor or a noun metaphor. In general, noun metaphors use one object to refer to another, and the generated sentences also have the same structure.

The method of using sentence pairs is worse than the method of expanding the training set. It might be because when the input is sentence pairs, and Bert is looking for the relationship between sentence pairs. Some core words of metaphorical sentences are not particularly obvious, and the generated pseudo-data is also not with a high quality, so Bert cannot accurately find the similar structure between the two sentences. Then the result will be worse than the method of expanding the data set.

Metaphor Emotion Analysis. Metaphor is a typical non-literal expression, often expressing emotions through implicit and indirect language. In fact, emotion often appears in the form of metaphor [12]. The core words of metaphorical sentences can often express certain emotions. Also, metaphor recognition and metaphor emotion analysis are inseparable, thus we also use our proposed data augmentation method to verify the effect on metaphor emotion analysis task. The metaphor emotion analysis result is also shown in Table 5. It is a 7-classes task, our method still achieves the best results on this emotion analysis dataset. This means that our data augmentation method can also capture the emotions of metaphorical sentences in emotion analysis tasks, and generate corresponding pseudo-data for these emotions.

In the CMRAS emotion analysis task, there is less train data on emotion categories, such as angry and surprise. Through the data augmentation method, the models can effectively learn how to recognize the emotion, so the accuracy of the results is improved to a certain extent.

Sentence Reconstruction with Dependency Parsing Tree. We also explore the core part of metaphor from the dependency parsing tree. We believe that “Root” in the dependency parsing tree is very important for verb metaphors. The subsequent data augmentation is also based on obtained words from the dependency parsing tree. In order to find out whether which ones are core words in metaphor, we have made the following hypothetical experiments on VUA and CMRSA dataset. We mask “Root” in metaphor sentences, or other unimportant word (word don’t reserved in Subsect. 3.1). The analysis results are shown as in Table 6. To verify this, The experiment is conducted with the CNNs text classification [10] and randomly initialize 300-dimensional word vectors.

Table 6. The results of masking different words on metaphorical classification task.

Datasets	Original sentence	Root-masked sentence	Unimportant-masked sentence
VUA	60.23	58.23	64.92
CMRSA-Task 1	75.57	69.82	76.18

Regardless of the Chinese and English dataset, if the core word (ROOT) of a sentence is masked, it will affect the results of metaphor recognition. On the contrary, if the unimportant words in the sentence is masked, it will improve the accuracy of the metaphor recognition to a certain degree. As for why the effects of masks on Chinese and English data sets are different, as far as we know, most of the Chinese data sets are verb metaphors in this CMRSA, and ROOT words are mostly verbs. We learn that the ROOT node of the dependency parsing tree plays a vital role in metaphorical sentences. In contrast, dependency parsing trees have some relationships that are not much important. It might become the noise part of the sentence. This implies that we should retain core information and remove the noise information for sentence reconstruction. Through these

core meanings, for data augmentation part, we can obtain similar metaphorical sentences to assist metaphor recognition or metaphor emotion analysis tasks.

Two Approaches for Data Augmentation. As mentioned in Sect. 3.2, we propose two methods to enhance the data: Generating Short sentences from Long sentences (S2L), Generating Long sentences from Short sentences (L2S). Both of these data augmentation methods start from the core words of metaphor. The methods are used to generate pseudo data as a supplementary training set. The results on two data sets are shown in Table 7. Although the pseudo sentences generated using the S2L method introduce some additional information. But the generated patterns are all metaphorical and generate more pseudo-data for training model, so the classification results of the model are improved.

Table 7. The performance on two tasks using different data augmentation methods.

Datasets	S2L	L2S
VUA	80.65	79.52
CMRSA-Task 1	86.44	85.95
CMRSA-Task 2	47.72	47.04

S2L method can generate more pseudo-data than L2S method. Because in S2L mode, input is the core word of metaphor, and output is the complete metaphor sentence. When generating pseudo-data, the input becomes a metaphorical sentence, so that the output becomes richer and contains more content. In L2S mode, the input is a complete metaphor sentence, while the output is the metaphor core word during training, thus metaphorical sentences used in such training way cannot be reused model, so the classification results of the model are improved.

5 Conclusion and Future Work

Metaphor, as a frequently used expression in our daily life, has become a crystallization of human language and culture. Based on the characteristics of metaphor, we propose an effective method to prune the dependency parsing tree, and it will focus on the more important of core words for metaphor data augmentation. By using data augmentation with sentence reconstruction to obtain the new generated data, the results of metaphor recognition and metaphor emotion analysis tasks are improved, and the model achieves the best performance on both Chinese and English metaphor dataset.

In the future, we will explore other sentence reconstruction methods to better analyze the core information of the metaphor sentences. Constraints can be added during training. It is required to have a certain degree of similarity between the generated text and the original sample. Meanwhile, how to generate more high-quality metaphorical sentences is also our important goal.

References

1. Cameron, L.: Metaphor in Educational Discourse. A&C Black, London (2003)
2. Citron, F.M., Goldberg, A.E.: Metaphorical sentences are more emotionally engaging than their literal counterparts. *J. Cogn. Neurosci.* **26**(11), 2585–2595 (2014)
3. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: learning augmentation policies from data (2019)
4. De Marneffe, M.C., Dozat, T.: Universal Stanford dependencies: cross-linguistic typology (2014)
5. Dunn, J.: Evaluating the premises and results of four metaphor identification systems. In: Gelbukh, A. (ed.) CICLing 2013. LNCS, vol. 7816, pp. 471–486. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37247-6_38
6. Fu, J., Wang, S., Wang, Y., Cao, C.: A practical method of identifying Chinese metaphor phrases from corpus. In: Lehner, F., Fteimi, N. (eds.) KSEM 2016. LNCS (LNAI), vol. 9983, pp. 43–54. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47650-6_4
7. Goodfellow, I., Pouget-Abadie, J., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
8. Guerin, F., Mao, R.: End-to-end sequential metaphor identification inspired by linguistic theories. In: 57th Annual Meeting of the Association for Computational Linguistics (ACL) (2019)
9. Hongyan, Z., Weiguang, Q., Fen, Z., Junsheng, Z.: Chinese verb metaphor recognition based on machine learning and semantic knowledge. *Eng. Technol.* **11**(3), 59–64 (2011)
10. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746–1751 (2014)
11. Klebanov, B.B., Leong, C.W., Gutierrez, E.D., Shutova, E., Flor, M.: Semantic classifications for detection of verb metaphors (2016)
12. Kovecses, Z.: Metaphor: A Practical Introduction. Oxford University Press, New York (2010)
13. Lakoff, G., Johnson, M.: Metaphors We Live By. U of Chicago P, Chicago (2003)
14. Martin, J.H.: A corpus-based analysis of context effects on metaphor comprehension. *Trends Linguist. Stud. Monographs* **171**, 214 (2006)
15. Qi, P., Dozat, T., Zhang, Y., Manning, C.D.: Universal dependency parsing from scratch. arXiv preprint [arXiv:1901.10457](https://arxiv.org/abs/1901.10457) (2019)
16. Shutova, E.: Design and evaluation of metaphor processing systems. *Comput. Linguist.* **41**(4), 579–623 (2015)
17. Steen, G.: A Method for Linguistic Metaphor Identification: From MIP to MIPVU, vol. 14. John Benjamins Publishing (2010)
18. Strzalkowski, T., Shaikh, S., Cho, K., et al.: Computing affect in metaphors. In: ACL 2014, vol. 42 (2014)
19. Veale, T.: A context-sensitive, multi-faceted model of lexico-conceptual affect. In: 50th Annual Meeting of the Association for Computational Linguistics, p. 75 (2012)
20. Wei, J.W., Zou, K.: Eda: easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint [arXiv:1901.11196](https://arxiv.org/abs/1901.11196) (2019)
21. Zhang, D., Lin, H., Yang, L., Zhang, S., Xu, B.: Construction of a Chinese corpus for the analysis of the emotionality of metaphorical expressions. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, vol. 2, Short Papers, pp. 144–150 (2018)



Exploring Generalization Ability of Pretrained Language Models on Arithmetic and Logical Reasoning

Cunxiang Wang^{1,2}, Boyuan Zheng⁴, Yuchen Niu⁵, and Yue Zhang^{2,3(✉)}

¹ Zhejiang University, Hangzhou, China

² School of Engineering, Westlake University, Hangzhou, China

{wangcunxiang,zhangyue}@westlake.edu.cn

³ Institute of Advanced Technology, Westlake Institute for Advanced Study, Hangzhou, China

⁴ Johns Hopkins University, Baltimore, USA

zhengboyuan@westlake.edu.cn

⁵ Imperial College London, London, UK

niuyuchen@westlake.edu.cn

Abstract. To quantitatively and intuitively explore the generalization ability of pre-trained language models (PLMs), we have designed several tasks of arithmetic and logical reasoning. We both analyse how well PLMs generalize when the test data is in the same distribution as the train data and when it is different, for the latter analysis, we have also designed a cross-distribution test set other than the in-distribution test set. We conduct experiments on one of the most advanced and publicly released generative PLM - BART. Our research finds that the PLMs can easily generalize when the distribution is the same, however, it is still difficult for them to generalize out of the distribution.

Keywords: Pretrained language model · Generalization · Mathematical reasoning

1 Introduction

Neural networks have shown strong capabilities in a range of NLP tasks [18, 19]. Recently, pretrained language models (PLMs) have achieved significantly levels of performance gains on many benchmark datasets [3, 8, 15]. Recently, some work shows that neural networks are lack of generalization ability in mathematical and logical reasoning [11, 13]. This can lead to more understanding of the limitation of existing models and motivate future work. However, no work has been done to quantitatively or intuitively explore the conditions under which PLMs can generalize, in terms of whether PLMs can understand the internal mathematical rules and logical rules. The example of mathematical rules is shown in Fig. 1. We suppose that if the model can effectively learn the underlying rules of Addition and Subtraction when giving sufficient training data, it can generalize to all two-number addition and subtraction calculation.

$$\begin{array}{r}
 256 \\
 +347 \\
 \hline
 603
 \end{array}
 \quad
 \begin{array}{r}
 347 \\
 -256 \\
 \hline
 91
 \end{array}$$

Addition Subtraction

Fig. 1. Example mathematical rules for Addition and Subtraction. If the model can master these rules, we suppose it can generalize well on all two-number addition and subtraction samples.

To this end, we conduct quantitative insights by designing a series of tasks for simple mathematical operations and logical reasoning, which includes numbering, addition, subtraction, comparison, and symbolic logic. We construct a set of corresponding datasets, where instances are in the form of text or mathematical expressions. Some examples are shown in the next section. For example, in the Addition task, ‘100 + 200’ is the question and ‘300’ is the answer.

There are various types of generalization [7, 10], such as question generalization on distribution differences between training set and test set [20], and answer generalization on distribution differences between training set and test set [13]. For example, in the Addition task, if the question and answer numbers in training data are of three-digit, but the question and answer numbers in the testing data are of two- or four-digit, they are in different distribution. To cover each type of generalization, we use different kinds of tasks and corresponding dataset. For example, we use *addition* to test the generalization on the question distribution differences data between training and testing. In this task, the numbers in the training set and development have three digits. However, the numbers in test set is set to consist of two, three, and four digits.

We conduct experiments using BART [8] since they can generate arbitrary text sequences and have been shown to achieve the state-of-art results on numerous Natural Language Processing (NLP) tasks. For each task, we fine-tune BART with training data, validate on the development set and finally evaluate on the test set. We find that strong PLMs can address simple generalization of the same answer distribution for counting, arithmetic and logic tasks. But they cannot master the underlying rules of arithmetic reasoning, for example, the model trained on 3-digit addition can handle the addition expressions with 2-digit or 4-digit.

We will release all the code and data set for future study.

2 Task

We construct five tasks related to algebraic and logical reasoning, namely **Numbering**, **Addition**, **Subtraction**, **Comparison**, **Symbolic Logic**. In order to test the generalization ability of models on the data with the same distribution and on the data with the different distribution, we create an in-distribution

Table 1. Data statistics of each task. For each task, we list the in-distribution dataset and cross-distribution test set.

Task	Train set	Dev set	In- + Cross-distribution test set	In- + Cross-distribution dataset
Numbering - Counting	3,744	468	468 + 2,610	4,680 + 2,610
Numbering - Listing	3,744	468	468 + 2,610	4,680 + 2,610
Addition	256,320	32,040	32,040 + 4,000	320,400 + 4,000
Subtraction	256,320	32,040	32,040 + 4,000	320,400 + 4,000
Comparison	648,000	81,000	81,000 + 6,100	810,000 + 6,100
Symbolic Logic	40,000	5,000	5,000 + 2,200	50,000 + 2,200

Task: Numbering

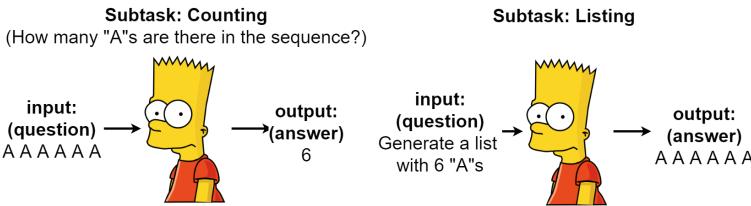


Fig. 2. The Numbering task has two subtasks, namely Counting and Listing.

dataset and a cross-distribution dataset for each task. The in-distribution dataset contains train set, development set and test set that are in the same distribution. The cross-distribution dataset only serves as the test set and it is in the different distribution in contrast to the in-distribution dataset. We believe that if the model can understand the underlying rules of arithmetic and logical Reasoning, it can both generalize well on in-distribution and cross-distribution test set.

Numbering

This task comprises two symmetric subtasks, namely **Counting** and **Listing**. Examples are shown in Fig. 2. The Counting task asks the model to count the number of characters in the input sequence. For example, ‘A A A A A A’ is a sequence with length ‘6’. The Listing task asks the model to output a list with a specific length and character. For example, the model receives a command ‘Generate a list of 6 A’ and the result is ‘A A A A A A’.

Addition

The Addition task is the standard summation of two input numbers. In order to make sure that all numbers are in the same distribution during training, we use only the equations whose left-hand-side and right-hand-side are both *three digits* in the in-distribution dataset. We also adopt two-digit and four-digit numbers on both sides in cross-distribution test set to further test the generalization ability of models. One example is shown in Fig. 3a.

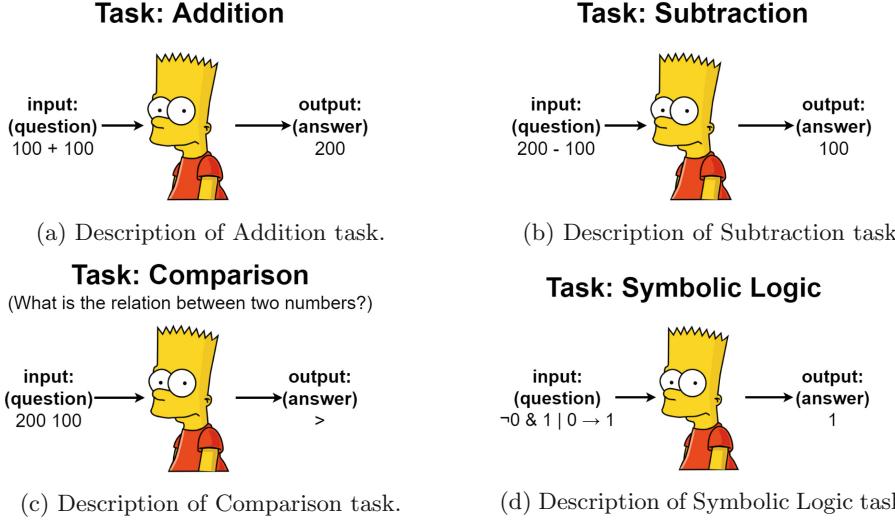


Fig. 3. The descriptions of four kinds of subtasks.

Subtraction

The Subtraction task is the standard task to subtract a subtrahend from a minuend. In order to make sure that all numbers are in same distribution during training, we use only equations whose left-hand-side and right-hand-side are both *three digits* in the in-distribution dataset. We also adopt two-digit and four-digit numbers on both sides in cross-distribution test set to further test the generalization ability of models. A example of Subtraction task is shown in Fig. 3b.

Comparison

The Comparison task is to determine which of the two numbers is greater or smaller. In order to make sure all numbers are in same distribution during training, we use only equations whose left-hand-side and right-hand-side are both *three digits* in the in-distribution dataset. We also adopt two-digit and four-digit numbers on both sides in cross-distribution test set to further test the generalization ability of models. One example is shown in Fig. 3c.

Symbolic Logic

As shown in Fig. 3d, this task is to reason over symbolic logic expressions. The input question expression consists of six basic components, which are ‘0’, ‘1’, ‘&’, ‘|’, ‘¬’ and ‘→’, representing *FALSE*, *TRUE*, *AND*, *OR*, *NOT* and *IMPLY*, respectively. The output answer is either 0 or 1, which represent *FALSE* and *TRUE*, respectively. This task asks the model to reason over the input logic expression and determine whether it is true or false.

In order to make sure all expressions are in the same distribution during training, we use only the expressions that contain **6–10** basic ‘0’ and ‘1’ components. For testing the generalization ability of models, we also adopt the some expressions with **1–15** basic ‘0’ and ‘1’ components in the test set.

Different from the other tasks, we select a subset from the overall dataset to serve as the in-distribution dataset because the data is large. We take only 10,000 of expressions with X basic components, where X is a number between 6–10, respectively. So, we end up with 50,000 samples in the in-distribution dataset.

Metrics. We use Exact Match to compute accuracy for Numbering, Addition, Subtraction and Comparison tasks. However, for the Symbolic Logic task, since the answer distribution is unbalanced (84% answers are ‘1’), we use the F1 score as the metric.

3 Experiments

In this section, we separate the generalization experiments to **In-Distribution Generalization** experiments and **Cross-Distribution** experiments. In the former, the testing data is in the same distribution with the training data. In the latter, the testing data is in the different distribution from the training data. We suppose that if the model can master the underlying rules of the mathematical and logical reasoning, it should achieve 100% accuracy on both In-Distribution Generalization experiments and Cross-Distribution experiments.

We have organized the details of in-distribution data and cross-distribution data in this section. In addition, We also sorted out the examples of them and put the examples in the Appendix Table 1¹.

3.1 Experimental Settings

We adopt BART [8] namely due to the following reasons. First, it is a generative pretrained language model, which means that they can generate arbitrary sequences of tokens. This is essential for the addition and subtraction task. Second, it has achieved state-of-art results on numerous tasks and they has received much research attention. Last, it has released model checkpoints, thus it can be more standardized and more fair can evaluate them.

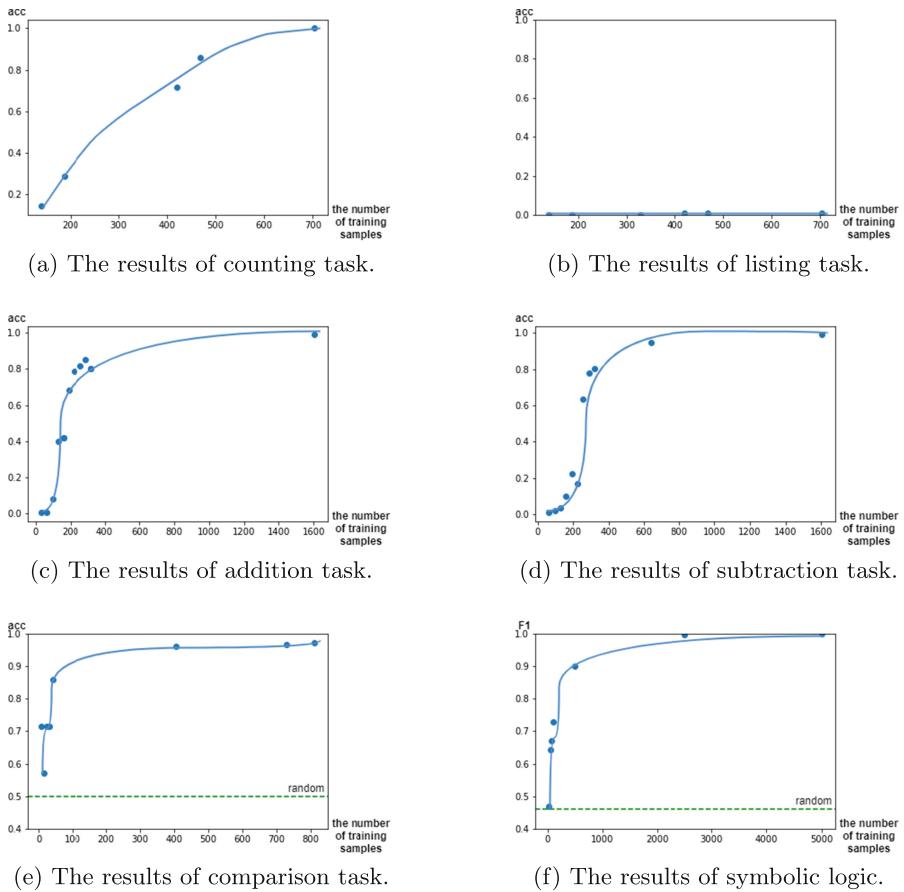
For the BART [8] model, we conduct experiments on the publicly released ‘BART-Large’ checkpoint². We insert spaces between numbers while representing them in the data. For example, ‘111’ is written as ‘1 1 1’ both in the question and answer. For the character sequence in the Numbering task, we also insert spaces between the sequence, such as ‘A A A’.

3.2 In-Distribution Generalization

In this subsection, we mainly explore models’ generalization ability on test data which in the same distribution with train data. For the Counting subtask of the Numbering task, each question is a sequence with 10–99 same character which

¹ We also present two extra interesting analysis in the Appendix.

² <https://huggingface.co/facebook/bart-large/tree/main>.

**Fig. 4.** The in-distribution results on each task.

is one character among the alphabet; each answer is an integer between 10 and 99. For the Listing task, each question is a textual sequence ‘Generate a list with X Y ’, where X is an integer between 10 and 99 and Y is one character among the alphabet; each answer is a sequence with 10–99 same characters. For the Addition task, each question is an addition expression, and the answer is a sum number. Each number in the question and answer is three digits. For the Subtraction task, each question is an subtraction expression , and the answer is a difference number. Each number in the question and answer is of three-digit. For the Comparison task, each question is made of two numbers and each answer is a single symbol which is either ‘ $>$ ’ or ‘ $<$ ’ or ‘ $=$ ’. The numbers in the question are all of three-digit. For the Symbolic Logic task, each question is a sequence with **5–10** basic ‘0’ and ‘1’ components; each answer is either 0 or 1.

For testing generalization ability on the same distributional data, we explore how the number of training samples affects the generalization. For each task, we extract subsets from the in-distribution train set and train on the subsets, but keep the distribution of development set and test set the same. Thus, we analyse how the number of training samples influences the performance, which also indicate the generalization ability of models on the data with the same distribution.

The in-distribution results on the Numbering task are shown in Fig. 4b Fig. 4a. For the Listing subtask, we find that the model’s generation results are very unstable, which means that the outputs often contain other tokens other than the needed character. For example, when the input is ‘Generate a list of 6 A’, the output can be ‘A A a Aa E T A’. When the sequence length increases, this kind of disruption will be more likely to occur. So, results are always around zero. We suppose this result is result from the instability of the generative model itself, because we also observe this situation from other generative models, such as T5 [16]. So, we mainly analyse the Counting task rather the Listing task in the following sections.

It can be seen that when the number of training samples increase, the performance of Counting will also improve.

The in-distribution results on the **Addition** task are shown in Fig. 4c. We can see that when the number of training samples is 1600 (0.5% of the dataset), the model can achieve 99% accuracy; even when the number of training samples is reduced to 160 (0.05% of the dataset), the model can still achieve around 40% accuracy. The in-distribution results on the **Subtraction**, **Comparison**, **Symbolic Logic** task are shown in Fig. 4d, Fig. 4e and Fig. 4f, respectively. It can be seen from the figures that when the number of training samples increase, the model can perform better in the in-distribution test set. And when the training samples increase to several hundreds, the model can achieve around 100% accuracy or F1, showing BART’s ability on the in-distribution generalization. Thus, we are wondering whether the model has truly learn the underlying rules of these tasks or they just use some spurious correlations to solve these questions, so, we design cross-distribution generalization test set to further explore the model’s generalization ability in the following section.

3.3 Cross-Distribution Generalization

In this section, we analyse how models generalize (1) when test question distribution is different from train question while the test answer distribution is the same; (2) when test answer distribution is different from the train answer while the test question distribution keeps the same; (3) when the test question distribution and test answer distribution are both different from train set. We have designed testing data for different types of cross-distribution on each task and list examples of the testing data in this section.

3.3.1 Varying Questions

In this part, we mainly talk about when the test question distribution is different from the train question while the test answer distribution keeps the same, how strong is the model's generalization ability. So, we use the Counting, Addition, Subtraction, Comparison, and Symbolic Logic tasks to analyse. For the Counting task, we use the instances whose character is not in letters of an alphabet while the number is still of two-digit. For example, the question is '@ @ @ @ @ @ @ @' and the answer is '10'. For the Addition task, we use the instances whose at least one added number is of two-digit. But we make sure answers of selected equations are all of three-digit. For example, the question is '50 + 170', the answer is '220'. For the Subtraction task, the situation is similar to the Addition task, we use the instances whose at least one number is of four-digit. But we make sure answers of selected instances are all of three-digit. For example, the question is '1000 - 500', the answer is '500'. For the Comparison task, the situation is also similar, we use the instances whose at least one number is of two-digit or four-digit. For example, the question is '56 176', the answer is '<'. For the Symbolic Logic task, the situation is also similar, we use the instances which has 1–5 or 11–15 basic '0' and '1' components. For example, the question is '*not 0 and 1 or 0*', the answer is '1'.

3.3.2 Varying Answers

In this part, we mainly talk about when the test answer distribution is different from the train answer while the test question distribution keeps the same, how strong is the model's generalization ability. As a result, we use the Addition and Subtraction to analyse.

For the Addition task, we use the instances whose two numbers are of three-digit while the answer is of four-digit. For example, the question is '500 + 600', the answer is '1100'. For the Subtraction task, the situation is similar to the Addition task, we use the instances whose two numbers are of three-digit while the answer is of two-digit. For example, the question is '550 - 500', the answer is '50'.

3.3.3 Varying Instances

In this part, we mainly talk about when the test question distribution and test answer distribution are both different from the train set, how strong is the model's generalization ability. So, we use the Counting, Addition and Subtraction tasks to analyse.

For the Counting task, we use the instances whose character is not in letters of an alphabet and number is not of two-digit. For example, the question is '@ @ @ @ @ @ @ @' and the answer is '9'. For the Addition task, we use the instances whose at least one number in question is of two- or four-digit and the answer number is also of two- or four-digit. For example, the question is '50 + 960', the answer is '1010'. For the Subtraction task, the situation is similar to the Addition task, we use the instances whose at least one number is of two- or four-digit and the answer is also of two- or four-digit. For example, the question is '1100 - 50', the answer is '1050'.

Table 2. The performance of BART on cross-distribution test set. For each task and different distribution type, we select the model checkpoint which has achieved 100% accuracy/F1 on the corresponding in-distribution test set. Note that the random result on Comparison is around 49.9%.

Task	Question cross-distribution	Answer cross-distribution	Instance cross-distribution
Counting	379/380 (99.7%)	/	12/1259 (0.95%)
Addition	15/1,500 (1.0%)	0/1,500	0/1,000
Subtraction	13/1,500 (0.87%)	0/1,500	1/1,000 (0.1%)
Comparison	2,555/5,600 (45.63%)	/	/
Symbolic Logic	2,200/2,200 (100%)	/	/

3.3.4 Analysis on Different Cross-Distributions

The model’s performance on the test set of different types of cross-distributions is shown in Table 2. From the table, we can see that although BART has achieved 100% accuracy on the in-distribution testing data, it fails to generalize on the cross-distribution testing data of arithmetic reasoning tasks.

Results of Counting and Symbolic Logic task on cross-distribution testing data are quite high. However, for Counting task, all correct instances are the instances which have different length but have the same character distributions with the training data. In addition, the cross-distribution testing data only have length difference from the training data. Thus, we can conclude that the model is not sensitive to the length of question if the basic components does not change. This conclusion is also consistent with the result of [2]. In addition, the results show that the model is especially weak in generalizing to the instances with different answer distributions.

To conclude, the model is still struggling on cross-distribution generalization, especially the carrying and borrowing in Addition and Subtraction tasks.

4 Related Work

Some works have investigated in Mathematical problems in NLP [4, 22, 25]. DROP [4] is a reading comprehension dataset comprising several kinds of mathematical tasks, such as Subtraction and Selection. However, all answers of its questions can be directly or indirectly found in the corresponding passages. Math23L [22] is simple math word problem dataset with 23k problems. Its problem is of the simple English context format, along with the equation and the answer. Ape210K [25] is a Chinese simple math word problem dataset with 210k questions. The questions are similar to Math23L’s questions. The data are taken from some elementary school math word problems. These datasets do not contain a generalization test set, the test set is in the same distribution with the train set. In addition, the often used methods for these datasets are first to predict

the equations or expression for the question and then to use calculation tool to get the result [22, 23]. However, our work concentrate on the generalization ability of models. Thus, we have designed test set with different distribution. In addition, we try to use the model to directly solve the questions, aiming test model's internal ability of understanding the deep rules of arithmetic and logical reasoning.

Some works have researched on models' the internal ability of solving mathematical expressions. [20] has investigated that how will different types of embedding, such as BERT [3] and GloVe [14], affect the performance of the same NAQANet model [4] on the same tasks including List Maximum, Decoding and Addition. Besides, this work [20] also explore that how the way numbers are represented and the way to do tokenization affect the performance of models. [5] try to inject numerical reasoning skill by adding a calculation module into the PLMs, which helps the performance on DROP [4] dataset.

There are also some works research focusing on the generalization ability of neural network models. [7] research on the compositional generalization skills of sequence-to-sequence models, such as LSTM [6] and GRU [1]. [10] explain that the generalization test in machine learning (ML) is not very reasonable, they put forward seven suggestions to better evaluate the generalization ability of ML models. [9] and [21] find that the PLMs cannot generalize well on Closed-book QA task [17], the model can handle the test instances which overlap with the train data, however, they cannot solve the non-overlapped instances. [12] find that even when the model's architecture is set, the generalization ability of the model is still influenced largely by the random luck, the random initialized weights and other things. [2] perform Transformer-based models on simple logic reasoning test, and their results show that the model can get quite promising results and the model is not sensitive to the question length. [24] analyses the generalization ability on the relation extraction task and find some specific problems can induce a significant decline in model performance.

5 Conclusion

We have designed a series of tasks for evaluating BART on simple mathematical operations and logic reasoning, which includes numbering, addition, subtraction, comparison, and symbolic logic. We constructed a corresponding in-distribution datasets, and also designed cross-distribution test set to further evaluate the model's generalization ability. If the model can understand the underlying rules of these mathematical operations and logic reasoning, it can generalize well on both in-distribution and cross-distribution test set. Our experiments showed that BART can only generalize on the in-distribution test set but cannot perform well on the cross-distribution test set, showing that the most advanced PLM still cannot understand the underlying rules of simple mathematical operations and logic reasoning.

References

1. Chung, J., Gülcöhre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. [arXiv:abs/1412.3555](https://arxiv.org/abs/1412.3555) (2014)
2. Clark, P., Tafjord, O., Richardson, K.: Transformers as soft reasoners over language. In: Bessière, C. (ed.) Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020, pp. 3882–3890. International Joint Conferences on Artificial Intelligence Organization (2020). <https://doi.org/10.24963/ijcai.2020/537>, main track
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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 (Long and Short Papers), vol. 1, pp. 4171–4186. Association for Computational Linguistics, Minneapolis, June 2019. <https://doi.org/10.18653/v1/N19-1423>, <https://www.aclweb.org/anthology/N19-1423>
4. Dua, D., Wang, Y., Dasigi, P., Stanovsky, G., Singh, S., Gardner, M.: DROP: a reading comprehension benchmark requiring discrete reasoning over paragraphs. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Long and Short Papers), vol. 1, pp. 2368–2378. Association for Computational Linguistics, Minneapolis, June 2019. <https://doi.org/10.18653/v1/N19-1246>, <https://www.aclweb.org/anthology/N19-1246>
5. Geva, M., Gupta, A., Berant, J.: Injecting numerical reasoning skills into language models. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 946–958. Association for Computational Linguistics, July 2020. <https://doi.org/10.18653/v1/2020.acl-main.89>, <https://www.aclweb.org/anthology/2020.acl-main.89>
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
7. Lake, B., Baroni, M.: Generalization without systematicity: on the compositional skills of sequence-to-sequence recurrent networks. In: ICML (2018)
8. Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7871–7880. Association for Computational Linguistics, July 2020. <https://doi.org/10.18653/v1/2020.acl-main.703>, <https://www.aclweb.org/anthology/2020.acl-main.703>
9. Lewis, P., Stenetorp, P., Riedel, S.: Question and answer test-train overlap in open-domain question answering datasets (2020)
10. Linzen, T.: How can we accelerate progress towards human-like linguistic generalization? In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 5210–5217. Association for Computational Linguistics, July 2020. <https://doi.org/10.18653/v1/2020.acl-main.465>, <https://www.aclweb.org/anthology/2020.acl-main.465>
11. Madsen, A., Johansen, A.R.: Measuring arithmetic extrapolation performance. [arXiv:abs/1910.01888](https://arxiv.org/abs/1910.01888) (2019)
12. McCoy, R.T., Min, J., Linzen, T.: BERTs of a feather do not generalize together: Large variability in generalization across models with similar test set performance. In: Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, pp. 217–227. Association for Computational

- Linguistics, November 2020. <https://doi.org/10.18653/v1/2020.blackboxnlp-1.21>, <https://www.aclweb.org/anthology/2020.blackboxnlp-1.21>
13. Nogueira, R., Jiang, Z., Li, J.: Investigating the limitations of the transformers with simple arithmetic tasks. [arXiv:abs/2102.13019](https://arxiv.org/abs/2102.13019) (2021)
 14. Pennington, J., Socher, R., Manning, C.: GloVe: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543. Association for Computational Linguistics, Doha, October 2014. <https://doi.org/10.3115/v1/D14-1162>, <https://www.aclweb.org/anthology/D14-1162>
 15. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8), 9 (2019)
 16. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**(140), 1–67 (2020). <http://jmlr.org/papers/v21/20-074.html>
 17. Roberts, A., Raffel, C., Shazeer, N.: How much knowledge can you pack into the parameters of a language model? In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 5418–5426. Association for Computational Linguistics, November 2020. <https://doi.org/10.18653/v1/2020.emnlp-main.437>, <https://www.aclweb.org/anthology/2020.emnlp-main.437>
 18. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. arXiv preprint [arXiv:1409.3215](https://arxiv.org/abs/1409.3215) (2014)
 19. Vaswani, A., et al.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS 2017, pp. 6000–6010. Curran Associates Inc., Red Hook (2017)
 20. Wallace, E., Wang, Y., Li, S., Singh, S., Gardner, M.: Do NLP models know numbers? probing numeracy in embeddings. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 5307–5315. Association for Computational Linguistics, Hong Kong, November 2019. <https://doi.org/10.18653/v1/D19-1534>, <https://www.aclweb.org/anthology/D19-1534>
 21. Wang, C., Liu, P., Zhang, Y.: Can generative pre-trained language models serve as knowledge bases for closed-book qa? (2021)
 22. Wang, Y., Liu, X., Shi, S.: Deep neural solver for math word problems. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 845–854. Association for Computational Linguistics, Copenhagen, September 2017. <https://doi.org/10.18653/v1/D17-1088>, <https://www.aclweb.org/anthology/D17-1088>
 23. Wangperawong, A.: Attending to mathematical language with transformers. [arXiv:abs/1812.02825](https://arxiv.org/abs/1812.02825) (2018)
 24. Zhang, N., et al.: Can fine-tuning pre-trained models lead to perfect nlp? a study of the generalizability of relation extraction. [arXiv:abs/2009.06206](https://arxiv.org/abs/2009.06206) (2020)
 25. Zhao, W., Shang, M., Liu, Y., Wang, L., Liu, J.: Ape210k: a large-scale and template-rich dataset of math word problems. CoRR abs/2009.11506 (2020). <https://arxiv.org/abs/2009.11506>

Multimodality and Explainability



Skeleton-Based Sign Language Recognition with Attention-Enhanced Graph Convolutional Networks

Wuyan Liang¹ and Xiaolong Xu²

¹ Nanjing University of Posts and Telecommunications, Nanjing 210023, Jiangsu, China

² School of Computer and Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, Jiangsu, China
xuxl@njupt.edu.cn

Abstract. The natural language processing of sign language is an important task in the field of artificial intelligence and information processing. In this paper, we propose an attention-enhanced graph convolutional networks (AEGCNs) for sign language recognition (SLR). First, there are four kinds of adaptive graphs for graph convolution and each graph topology can be either uniformly or individually learned based on the skeleton data in an end-to-end manner. In addition, we employ the spatial-temporal-channel attention mechanisms to give higher weight to the relative important joints, frames and features, and the higher-order connection with Chebychev polynomial approximation to enlarge the receptive field of graph convolution. Meanwhile, the information of both the joints and bones is simultaneously modeled in a framework, which further improves the representation of the movement about hand and finger. Finally, experiments on the DEVISIGN-D, DSL50 and ASL20 datasets show that the accuracies for top1 of three datasets reach 82.96%, 95.09% and 90.23% respectively and the accuracies for top5 of three datasets achieve 96.07%, 99.18% and 100% respectively. Compared with ST-GCN and BHOF, the accuracy of AEGCNs obtains significant improvements of +33.5% and +5.32% on ASL20 datasets, respectively, which demonstrates the effectiveness of our method on SLR.

Keywords: Skeleton-based sign language recognition · GCNs · Adaptive graph · Higher-order connection · Attention mechanism

1 Introduction

Sign language [1] is the primary language for the deaf and hard of hearing. Currently, with the development of information technology, the focus on the information processing of spoken language and written language, is gradually to depth computing. However, the sign language information processing is seriously lagging behind and remains at the starting stage. The essential difference between sign language processing and traditional language processing is spatial modeling. Sign language recognition (SLR) aims to automatically transcribe sign language to text or speech by the computer. In this paper,

we address the problem of modeling the dynamic spatial-temporal correlations of the skeleton-based sign language.

Recently, the skeleton-based SLR has achieved promising performance. Some skeleton-based SLR approaches [3, 4] convert the sequence of skeletons as a sequence of vectors, to capture the joints' relationship in both temporal and spatial domains. And some other skeleton-based methods [5, 6] convert the sequence of skeletons into pseudo-images by reorganizing the joints' coordinates into a 2D map and employ the CNN model to extract temporal and spatial features. As these approaches convert the skeleton into a sequence or a regular grid, it cannot utilize the complex, irregular and non-Euclidean the skeleton structure.

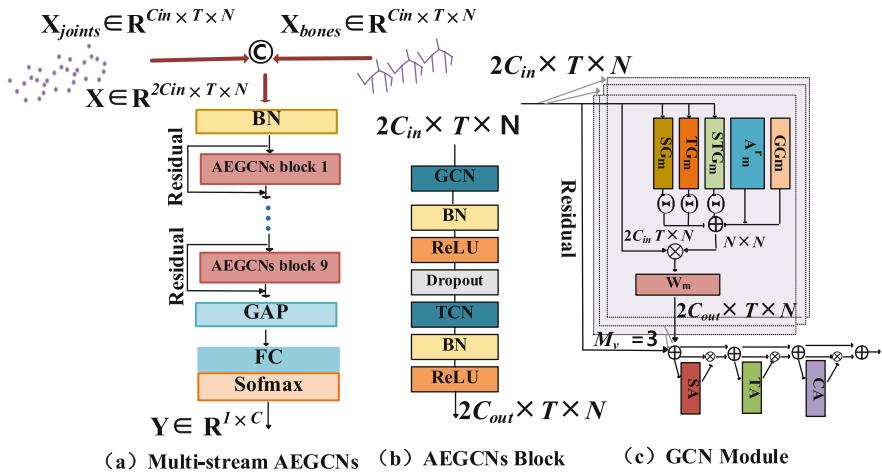


Fig. 1. Illustration of the AEGCNs pipeline: (a) Workflow of the multi-stream AEGCNs (joints, bones); (b) AEGCNs block architecture; the GCN and TCN represents the spatial and temporal GCN respectively, both of which are followed by a BN and ReLu layer; (c) GCN module used in the AEGCNs block. There are four kinds of sub-graphs, i.e., GG_m , SG_m , TG_m , STG_m , A^r_m , and three kinds of attention sub-modules, i.e., SA, TA, CA. Here, \odot denotes the matrix concatenation. \oplus Denotes the elementwise summation. \otimes Denotes the matrix multiplication. M_v Denotes the number of subsets. T Denotes the gating mechanism.

Subsequently, the great effort in skeleton-based SLR has shown graph convolutional networks (GCNs) [2, 7, 9–11] is an effective solution to the skeleton data. In contrast to other methods, GCNs-based can treat the skeleton data as a graph structure representing the body joints (graph nodes) and their natural connections (graph edges). Our GCNs-based SLR method, namely an attention-enhanced graph convolutional networks (AEGCNs), as shown in Fig. 1, mainly contains three improvements: (1) A novel adaptive graph convolutional layer (GCN) is proposed and multiple graphs are parameterized for graph convolution, namely the global graph (GG), the spatial graph (SG), the temporal graph (TG), the spatial-temporal graph (STG). The global graph determines the basic graph topology for SLR. The spatial, temporal, spatial-temporal graph respectively provides more flexibility of the model and brings more generality to adapt to

various data samples. The multiple graphs are fused by used a gating mechanism (Θ), which can adaptively adjust their importance in each of the model layers. (2) Multiple attention-augmented mechanisms, i.e., spatial attention (SA), temporal attention (TA) and spatial-temporal attention (STA), are introduced to select the relatively important information of the joints, frames and channels, which can better understand sign language. (3) The appropriately higher-order connections are well introduced to enlarge the receptive field of graph convolution. (4) To further improve the representation of the movement about the hand and finger, we refer the 2D/3D coordinates of the skeleton data as the first-order information, i.e., the joint, and exploit the second-order information, i.e., the bone between two joints. In detail, the bone information is reformulated as a vector pointing from its source joint to its target joint. Our proposed AEGCNs can effectively learn the features between the joints and bones in end-to-end manner, and notably improve the performance.

2 AEGCNs Method

2.1 Graph Construction

In this study, we define a skeleton sequence as a spatial-temporal graph $G = (V, E, A)$, where $V = \{v_{ti} | t = 1, \dots, T, i = 1, \dots, N\}$ is a set of nodes with N joints and T frames; E is a set of edges, comprised by the subsets of $E_S = \{v_{ti}v_{tj} | (i, j) \in H\}$ and $E_T = \{v_{ti}v_{(t+1)i}\}$, the first subset E_S is a set of the spatial edges corresponding to the natural connections in human joints at each frame, where H is the set of naturally connected human joints, the second subset E_T is a set of the temporal edges corresponding to the same human joint connection in sequent skeletons; $A \in \mathbb{R}^{N \times N}$ denotes the adjacency matrix of graph G , which is a symmetric binary matrix defined as $A_{ij} = 1$ if there is a connection between nodes v_{ti} and v_{tj} in time step t , otherwise $A_{ij} = 0$. Given A as the spatial graph adjacency matrix that represents the joint connections in a single skeleton, the normalized adjacency matrix \hat{A} with self connections is computed as [8]:

$$\hat{A} = D^{-\frac{1}{2}} \tilde{A} D^{-\frac{1}{2}} \quad (1)$$

where $\tilde{A} = A + I$, D is the diagonal degree matrix of \tilde{A} , $D_{ii} = \sum_j^N A_{ij}$.

2.2 Adaptive Graph Convolution

The spatial-temporal GCNs for the skeleton data contains the graph convolution to capture the spatial pattern and the common standard convolution to describe the temporal feature. In this work, we propose an adaptive graph convolutional layer, where the sub-graph topology is optimized together with other parameters of the network in an end-to-end learning manner, i.e., GG, SG, TG, STG. The graph is unique for different layers and samples, which greatly increases the flexibility of the model. Meanwhile, it is designed as a residual branch, which can guarantee the stability of the original model.

In detail, we apply the layered propagation rule of GCNs proposed in [8] to implement the proposed adaptive graph convolutional layer, as shown in Fig. 1(c), which constructs

as follows:

$$X_{out} = \sum_m^M W_m X_{in} (A_m^r + GG_m + \alpha SG_m + \beta TG_m + \gamma STG_m) \quad (2)$$

where, $X_{in} \in \mathbb{R}^{2C_{in} \times T \times N}$ is the input feature vector, C_{in} denotes the input number of channels, T denotes the number of skeletons, N denotes the number of joint at each skeleton. $X_{out} \in \mathbb{R}^{2C_{out} \times T \times N}$ is the output feature vector, C_{out} denotes the out number of channels. M is represents partitioning strategy mentioned in [8], which divides the neighbor nodes of each joint into three subsets, namely $M = 3$, m denotes the index of subset. $W_m \in \mathbb{R}^{2C_{out} \times 2C_{in}}$ is weighting matrix to extract the node features of each partition.

Higher-Order Connection (A_m^r): here, the graph is determined by the adjacency matrix A . The respective field of GCN convolution can be enlarged by the introduction of higher-order connection, while can increase the computational expense compared to one-order connection. Current work [11] discovered that the two-order hop connection was much welcomed than any other one. When $r = 2$, A_m^r is defined as follows:

$$A_m^2 = 4A^2 - A - 2I_n \quad (3)$$

where, A is an $N \times N$ adjacency matrix, which represents the graph of the natural human joint connections, I_n is its identity matrix. when $r = 1$, A_m^1 is the summation of adjacency matrix A and identity matrix I_n , i.e., \tilde{A} , as in Eq. 1. Here, the r set as 2.

The First Sub-graph (GG_m) is the global graph learned from the skeleton data. It represents the graph topology that is more suitable for the SLR task. And it is initialized with the adjacency matrix (A) of the human-hand-based graph in Eq. 1. Compared with \tilde{A} , the elements of GG_m are parameterized and updated together with other parameters in the training process.

The Second Sub-graph (SG_m) is the spatial graph that can learn a unique topology for each sample in spatial dimension. To determine how strong the connections between two vertexes is, we use the normalized Gaussian function to calculate the similarity of the two vertexes as the correlation $P'_{i,j}$ [12]:

$$P'_{i,j} = \frac{\exp(\theta(i)^T \otimes \phi(j))}{\sum_{j=1}^N \exp(\theta(i)^T \otimes \phi(j))} \quad (4)$$

where the \otimes represents matrix multiplication. The $\theta(\cdot)$ and $\phi(\cdot)$ are two embedded functions, and can be achieved by the channel-wise convolution filter. In this way, the dynamic correlations between vertexes can be captured to build the spatial graph. Since the normalized Gaussian is equipped with softmax operation, the SG_m can be computed based on Eq. 4 as follows:

$$SG_m = \text{softmax}\left(X_{in}^T W_{\theta m}^T W_{\phi m} X_{in}\right) \quad (5)$$

where $X_{in} \in \mathbb{R}^{2C_{in} \times T \times N}$ is the input feature vector, and its transpose is X_{in}^T . $W_{\theta m}$ and $W_{\phi m}$ are the parameters of the embedding function $\theta(\cdot)$ and $\phi(\cdot)$, respectively.

The Third Sub-graph (TG_m) is the temporal graph that can learn a unique topology for each sample in temporal dimension. In detail, we introduce two temporal convolutions to extract the temporal information of each vertex correlations with Eq. 4. In this way, the vertex interaction between neighbor frames is involved when we calculate the vertex connections. To this regard, we also introduce a Gaussian function, as in Eq. 4, to computer the vertex correlation and to build individual temporal graph TG_m , as in Eq. 5. The functions $\theta(\cdot)$ and $\phi(\cdot)$ are implemented by the temporal convolution.

The Fourth Sub-graph (STG_m) is the spatial-temporal graph that can learn a unique topology for each sample in spatial-temporal dimension. Here, we build the spatial-temporal graph STG_m is straightforward to use the two graphs: SG_m and TG_m .

Gating Mechanism (Θ): For bottom layers of the AEGCNs network, the receptive fields are small and the features are mostly low-level features, which limits the ability of learning the graph topology from diverse samples. Thus, the global graph, such as GG_m , should be more important in these layers because it is irrelevant with the input features. But for top layers, the model gathers more comprehensive information and the features are more semantic, which provides more diversity and requires more individuality of the graph topology. Thus, the individual graph, such as SG_m , TG_m , STG_m , should be more important because it is constructed based on the input features and is individual for each of the samples.

To address this problem, we apply gating mechanism to adjust the importance of the individual graph for different layers. In detail, the SG_m , TG_m , STG_m is respectively multiplied with parameterized coefficient α , β , γ , which are unique for each layer and are updated in the training process. Here, these α , β , γ are initialized to be 0.

In addition, multiple attention-augmented mechanisms proposed in [9], i.e., SA, TA and STA, are embedded in graph convolutional layer to helps the model paying more attention to the important joint, frame and feature.

2.3 Network Architecture

Figure 1 shows the detailed framework of AEGCNs. In Fig. 1(a) shows the overall architecture of the model, which consists of batch normalization (BN), 9 AEGCNs blocks, a global average pooling (GAP), fully connected layer (FC) and softmax layer. The AEGCNs block is the series of spatial GCN (GCN), temporal GCN (TCN) with the kernel size 9×1 , BN, ReLU and dropout layer with the drop rate set as 0.5, as shown in Fig. 1(b). And the number of output channels for each AEGCNs blocks is 64, 64, 64, 128, 128, 128, 256, 256 and 256. Besides, the residual convolutional neural network (Residual) is applied on each AEGCNs blocks. In addition, to improve the representation of the movement about hand and finger, both the joint data (the coordinates of the joints) and bone data (the direction and length of the bones) are modelled in our framework.

3 Experiments

3.1 Datasets

DEVISIGN-D Dataset [13]. This dataset is a public Chinese dataset for SLR. It contains 500 sign language classes. In total, the dataset contains 6000 videos (4500 trainings,

750 validations, and 750 test instances). In addition, we randomly selected the estimated subset for 50 signs from the DEVISIGN-D dataset, called DSL50. We apply the 3-fold cross-validation to divided the DSL50 dataset (totaling 620 samples) into two subsets (80% training data and 20% test instances).

ASLLVD Dataset [14]. This dataset is a broad public dataset with video sequences of thousands of American Sign Language signs. In total, the ASLLVD dataset contains 10 k samples covering 2745 signs. To validate the proposed approach, we follow the previous works [7] to randomly select 20 signs from the ASLLVD dataset, called ASL20, totaling 1080 samples. We also apply the 3-fold cross-validation to divided into two subsets (720 training and 360 test instances).

3.2 Implementation Details

Training. All experiments are conducted on the Pytorch platform with one NVIDIA GeForce GTX 1070 GPU card. The batch size is 8. We use the stochastic gradient descent (SGD) with Nesterov momentum (0.9). We use a weight decay of 0.0001 and the initial learning rate of 0.001. The learning rate decays for DEVISIGN-D dataset by a factor of 10 at the 25_{th} epoch, 50_{th} epoch, 75_{th} epoch and 100_{th} epoch. While the learning rate decays for DSL50 and ASL20 datasets by a factor of 10 at the 50_{th} epoch, 100_{th} epoch, 150_{th} epoch and 200_{th} epoch.

Data Processing. First, it is necessary to preprocess the aforementioned datasets to be compatible with the input of AEGCNs. The associated skeleton sequence is shown in Fig. 2. A sample input skeleton estimated with the OpenPose library [15] is provided in Fig. 2(a) (60 joint-based skeletons). While the original skeleton included 60 joints, only 28 were considered in this work to reduce runtime and improve algorithm efficiency, as shown in Fig. 2(b). Furthermore, the bones were computed from the 28 joints and formed new training data to further boosting the performance of EAGCNs.

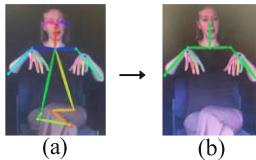


Fig. 2. An illustration of skeletal sequence for the sign “AFRAID”. (a) it estimated by the Openpose with 60 joints. (b) it estimated by the Openpose with 28 joints.

3.3 Ablation Study of Our AEGCNs

Effectiveness of Multiple graph Module. As introduced in Sect. 2.2, there are four kinds of sub-graphs in our proposed adaptive graph convolutional layer, i.e., the global graph (GG), the spatial graph (SG), the temporal graph (TG) and the spatial-temporal

graph (SSG). We test the performance of using each of the graphs along and combining them together. The results are shown as Ours-GG, Ours-SG, Ours-TG, Ours-STG, Ours-SG/TG/STG and Ours-GG/SG/TG/STG in Table 1, respectively. It shows that the four designed graphs can bring notable improvement for the sign language recognition task. With four graphs added together, the model obtains the best performance.

Besides, we verify the effectiveness of introducing the high-order approximations (A^r , shown as Our-Ar/GG/SG/TG/STG in Table 1), which suggests the strategy is better.

Moreover, we verify the effectiveness of adding the gating mechanism (G, shown as Ours-Ar/GG/SG/TG/STG/G in Table 1), which also brings encouraging improvement. Overall, the complete adaptive graph convolutional layer brings improvement of +13.46% and +3.46% compared with ST-GCN on top1 and top5 accuracies, respectively.

Table 1. Evaluation of multiple graphs module on the DSL50 dataset.

Methods	Top1 (%)	Top5 (%)
ST-GCN [7]	76.23	94.26
Ours-GG	86.89	96.46
Ours-SG	84.43	96.09
Ours-TG	85.07	96.16
Ours-STG	85.13	96.28
Ours-SG/TG/STG	86.04	96.34
Ours-GG/S/STG	87.51	97.58
Our-Ar/GG/S/STG	89.34	97.64
Ours-Ar/GG/S/STG/G	89.69	97.72

Effectiveness of Multiple Attention Module. In this section, we test the effectiveness of the proposed Multiple Attention Module introduced in Sect. 2.2. There are three kinds of sub-modules, i.e., spatial attention module (SA), temporal attention module (TA), channel attention module (CA), shown as Ours-GG/SA, Ours-GG/TA and Ours-GG/CA, in Table 2 respectively. It shows that the three sub-modules can help improving the performance. Then we embed the STC-attention module into adaptive graph convolutional layer, shown as Ours-Ar/GG/S/STG/G/SA/TA/CA in Table 2, and obtains improvements of +16.39% and +4.1% compared with ST-GCN on top1 and top5 accuracies, respectively.

Effectiveness of Multi-stream Module. Finally, we test the performance of using two streams and show the results in Table 3. Here, the J and B denote the joint stream and the bone stream respectively. Clearly, the multi-stream method outperforms the single-stream method. For single-stream method, the joint stream (J-Ours) better than the bone stream (B-Ours). This suggests the complementarity of the two streams. By combining the joints and bones (JB-Ours), it brings notable improvement as expected.

Table 2. Evaluation of attention module on the DSL50 dataset.

Methods	Top1 (%)	Top5 (%)
ST-GCN [7]	76.23	94.26
Ours-GG	84.43	96.09
Ours-GG/SA	87.52	96.59
Ours-GG/TA	87.87	96.67
Ours-GG/CA	87.61	96.72
Ours-GG/STC	88.52	97.54
Ours-Ar/B/C/D/CD/G/SA/TA/CA	92.62	98.36

Table 3. Evaluation of multi-stream module on the DSL50 dataset.

Methods	Top1 (%)	Top5 (%)
J-Ours	92.62	98.26
B-Ours	91.80	97.54
JB-Ours	95.09	99.18

3.4 Comparison with the State-of-the-Arts

Comparison Results on DEVISIGN-D Dataset. We compare the final model with the state-of-the-art skeleton-based sign language recognition methods on DEVISIGN-D dataset. The compared methods include ST-GCN [8], AGCN [2] and AAGCN [9]. The results on DEVISIGN-D are shown in Table 4, which shows our method outperforms the state-of-the-art methods on recognition accuracy.

Table 4. Comparison on the DEVISIGN-D dataset.

Methods	Top1 (%)	Top5 (%)
ST-GCN [8]	63.81	87.67
AGCN [2]	72.21	91.87
AAGCN [11]	80.55	94.28
J-Ours	81.13	95.81
B-Ours	80.21	95.54
JB-Ours	82.96	96.07

Comparison Results on ASL20 Dataset. Besides, we make comparisons with state-of-the-art SLR methods on ASL20 dataset. For a fair comparison, we use the same sign language data, and the reported results are shown in Table 5. The methods used for comparisons include traditional methods [16–19], and deep learning methods [7, 20].

Table 5. Comparison on the ASL20 dataset.

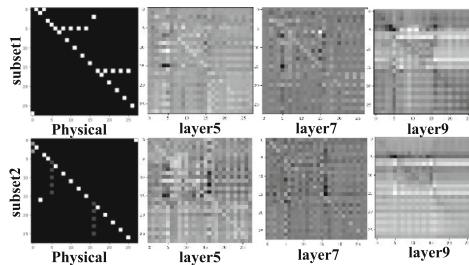
Methods	Accuracy (%)
MHI [18]	10.00
MEI [19]	25.00
PCA [20]	45.00
ST-GCN [7]	56.82
HOF [21]	70.00
BHOF [22]	85.00
Ours	90.32

As shown in Table 5, the proposed model achieves the state-of-the-art performance with a large margin on ASL20, which suggests the superiority of our model. For instance, compared to the baseline of ST-GCN and BHOF, the accuracy of AEGCNs acquires an improvement of 33.5% and 5.32% respectively.

3.5 Visualization of the Learned Graphs

There are four kinds of graphs in our model: the global graph and the three individual graphs. Figure 3 shows an example of the learned adjacency matrices of the global graph for different subsets and different layers. The first and second rows show the adjacency matrices of the centripetal subset (S2) and the centrifugal subset (S3) introduced in [8], respectively. The first column shows the graph structure defined based on the natural connectivity of the human body, i.e., \hat{A} in Eq. 1. Others are the adjacency matrices of the global graph in different layers. The gray scale of each element in the matrix represents the strength of the connection.

It shows that the topology of the learned graph is updated and the changes in the graph topology of the higher layer more than the lower layer. It is possibly because the information contained in higher layer is more semantic.

**Fig. 3.** Example of learned adjacency matrices of the global graph.

Similarly, some examples of the learned adjacency matrices of individual graph for two different samples are included in Figs. 4, 5, and 6. It shows that different samples

and layers need different graph topologies, which confirms our motivation. For higher layers, the temporal individual graph is more preferred, which proves the effectiveness of the proposed temporal function methods.

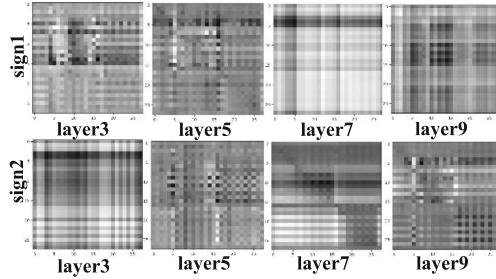


Fig. 4. Example of learned adjacency matrices of the spatial individual graph.

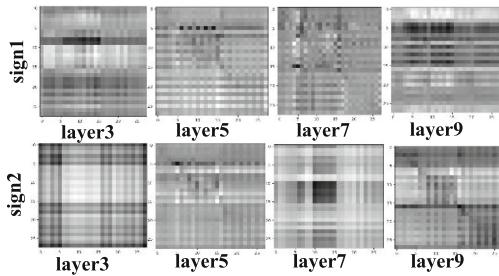


Fig. 5. Example of learned adjacency matrices of the temporal individual graph.

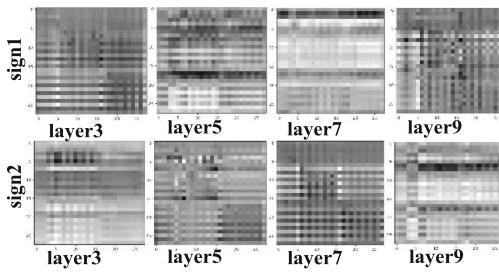


Fig. 6. Example of learned adjacency matrices of the spatial-temporal individual graph.

Figure 7(a) is a visualization of the skeleton graph for different layers of one sample (from left to right is the 3_{th} , 5_{th} , 7_{th} and 9_{th} layers respectively). The size of the circle represents the importance of the corresponding joint. It shows that the model pays more attention on the joints of hands and head. Besides, for lower layers, the strength of the connection between the current joints is not obvious. This result is intuitive since the receptive fields of lower layers are relatively smaller, while the global information cannot be observed.



Fig. 7. Visualization of the graphs. (a) Different layers; (b) different samples.

Figure 7(b) shows a similar visualization of Fig. 7(a) but for different samples. The learned adjacency matrix is extracted from the second subset of the 7_{th} layers. It shows that the graph structures learned by our model for different samples are also different, even for the same convolutional subset and the same layer, which also confirms our motivation that different samples need different topologies of the graph.

3.6 Summary of Performance Analysis

The top1 accuracies of AEGCNs in DEVISIGN-D, DSL50 and ASL20 datasets are 82.96%, 95.09% 90.23% severally, while the top5 accuracies are 96.07%, 99.18% and 100%. In addition, Fig. 8 shows the recognition confusion matrices for testing performance from ASL20 and DSL50 datasets, respectively. As see in the figures, the top1 accuracies are high and the AEGCNs achieves good identification performance across different classes.

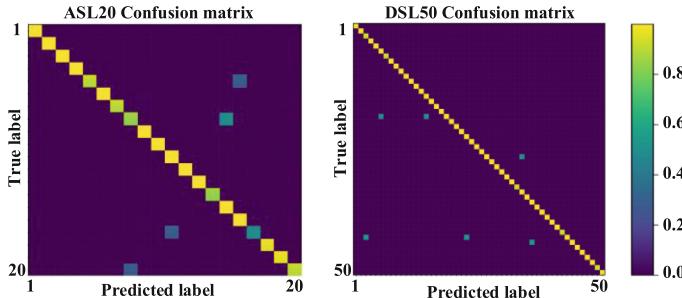


Fig. 8. Confusion matrices for testing performance on the ASL20 and DSL50 datasets.

4 Conclusions

In this work, an attention-enhanced graph convolutional networks (AEGCNs) for the high performance of skeleton-based SLR is proposed. We apply an effective graph convolutional layer with four kinds of adaptive graphs to model the embedded spatial and temporal dynamics. And in order to enlarge the receptive field of GCN convolution, we introduce the higher-order connections with Chebyshev polynomial approximation. Besides, to give higher weight to the relative important joints, frames, features, we propose spatial-temporal-channel attention mechanisms directly embedded in graph convolutional layer. Furthermore, we implement an effective fusion method for the joints

and bones in the AEGCNs framework, which performs well in most classification of SLR. We hope our work could encourage and facilitate future research on sign language processing.

Acknowledgements. We would like to thank the reviewers for their comments to help us improve the quality of this paper. The study is supported by the National Natural Science Foundation of China under Grant no. 62072255.

References

1. Emmorey, K.: Language, cognition, and the brain: insights from sign language research. *Sign Lang. Stud.* **4**(3), 325–328 (2014)
2. Lei, S., Yifan, Z., Jian, C.: Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.12026–12035 (2019)
3. Qinkun, X., Minying, Q., Yuting, Y.: Skeleton-based Chinese sign language recognition and generation for bidirectional communication between deaf and hearing people. *Neural Networks* **125**, 41–55 (2020)
4. Liu, T., Zhou, W., Li, H.: Sign language recognition with long short-term memory. In: IEEE International Conference on Image Processing (ICIP), pp. 2871–2875 (2016)
5. Unutmaz, B., Karaca, A.C., Güllü, M.K.: Turkish sign language recognition us-ing kinect skeleton and convolutional neural network. In: 2019 27th Signal Processing and Communications Applications Conference (SIU), pp.1–4 (2019)
6. Kumar, E.K., Kishore, P., Kumar, M.: 3D sign language recognition with joint distance and angular coded color topographical descriptor on a 2 - stream CNN. *Neurocomputing* **372**(8), 40–54 (2020)
7. de Amorim, C.C., Macêdo, D., Zanchettin, C.: Spatial-temporal graph convolutional networks for sign language recognition. In: Tetko, I.V., Kůrková, V., Karpov, P., Theis, F. (eds.) ICANN 2019. LNCS, vol. 11731, pp. 646–657. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30493-5_59
8. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. *Proc. AAAI Conf. Artif. Intell.* **32**(1) (2018)
9. Shi, L., Zhang, Y., Cheng, J.: Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
10. Jiang, S., Sun, B., Wang, L.: Skeleton Aware Multi-modal Sign Language Recognition (2021). <https://arxiv.org/abs/2103.08833>
11. Peng, W., Hong, X., Zhao, G.: Learning graph convolutional network for skeleton-based human action recognition by neural searching. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2020), pp. 669–2676 (2020)
12. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision* **60**(2), 91–110 (2004)
13. Chai, X., Wang, H., Chen, X.: The devisign large vocabulary of Chinese sign language database and baseline evaluations. In: Technical report VIPL-TR-14-SLR-001. Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS (2014). http://vipl.ict.ac.cn/homepage/ksl/data_ch.html

14. Neidle, C., Thangali, A., Sclaroff, S.: Challenges in development of the American sign language lexicon video dataset (ASLLVD) corpus. In: 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon, LREC (2012). http://vlm1.uta.edu/~athitsos/asl_lexicon/
15. D'Antonio, E., Taborri, J., Palermo, E.: A markerless system for gait analysis based on OpenPose library. In: 2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC). IEEE (2020)
16. Babu, R.V., Ramakrishnan, K.R.: Recognition of human actions using motion history information extracted from the compressed video. *Image Vis. Comput.* **22**(8), 597–607 (2004)
17. Athitsos, V., Neidle, C., Sclaroff, S., Nash, J.: The American sign language lexicon video dataset. In: The IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Anchorage, AK, USA, pp. 1–8. IEEE (2008)
18. Dreuw, P., Stein, D., Deselaers, T.: Spoken language processing techniques for sign language recognition and translation. *Technol. Disabil.* **20**(2), 121–133 (2008)
19. Laptev, I., Marszalek, K., Schmid, C.: Learning realistic human actions from movies. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Anchorage, AK, USA, pp. 1–8. IEEE (2008)
20. Lim, K.M., Tan, A.W., Tan, S.C.: Block-based histogram of optical flow for isolated sign language recognition. *J. Vis. Commun. Image Represent.* **40**, 538–545 (2016)



XGPT: Cross-modal Generative Pre-Training for Image Captioning

Qiaolin Xia^{1(✉)}, Haoyang Huang^{2(✉)}, Nan Duan^{2(✉)}, Dongdong Zhang^{2(✉)},
Lei Ji^{2(✉)}, Zhifang Sui², Edward Cui^{2(✉)}, Taroon Bharti^{2(✉)},
and Ming Zhou^{2(✉)}

¹ MOE Key Laboratory of Computational Linguistics,
Peking University, Beijing, China
{xql,szf}@pku.edu.cn

² Microsoft Research Asia, Beijing, China
{hachua,nanduan,dongdong.zhang,leiji,edwac,tbharti,mingzhou}@microsoft.com

Abstract. In this paper, we propose XGPT, a new method of **Cross-modal Generative Pre-Training** for Image Captioning that is designed to pre-train text-to-image caption generators through four novel generation tasks, including Adversarial Image Captioning (AIC), Image-conditioned Masked Language Modeling (IMLM), Image-conditioned Denoising Autoencoding (IDA), and Text-conditioned Image Feature Generation (TIFG). As a result, the pre-trained XGPT can obtain new state-of-the-art results on the benchmark datasets, including COCO Captions and Flickr30k Captions. We also use XGPT to generate image captions as data augmentation for the image retrieval task and achieve significant improvement on all recall metrics.

1 Introduction

Cross-modal pre-training has substantially advanced the state of the art across a variety of understanding Vision-and-Language (VL) tasks, such as Image-Text Retrieval [16], Visual Question Answering (VQA) [2], Visual Commonsense Reasoning (VCR) [34], Referring Expression Comprehension [17].

However, Vision-and-language generation tasks (e.g., Image Captioning and Text-to-Image Generation) require the model to not only understand cross-modal representations but also have generation capabilities. Some existing encoder-only Vision-language pre-training models are designed for image captioning. For example, VLP [36] proposed a masked span prediction task during pre-training by modifying the self-attention mask and then fine-tuning with image captioning. Oscar [20] extent Vision language pre-training with tag input and more large-scale corpus. It pre-trained with masked token task and contrastive learning, and then applied to downstream tasks like image captioning. These two works are only for encoder-only architecture and limited to a simple span mask task in the pre-training stage.

Motivated by these, we present Cross-modal Generative Pre-Training for Image Captioning (XGPT). The XGPT model uses a encoder-decoder architecture and is directly optimized for VL generation tasks. To pre-train both encoder and decoder, we include four generative pre-training tasks: 1) *Adversarial Image Captioning* (AIC) to empower the model with more robustness during pre-training. Compared with VILLA [9] by adding perturbation into encoder only for understanding tasks, we modify it to adapt both encoder and decoder during image captioning, 2) *Image-conditioned Masked Language Modeling* (IMLM), a span mask generative task to learn the relationship between vision and language by predicting consecutive tokens in the decoder side. 3) *Image-conditioned Denoising Autoencoding* (IDA), to explore text-image alignments during generation by using the corrupted caption and attention matrix between two modalities to recover caption, 4) *Text-conditioned Image Feature Generation* (TIFG), differ from image-to-text generating tasks, this task is designed to leverage underlying semantic in the reverse direction, text-to-image.

In addition to vision-to-language generation, the proposed XGPT can also help in understanding tasks. To verify the idea, we perform data augmentation for image retrieval by using our XGPT as a generator. We retrain a model that has state-of-the-art performance in image retrieval by adding XGPT generated captions for data augmentation, and achieve significant improvement.

Our contributions can be summarized as follows:

- We introduce XGPT, a new method of Cross-modal Generative Pre-Training for Image Captioning, design four novel pre-training tasks that are especially effective for image-to-text generation.
- We achieve state-of-the-art (Sota) results with the same scale pre-training corpus for COCO Captions, Flickr30k on all metrics. We also present extensive experiments and analysis to provide useful insights on the effectiveness of each pre-training task and model variant.
- We employ XGPT to help vision-language understanding tasks like image retrieval by performing data augmentation. After retraining, the model that has state-of-the-art performance still achieves significant improvement on all recall metrics.

2 Related Work

Pre-training for NLP Tasks. Recently, pre-trained language models (LM) over large language corpus have shown great advances for NLP tasks. Three Transformer-based methods that are most relevant to our approach, namely MASS [27], Unicoder [13], and BART [19].

MASS [27] adopts the encoder-decoder framework to predict masked fragments given the remaining part of the sentence. Unicoder is a universal language encoder pre-trained based on three pre-training tasks. The new tasks help the model learn mappings among different languages from more perspectives. BART [19] uses a denoising autoencoder for pre-training. Our method is inspired by these works, but since images are not sequential data, we have to tailor our model for cross-modal tasks in particular.

Pre-training for Cross-modal Generation Tasks. Very recently, several attempts have been made to pre-train models for cross-modal generation tasks. Both VideoBERT [29] and CBT [28] are seeking to conduct pre-training for the video captioning task. But they perform pre-training only for the BERT-based encoder to learn bidirectional joint distributions over sequences of visual and linguistic tokens. So they have to train a separate video-to-text decoder. In contrast, Unified VLP [36] uses a shared multi-layer transformer network for both encoding and decoding. Following UniLM [7], they pre-train the model on two masked language modeling (MLM) tasks, like close tasks designed for sequence-to-sequence LM. So target prediction is still masked tokens, not the whole sentence. However, we find that by using more generative pre-training objectives can outperform Unified VLP significantly on Image Captioning.

Adversarial Training. Adversarial training (AT) [10, 24] is proposed to improve model robustness and withstand adversarial attacks, and has been well studied in computer vision. Some recent works in NLP also tried to explore adversarial training for pre-training [3, 6, 9, 12, 15, 21, 31, 37]. Zhu et al. [37] shows that Transformer-based model (BERT, RoBERTa and ALBERT) can be significantly boosted by adopting large-batch adversarial training. VILLA [9] first propose to add adversarial perturbations to both image and word embedding for understanding tasks and encoder-only model. Inspired by above studies, we propose to combine feature-level adversarial perturbations with image captioning objective and prove that AT can also be effective incorporated for generation task and encoder-decoder architecture.

3 Preliminaries

Linguistic Representation. For each token in the input language sequence, its representation is a sum of token embedding and position embedding. We denote the input tokens as $\mathbf{w} = \{w_1, w_2, \dots, w_M\}$ and the corresponding representations as $\mathbf{x}^T = \{x_1^T, x_2^T, \dots, x_M^T\}$.

Image Representation. For each input image, we first detect objects using a pre-trained Faster R-CNN model [1]. Here, the top 100 objects with highest confidence scores are selected, each of which has a feature vector computed by mean-pooling the last-layer convolutional feature of its region of interest. To represent the position of each object, we construct a 5-d position vector from its spatial location (normalized top-left and bottom-right coordinates) and the fraction of image area it covered. Next, we concatenate the feature vector and position vector of each object and transform it into another vector by linear projection, to make sure the dimensions h of linguistic tokens and visual tokens are identical, and we denote as image regions as $\mathbf{v} = \{v_1, v_2, \dots, v_N\}$ and the corresponding representations as $\mathbf{x}^I = \{x_1^I, x_2^I, \dots, x_N^I\}$.

Image Refining. Unlike words in text, image regions lack a natural order. To better model the relationship among objects in an image, we add an additional image refining layer following AoANet [14] to refine the image representation before feeding them to the encoder.

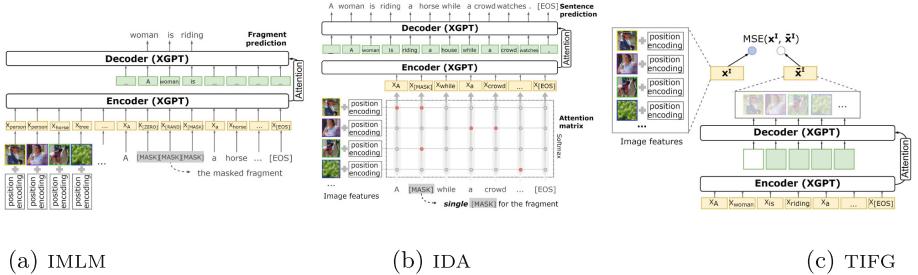


Fig. 1. Cross-modal generative pre-training tasks

4 Cross-modal Generative Pre-Training for Image Captioning

4.1 Model Architecture

Unlike encoder-only models, Unified VLP and Oscar, XGPT applies a encoder-decoder architecture and can be pre-trained through different generative pre-training tasks. Basically, both encoder and decoder are multi-layer Transformer networks. The encoder reads the source image and sentence and generates a set of representations in decoder as introduced in Preliminaries.

4.2 Generative Pre-training Tasks

For image captioning, a basic generative task. It only takes images as inputs (single modality) and generate the caption. We include four new cross-modal generative tasks to enhance image captioning that can jointly pre-train the encoder and decoder.

Adversarial Image Captioning (AIC). A major approach to Image Captioning (IC) is encoder-decoder framework. In natural language processing, adversarial perturbation has proven effective for improving a model’s generalization and robustness [6, 8, 37]. We propose to combine adversarial training with an image captioning objective in this task.

Without adversarial training, the standard objective is to generate the caption \mathbf{w} in an autoregressive manner given the image regions \mathbf{v} , by minimize the negative log-likelihood

$$\mathcal{L}_{\text{IC}} = -\lambda_{\text{IC}} \sum_{t=1}^T \log p_{\theta}(w_t | \mathbf{w}_{<t}, \mathbf{v}) \quad (1)$$

where $\mathbf{w}_{<t}$ is the context history produced by the neural generator, and λ_{IC} is the pre-defined weight of IC loss. This objective requires the model to predict the whole sentence from scratch.

Inspired by Zhu et al. [37], we modify the training objective by applying small perturbation δ to input image embedding \mathbf{x}^I in the encoder and generated text embedding $\mathbf{x}_{<t}^T$ in the decoder. Denote the model as a function $\mathbf{y} = f_\theta(\mathcal{X})$, the number of inner ascent steps as K . We maximize the adversarial loss:

$$\min_{\theta} \mathbb{E} \left[\frac{1}{K} \sum_{i=0}^K \max \mathcal{L}_{AIC}(f_\theta(\mathbf{x}^I + \delta_i^I, \mathbf{x}_{<t}^T + \delta_{<t,i}^T), \mathbf{y}) \right] \quad (2)$$

\mathcal{L}_{AIC} can be solved reliably by PGD like VILLA [9], a standard method for large-scale constrained optimization. Take δ_i^I for example: PGD takes the following step in each iteration:

$$\delta_{i,n+1}^I = \prod_{\|\delta_i^I\| \leq \epsilon} (\delta_{i,n}^I + \alpha g(\delta_{i,n}^I) / \|g(\delta_{i,n}^I)\|_F) \quad (3)$$

where $g(\delta_{i,n}^I) = \nabla_{\delta_i^I} \mathcal{L}_{AIC}(f_\theta(\mathbf{x}^I + \delta_i^I, \mathbf{x}_{<t}^T), y)$ is the gradient of the loss, α is the update steps and $\prod_{\|\delta_i^I\| \leq \epsilon}$ performs a projection onto the ϵ -ball.

Image-conditioned Masked Language Modeling (IMLM). IMLM aims to teach the model to learn the relationship between vision and language by predicting consecutive tokens in the decoder side.

XGPT is trained to reconstruct the n-gram masked words through a sequence to sequence framework. This task is similar in the idea to n-gram MLM in BERT or Masked Seq-to-Seq in MASS. The difference lies in that (1) the task encourages the encoder to learn the cross-modality relationships between the unmasked tokens and image regions, and (2) the decoder has to generate masked tokens of the fragment, and extract useful image-conditioned information from the encoder side.

As shown in Fig. 1(a), we concatenate the regions and unmasked tokens as input to the encoder during pre-training. And we let the decoder predict the masked fragment by minimizing the negative log-likelihood loss:

$$\mathcal{L}_{IMLM} = -\lambda_{IMLM} \sum_{t=1}^M \log p_\theta(w_t | \mathbf{w}_{<t}, \hat{\mathbf{w}}, \mathbf{v}) m_t \quad (4)$$

where $\hat{\mathbf{w}}$ is the corrupted caption, and $m_t = 1$ if w_t is in the masked fragment, and 0 otherwise. λ_{IMLM} is the pre-defined weight of IMLM loss.

Image-conditioned Denoising Autoencoding (IDA). In IDA we take distance between feature spaces of two modalities into account, and use an attention matrix to model the underlying text-image alignments. Besides, IDA forces the model to reconstruct the whole sentence without giving it the length of the masked fragment, as illustrated in Fig. 1(b). Specifically, we use *single [MASK] token for fragment prediction*. Inspired by text filling task in [19], we first sample n-gram fragments to be mask and, then, replace each fragment with a single [MASK] token. This is more challenging because the model have to predict not

only the missing tokens but also the length of the original sentence. To model the text-image alignments, we first compute an *image-text attention matrix* for each token-region pair (w_i, v_j) : $A_{ij} = W[x_i^T, x_j^I, x_i^T \odot x_j^I]$ where $W \in \mathbf{R}^{3 \times h}$ is a trainable weight and \odot is elementwise multiplication. Then, we represent each word as the weighted sum of all region representations based on the attention matrix: $x_i^T = \sum_{j=1}^N \text{softmax}(A_{ij})x_j^I$. Finally, XGPT takes new x_i^T as input and tries to predict the original word sequence \mathbf{w} . The loss function is defined as

$$\mathcal{L}_{\text{IDA}} = -\lambda_{\text{IDA}} \sum_{t=1}^M \log p_\theta(w_t | \mathbf{w}_{<t}, \hat{\mathbf{w}}, \mathbf{v}) \quad (5)$$

where $\hat{\mathbf{w}}$ is the corrupted caption, and λ_{IDA} is the pre-defined weight of IDA loss.

Text-conditioned Image Feature Generation (TIFG). Text-to-image (T2I) generation can be regarded as the inverse problem of image captioning, a Image-to-Text (I2T) problem. It is natural and reasonable to unify the model to leverage the underlying semantic in both domains.

In contrast to Uniter [5], which involved image feature generation by learning to regress the transformer output of each masked region to its visual features, TIFG aims to regress the decoder output of all image regions conditioned on text descriptions rather than only the masked regions. As shown in Fig. 1(c), we employ the encoder-decoder pipeline to convert linguistic representations into \bar{x}_i^I of the same length and dimension as image representations \mathbf{x}_i^I . Then we train with mean squared error to supervise the XGPT to generate semantically consistent image features. Mathematically, this loss can be expressed as:

$$\mathcal{L}_{\text{TIFG}} = \lambda_{\text{TIFG}} \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_i^I - \bar{\mathbf{x}}_i^I\|_2^2 \quad (6)$$

where λ_{TIFG} is the weight of TIFG loss.

4.3 XGPT Pre-training

We calculate task-specific loss in turns and update the model for each task in every pre-training iteration. To study how each objective works for pre-training, we include these tasks with individual weight of loss.

The initial weight of the individual weight of loss is set to $\lambda(T) = 1$, where T indicates epoch number. XGPT will gradually shift to IC.

We first conduct *out-of-domain pre-training* on Conceptual Captions (CC) dataset [26] which contains about 3M image-caption pairs scraped from alt-text enabled web images. The automatic collection leaves some noise (e.g., not relevant and too short) in the dataset but brings a massive scale.

Then we use data from downstream tasks with the proposed pre-training objectives for *in-domain pre-training*. This step allows the model to adapt to the target domain. So we reduce the weights of the cross-modal tasks (e.g., $\lambda_{\text{IMLM}}, \lambda_{\text{IDA}}, \lambda_{\text{TIFG}} \rightarrow 0.3$) and keep the image caption task $\lambda_{\text{IC}} = 1$ unchanged.

During the *fine-tuning* stage, the model only takes image features and position information as input, and the decoder is trained to predict the whole sentence in an autoregressive manner.

The two-stage pre-training takes about 8 d to converge on 8x V100 GPUs with a total batch size of 512 by gradient accumulation. For adversarial image captioning, we set $\alpha = 3$ during PGD training. We fine-tuned the model 30 epochs with four GPUs.

5 Experiments and Results

5.1 Datasets

The datasets for downstream tasks include COCO Captions [4] and Flickr30k [33]. In these datasets, each image is labeled with 5 captions. We follow Karpathy’s split¹, which gives 113.2k/5k/5k and 29.8k/1k/1k images for train/val/test splits respectively. We use standard metrics for Image Captioning, including BLEU@4, METEOR, CIDEr, SPICE, to evaluate the propose method and compare with other methods.

5.2 Implementation Details

In all experiments, the backbone Transformer of XGPT follows Vaswani et al. [30], and we modify it to the BERT-based encoder-decoder architecture with 768 hidden units, 8 heads, GLEU activations used as GPT [11]. The dropout rate is 0.1. We train XGPT with mixed-precision training and FP16, which makes use of GPUs more efficiently. The Adam [18] with $\beta_1 = 0.9$, $\beta_2 = 0.98$ is used for optimization. The learning rate is varying from $1e - 4$ to $2e - 5$ for out-of-domain pre-training with invert square root decay [30]. During the in-domain pre-training and fine-tuning stage, we take an average of the top-4 pre-trained weights and reduce the initial learning rate to $1e - 5$. For caption inference, we use greedy search on the validation set and beam search with beam size 5 on the test set.

5.3 Results and Analysis

Comparisons against SotAs. We compare with several existing methods on image captioning, including BUTD [1], VLP [36], AoANet [14] and OSCAR [20]. We also apply CIDEr-D optimization [25] to improve model performance with Self-Critical Sequence training.

Table 1 summarizes the overall results of two tasks. As shown in the table, our baseline method XGPT outperforms all previous works that do not use pre-training, like BUTD and AoANet, on C, M, S. It only slightly worse than AoANet [14] on B@4, partially because AoANet applied AoA to both the encoder and the decoder.

¹ cs.stanford.edu/people/karpathy/.

Table 1. Comparison with the previous state-of-the-art methods. **Bold** indicates best value overall. OSCAR (6.5M) [20] is pre-trained on 6.5 million pairs, while Unified VLP (3M) [36] and XGPT (3M) are pre-trained on 3 million pairs, which is a subset of OSCAR’s corpus. C,B@4,M,S stand for CIDEr, BLEU@4, METEOR, SPICE.

Model	Flickr30k				COCO			
	C	B@4	M	S	C	B@4	M	S
Approaches that <i>do not</i> optimize for CIDEr								
BUTD [1]	56.6	27.3	21.7	16.0	113.5	36.2	27.0	20.3
NBT (with BBox) [23]	57.5	27.1	21.7	15.6	107.2	34.7	27.1	20.1
GCN-LSTM (spa) [32]	–	–	–	–	115.6	36.5	27.8	20.9
GCN-LSTM (sem) [32]	–	–	–	–	116.3	36.8	27.9	20.9
GVD [35]	62.3	27.3	22.5	16.5	–	–	–	–
AoANet [14]	–	–	–	–	119.8	37.2	28.4	21.3
Unified VLP (3M) [36]	67.4	30.1	23.0	17.0	116.9	36.5	28.4	21.2
OSCAR (6.5M) [20]	–	–	–	–	123.7	36.5	30.3	23.1
XGPT (3M)	72.0	32.1	24.2	18.4	122.1	36.9	29.6	22.5
Approaches that <i>do</i> optimize for CIDEr								
BUTD [1]	–	–	–	–	120.1	36.3	27.7	21.4
Unified VLP (3M) [36]	–	–	–	–	129.3	39.5	29.3	23.2
AoANet [14]	–	–	–	–	129.8	38.9	29.2	22.4
OSCAR (6.5M) [20]	–	–	–	–	137.6	40.5	29.7	22.8
XGPT (3M)	–	–	–	–	133.4	40.2	29.5	22.6

Compare to other pre-trained models, XGPT is very different in the way that the four generative pre-training tasks are used and the model architecture is based on encoder-decoder, as outlined in Fig. 1. With same amount of image-text pairs, XGPT outperforms the Unified VLP [36] by a large margin, e.g., improving M and C by 1 and 5 points. The results demonstrate the effectiveness of our proposed pre-training scheme. Comparing to OSCAR [20], another pre-training model which uses much more training data, XGPT which use less than half of the data only performs slightly better on B@4.

Effectiveness of Proposed Tasks. We analyze the effectiveness of different pre-training tasks through ablation studies over COCO Captions and Flickr30K. The results are shown in Table 2.

As for the out-of-domain pre-training stage, AIC outperforms IC by a large margin (+1.5 on CIDEr), demonstrating that adversarial training is beneficial for the pre-training and fine-tuning task. There are also significant improvements across all three tasks (comparing Row 3,4,5 with Row 1 baseline). Among the three, we observe IDA which helps the model to learn text-image alignments achieves the biggest improvement, while TIFG the smallest. This is probably because of the discrepancy of the decoder which is originally designed to predict captions and the task objective which is to predict all image region features. When combining all four tasks, we achieve the highest gain of approximately +4.6 on CIDEr over Row 1.

Table 2. Ablation analysis of pre-training tasks on COCO Captions.

Stage	Pre-training Tasks	COCO			
		C	B@4	M	S
Out-of-domain (CC)	IC	116.4	35.9	28.2	21.1
	AIC	117.9	36.1	28.3	21.2
	IC + IMLM	117.7	36.2	28.2	21.2
	IC + IDA	118.1	36.4	28.3	21.3
	IC + TIFG	117.3	36.0	28.2	21.2
	IC + IMLM + IDA + TIFG	119.3	36.7	28.4	21.6
	AIC + IMLM + IDA + TIFG	121.0	36.7	29.4	22.3
Out-of-domain (CC) + In-domain (COCO)	IC + IMLM	119.1	36.7	28.5	21.5
	IC + IDA	119.2	36.6	28.5	21.6
	IC + TIFG	118.2	36.4	28.4	21.3
	IC + IMLM + IDA + TIFG	120.1	37.2	28.9	21.8
	AIC + IMLM + IDA + TIFG	122.1	36.9	29.6	22.5

Table 3. impacts of weight strategies Evaluation results on COCO Captions.

Model	C	B@4	M	S
Tiny EncDec	110.8	33.7	27.6	20.5
Tiny EncDecShare	112.7	34.6	27.9	20.7

Parameters Sharing. To find the best model setting for image captioning, we also designed two model variants. **EncDec** is a Transformer encoder-decoder architecture in which all weights are initialized randomly. **EncDecShare** is like EncDec, but the parameters for self-attention in encoder and cross-attention between encoder and decoder are shared. We add a signal to distinguish whether keys and values are from the encoder output or decoder input. complexity won't change but the amount of parameters has been reduced. Table 3 reports results of these settings. Tiny Enc model which simply reuses the encoder for decoding can outperform Tiny EncDec by a large margin, e.g., improving CIDEr by 1.9. This is probably because the shared structure can help model to leverage underlying relationship between two modalities. We use this as the optimal architecture in Table 1.

5.4 Data Augmentation for Image Retrieval

In addition to generation tasks, our XGPT can also help vision-and-language understanding tasks, such as image retrieval, by performing data augmentation as an image description generator. Image retrieval is a task of identifying an image from a pool given a caption describing its content. We generate 62k more captions for all 29k images (about 2.1 captions for each) in the Flickr30k training set, which originally contains 145k captions. We continue to fine-tune the

open-source state-of-the-art model² introduced in [22] on the combination of the augmentation and the original training data.

ViLBERT got 58.2 on R@1, 84.9 on R@5, and 91.5 on R@10. While ViLBERT trained on augmented data got 60.4 on R@1, 86.4 on R@5, and 91.9 on R@10. The improvement is significant (2.2% on R@1, 1.5% on R@5, and 0.4% on R@10). The higher relative gain on R@1 also indicates that the generator can produce high-quality image captions which can help the model better understand images.

Table 4. Two examples of generated captions for given images. Underlined text shows the difference between captions.

Example A



Human-generated captions

A person trying desperately not to be photographed by putting their sweater ...
A person wearing red pants hides their head under a black jacket in front of ...
A person with red pants with cover over her head sitting in front of multiple ...
The woman tries to hide from work under a black sweatshirt, but her red ...
a child is hiding under a sweater in a chair.

XGPT-generated captions

A woman wearing red pants is sitting at a desk in front of a computer.
A person in a blue sweatshirt is sleeping in a chair.
A woman in a blue sweatshirt is sleeping in a chair in front of a computer.

Example B



Human-generated captions

A young boy wearing a black shirt, BROWN pants and a black watch has his ...
A young man wearing a black shirt takes a folding chair from a large stack.
A person helping to set up chairs for a big event.
A young man in a black shirt stacks chairs.
A boy is setting up folding chairs.

XGPT-generated captions

A man in a black t shirt and black shorts is putting up a white chair.
A man in a black t shirt and black t shirt works on a folding chair.

5.5 Qualitative Studies

A positive and a negative example in the generation results are provided in Table 4. In Example A, we can see that XGPT-generated captions are grammatically and semantically correct, and also can increase the diversity of the data. In Example B, the first sentence contains wrong information (brown→black); the second has a duplicated phrase. Both can be considered as noise.

6 Conclusion

In this paper, we present XGPT, Cross-modal Generative Pre-Training for Image Captioning. Three main pre-training tasks are proposed and the ablation study

² https://github.com/jiasenlu/vilbert_beta.

shows that the effectiveness of each task is different. The combination of all tasks achieves stronger performance on all evaluation metrics suggested that they are complementary to each other. After in-domain and out-of-domain pre-training, XGPT outperforms state-of-the-art models by a significant margin. For future works, we are curious about extending XGPT to cross-modal understanding tasks, such as VQA and VCR.

Acknowledgments. This paper is supported by the National Key Research and Development Program of China 2020AAA0106700 and NSFC project U19A2065.

References

1. Anderson, P., et al.: Bottom-up and top-down attention for image captioning and visual question answering. In: CVPR, pp. 6077–6086 (2018)
2. Antol, S., et al.: Vqa: Visual question answering. In: ICCV, pp. 2425–2433 (2015)
3. Chen, T., Liu, S., Chang, S., Cheng, Y., Amini, L., Wang, Z.: Adversarial robustness: From self-supervised pre-training to fine-tuning. In: CVPR, pp. 699–708 (2020)
4. Chen, X., et al.: Microsoft coco captions: data collection and evaluation server. arXiv preprint [arXiv:1504.00325](https://arxiv.org/abs/1504.00325) (2015)
5. Chen, Y.C., et al.: Uniter: learning universal image-text representations. arXiv preprint [arXiv:1909.11740](https://arxiv.org/abs/1909.11740) (2019)
6. Cheng, Y., Jiang, L., Macherey, W.: Robust neural machine translation with doubly adversarial inputs. In: ACL, pp. 4324–4333 (2019)
7. Dong, L., et al.: Unified language model pre-training for natural language understanding and generation. arXiv preprint [arXiv:1905.03197](https://arxiv.org/abs/1905.03197) (2019)
8. Ebrahimi, J., Rao, A., Lowd, D., Dou, D.: Hotflip: white-box adversarial examples for text classification. In: ACL, pp. 31–36 (2018)
9. Gan, Z., Chen, Y.C., Li, L., Zhu, C., Cheng, Y., Liu, J.: Large-scale adversarial training for vision-and-language representation learning. [arXiv:2006.06195](https://arxiv.org/abs/2006.06195) (2020)
10. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint [arXiv:1412.6572](https://arxiv.org/abs/1412.6572) (2014)
11. Hendrycks, D., Gimpel, K.: Bridging nonlinearities and stochastic regularizers with gaussian error linear units (2016)
12. Hendrycks, D., Lee, K., Mazeika, M.: Using pre-training can improve model robustness and uncertainty. In: ICML (2019)
13. Huang, H., et al.: Unicoder: a universal language encoder by pre-training with multiple cross-lingual tasks. arXiv preprint [arXiv:1909.00964](https://arxiv.org/abs/1909.00964) (2019)
14. Huang, L., Wang, W., Chen, J., Wei, X.Y.: Attention on attention for image captioning. In: ICCV, pp. 4634–4643 (2019)
15. Jiang, H., He, P., Chen, W., Liu, X., Gao, J., Zhao, T.: Smart: robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. arXiv preprint [arXiv:1911.03437](https://arxiv.org/abs/1911.03437) (2019)
16. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: CVPR, pp. 3128–3137 (2015)
17. Kazemzadeh, S., Ordonez, V., Matten, M., Berg, T.: Referitgame: referring to objects in photographs of natural scenes. In: EMNLP, pp. 787–798 (2014)
18. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. international conference on learning representations (2015)

19. Lewis, M., et al.: Bart: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint [arXiv:1910.13461](https://arxiv.org/abs/1910.13461) (2019)
20. Li, X., et al.: Oscar: object-semantics aligned pre-training for vision-language tasks. arXiv preprint [arXiv:2004.06165](https://arxiv.org/abs/2004.06165) (2020)
21. Liu, X., et al.: Adversarial training for large neural language models. arXiv preprint [arXiv:2004.08994](https://arxiv.org/abs/2004.08994) (2020)
22. Lu, J., Batra, D., Parikh, D., Lee, S.: Vilbert: pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In: Advances in Neural Information Processing Systems, pp. 13–23 (2019)
23. Lu, J., Yang, J., Batra, D., Parikh, D.: Neural baby talk. In: CVPR, pp. 7219–7228 (2018)
24. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: ICLR (2018)
25. Rennie, S.J., Marcheret, E., Mroueh, Y., Ross, J., Goel, V.: Self-critical sequence training for image captioning. In: ICCV, pp. 7008–7024 (2017)
26. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: a cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: ACL, pp. 2556–2565 (2018)
27. Song, K., Tan, X., Qin, T., Lu, J., Liu, T.Y.: Mass: masked sequence to sequence pre-training for language generation. arXiv preprint [arXiv:1905.02450](https://arxiv.org/abs/1905.02450) (2019)
28. Sun, C., Baradel, F., Murphy, K., Schmid, C.: Contrastive bidirectional transformer for temporal representation learning. ArXiv [abs/1906.05743](https://arxiv.org/abs/1906.05743) (2019)
29. Sun, C., Myers, A., Vondrick, C., Murphy, K., Schmid, C.: Videobert: a joint model for video and language representation learning. ICCV (2019)
30. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
31. Wang, D., Gong, C., Liu, Q.: Improving neural language modeling via adversarial training. arXiv preprint [arXiv:1906.03805](https://arxiv.org/abs/1906.03805) (2019)
32. Yao, T., Pan, Y., Li, Y., Mei, T.: Exploring visual relationship for image captioning. In: ECCV, pp. 684–699 (2018)
33. Young, P., Lai, A., Hodosh, M., Hockenmaier, J.: From image descriptions to visual denotations: new similarity metrics for semantic inference over event descriptions. TACL **2**, 67–78 (2014)
34. Zellers, R., Bisk, Y., Farhadi, A., Choi, Y.: From recognition to cognition: visual commonsense reasoning. In: CVPR, pp. 6720–6731 (2019)
35. Zhou, L., Kalantidis, Y., Chen, X., Corso, J.J., Rohrbach, M.: Grounded video description. In: CVPR, pp. 6578–6587 (2019)
36. Zhou, L., Palangi, H., Zhang, L., Hu, H., Corso, J.J., Gao, J.: Unified vision-language pre-training for image captioning and vqa. In: AAAI (2020)
37. Zhu, C., Cheng, Y., Gan, Z., Sun, S., Goldstein, T., Liu, J.: Freelb: enhanced adversarial training for language understanding. [arXiv:1909.11764](https://arxiv.org/abs/1909.11764) (2019)



An Object-Extensible Training Framework for Image Captioning

Yike Wu, Ying Zhang^(✉), and Xiaojie Yuan

College of Computer Science, Nankai University, Tianjin 300350, China
wuyike@dbis.nankai.edu.cn, {yingzhang,yuanxj}@nankai.edu.cn

Abstract. Recent years have witnessed great progress in image captioning based on deep learning. However, most previous methods are limited to the original training dataset that contains only a fraction of objects in the real world. They lack the ability to describe other objects that are not in the original training dataset. In this paper, we propose an object-extensible training framework that enables a widely-used captioning paradigm to describe objects beyond the original training dataset (i.e., extended objects) by generating high-quality training data for these objects automatically. Specifically, we design a general replacement mechanism, which replaces the object (An object includes the object region in the image, and the corresponding object word in the caption) in the original training dataset with the extended object to generate new training data. The key challenge in the proposed replacement mechanism is that it should be context-aware to get the meaningful result that complies with common knowledge. We introduce the multi-modal context embedding to ensure that the generated object representation is coherent in the visual context and the generated caption is smooth and fluent in the linguistic context. Extensive experiments show that our method improves significantly over the state-of-the-art methods on the held-out MSCOCO in both automatic and human evaluation.

Keywords: Image captioning · Extended objects · Context-aware replacement

1 Introduction

Image captioning is an important task in the intersection between computer vision and natural language processing. We have witnessed much progress in image captioning based on deep learning. However, most previous methods can only describe objects in the original training dataset, but lack the ability to generate captions for

This research is supported by the NSFC-Xinjiang Joint Fund (No. U1903128), NSFC General Technology Joint Fund for Basic Research (No. U1836109, No. U1936206), Natural Science Foundation of Tianjin, China (No. 18ZXZNGX00110, No. 18ZXZNGX00200), and the Fundamental Research Funds for the Central Universities, Nankai University (No. 63211128).

other objects in the real world. For example, if the original training dataset contains the image-text pairs of “giraffe” but not that of “zebra”, a caption model built upon it can describe an image with a giraffe but fails to understand one with a zebra.

The key issue lies in the limitation of the original training dataset that is manually constructed and contains only a small fraction of objects in the real world. Supposing we have a “complete” training dataset covering all objects, we can use it to train a caption model that can describe any object. Therefore, to enable a caption model to describe objects not in the original training dataset, a naive solution is to manually construct additional training data for such objects. However, this process is time-consuming and laborious, which hinders its feasibility in realistic applications. A question naturally arises: can we automatically generate training data for such objects without manual efforts?

We find that the UpDn model [2] provides convenience for us to achieve the automatic generation. It represents the input image by object regions instead of a single feature vector [15] or spatial grids [17], which means it does not require direct access to the original image and only uses the object representation instead. Extensive works (e.g., [4, 9, 13]) follow this captioning paradigm and all use the object representation, which we define as *UpDn-style caption model*. One merit of such models is that it makes generating training data for an object simple. For example, we want to create a new image-text pair of the object “zebra” that is not in the original training dataset. Suppose that we already have an original image with the caption “a giraffe walking across the grass next to some antelope” as shown in Fig. 1. We could simply replace the giraffe region in the object representation of the original image by the zebra region from another unpaired image¹ to generate the object representation for the new image. And we don’t need to generate the new image itself, which is a relatively hard task, as the UpDn-style caption model only needs the object representation as input. To generate the corresponding caption for this new image, we can simultaneously do the replacement of the object word “giraffe” in the original caption and get “a zebra walking across the grass next to some antelope”.

In this paper, we propose an object-extensible training framework that enables the UpDn-style caption model to describe objects beyond the original training dataset (i.e., *extended objects*) by generating new training data for these objects automatically. Specifically, we introduce a general replacement mechanism which replaces the object region and object word in the original training dataset simultaneously with the object region and object word of an extended object. The generated data can be used to train any UpDn-style caption model as the input of the UpDn-style caption model is the object representation of an image rather than the image itself. The entire process of data generation and model training is automatic and requires no additional manual efforts.

The key challenge in the proposed replacement mechanism is to ensure that the replacement result is meaningful and complies with common knowledge. In the example of Fig. 1, if we replace the “giraffe” region-word pair (i.e., object region

¹ This image is not paired with a caption and easy to obtain without manual efforts.

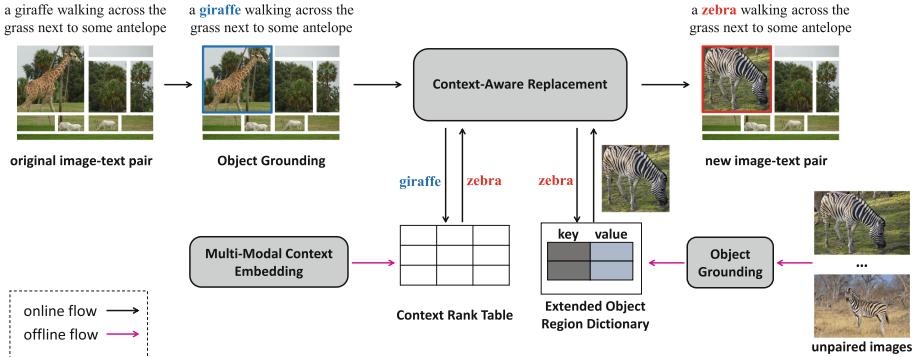


Fig. 1. The framework of the general replacement mechanism.

and object word) with a “bus” region-word pair instead of the “zebra” region-word pair, we will get ridiculous results in two aspects. First, as the bus region is not likely to appear together with grass regions or antelope regions in a real valid image, the resulting object representation is not meaningful. Second, in the caption after replacement, “a bus walking across the grass ...”, “bus” does not collocate with “walking” in the natural language. Thus, to ensure that the replacement is meaningful, we need to consider both the visual context of the object region and the linguistic context of the corresponding object word. To this end, we propose the context-aware replacement (CAR), which uses the multi-modal context embedding to find the replacement with the most similar visual context and linguistic context to the given object. In summary, our contributions are three-fold:

- We propose an object-extensible training framework that enables the UpDn-style caption model to describe extended objects by a general replacement mechanism.
- We introduce the multi-modal context embedding to make the replacement process aware of the visual context and linguistic context.
- Extensive experimental results show that the proposed method outperforms the state-of-the-art methods on the held-out MSCOCO dataset.

2 Related Work

In recent years, image captioning methods based on deep learning have made much progress [2, 12, 13, 15, 17, 19]. However, most of them can only describe the objects in the original training dataset that is manually constructed, and are difficult to be generalized to other objects in the real world.

Some approaches have been proposed to solve this problem. Deep Compositional Captioner (DCC) [3] pretrains a lexical classifier and a language model on unpaired image/text data respectively, and composes them into a caption model. It further trains the caption model on image-text pairs and transfers knowledge between semantically-related words. Venugopalan *et al.* [14] extend

DCC by jointly training the lexical classifier, language model and caption model in an end-to-end manner, which obviates the explicit transfer and achieves better performance. More recently, Yao *et al.* [18] incorporate the copy mechanism into the caption model, which can not only generate a word from the language model but also copy one from objects detected in the image. Li *et al.* [5] further consolidate the method [18] by the pointing mechanism and coverage of objects. In addition, Mogadala *et al.* [8] annotate entity labels for images with the guidance of knowledge base, and build the semantic attention and constrained inference over these entity labels. Another approach [1] proposes the constrained beam search, which forces the visual tags of the image to appear in the generated caption during the inference process. Furthermore, the Decoupled Novel Object Captioner (DNOC) [16] first generates a sentence with placeholders, and then retrieves object words from a key-value object memory to fill them. Neural Baby Talk [7] shares a similar spirit with DNOC, which first generates a sentence with slots tied to object regions in the image, and then fills the slots by the corresponding object words.

Previous works usually design a special model architecture for image captioning to incorporate more objects, which is tightly coupled with the architecture itself and difficultly generalized. In contrast, our solution tackles the problem in a data-driven way, which is fully compatible with any UpDn-style caption model and thus can seamlessly benefit from its potential improvement.

3 Methodology

3.1 Framework Overview

The general replacement mechanism is shown in Fig. 1, which is composed of the online flow and the offline flow. Given an image-text pair (\mathbf{R}, S) in the original training dataset \mathcal{D}_o , we feed it into the online flow to get a new image-text pair (\mathbf{R}', S') . We perform this procedure on all image-text pairs in \mathcal{D}_o to obtain an extended training dataset \mathcal{D}_e , which contains not only objects in \mathcal{D}_o but also the extended objects. Finally, we use \mathcal{D}_e to train a caption model that can generate captions for all the objects in $W_{obj} \cup W_{ext}$, where W_{obj} and W_{ext} denote the vocabulary of objects in \mathcal{D}_o and that of extended objects respectively.

Online Flow. The input is an image-text pair (\mathbf{R}, S) in the original training dataset \mathcal{D}_o . The symbol $\mathbf{R} = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_o^*, \dots, \mathbf{r}_M\}$ is the object representation of an image and $S = \{w_1, w_2, \dots, w_o^*, \dots, w_N\}$ is the corresponding caption, where \mathbf{r} and w denote an object region and a word respectively. First, from the input we extract the object word $w_o^* \in W_{obj}$ and identify its corresponding object region \mathbf{r}_o^* via the object grounding. Then, we replace the region-word pair (\mathbf{r}_o^*, w_o^*) by a new pair (\mathbf{r}_e^*, w_e^*) of an extended object through the context-aware replacement. Finally, the online flow outputs a new image-text pair (\mathbf{R}', S') for the extended object, where $\mathbf{R}' = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_e^*, \dots, \mathbf{r}_M\}$ and $S' = \{w_1, w_2, \dots, w_e^*, \dots, w_N\}$.

Offline Flow. Before the data generation of online flow, we offline construct two data structures leveraged by the context-aware replacement: (1) We build

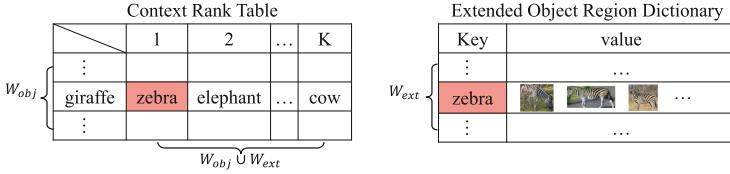


Fig. 2. The details of two data structures constructed by the offline flow.

the *context rank table* with the multi-modal context embedding, which will be used to find the extended object word w_e^* with the most similar visual and linguistic context to w_o^* . (2) We generate the *extended object region dictionary* by the object grounding, which will be queried with w_e^* as the key to find the corresponding object region r_e^* .

Caption Model. We employ the UpDn model [2] as a representative of UpDn-style caption models to verify the effectiveness of our method. The model details are elaborated in the previous work [2] and we will not go into them since our focus is the proposed framework in this work.

3.2 Multi-modal Context Embedding

We construct the context rank table based on the similarity of visual and linguistic context between object words, which is measured by the cosine similarity of their multi-modal context embeddings. The structure of context rank table is shown in Fig. 2. Each row corresponds to an object word in W_{obj} , and each column corresponds to a rank value which is assigned to an object word in $W_{obj} \cup W_{ext}$. In the corresponding row of an object word $w_o \in W_{obj}$, we rank each object word in $W_{obj} \cup W_{ext}$ from high to low according to the cosine similarity between its multi-modal context embedding and that of w_o , and only keep the top K rank values to ensure that the object words in the row are similar enough to w_o in both visual and linguistic context.

Now we focus on how to obtain the multi-modal context embedding of an object word. The general idea is to align the visual representation of the object region and the linguistic representation of the corresponding object word in a common latent space. We train a model composed of an object detector, a visual MLP layer f_{vis} , a linguistic MLP layer f_{lin} , and an embedding matrix E initialized with the pretrained GloVe embedding [10]. The input of the model is an image with its object labels $L = \{l\}$, which is composed of the corresponding object words of objects contained in the image. In the training process, we first extract the object representation \mathbf{R} of the image by an object detector, and map each object region $\mathbf{r} \in \mathbf{R}$ into the common latent space by the layer f_{vis} . Then, we also map the corresponding object word (i.e., each label $l \in L$) into the common latent space by applying the layer f_{lin} on its embedding in E . Next, we define the score function which measures how likely the object region \mathbf{r} is to contain the label l as follows:

$$sc(\mathbf{r}, l) = \text{cos_sim}(f_{vis}(\mathbf{r}), f_{lin}(E(l))), \quad (1)$$

where cos_sim means cosine similarity. Furthermore, for the entire image \mathbf{R} , the score function of containing the label l is defined as:

$$\text{sc}(\mathbf{R}, l) = \max(\text{sc}(\mathbf{r}, l)), \quad \mathbf{r} \in \mathbf{R}. \quad (2)$$

The greater value of $\text{sc}(\mathbf{R}, l)$ means the image \mathbf{R} is more likely to contain the label l and vice versa. Finally, we define the training loss on the image \mathbf{R} :

$$\mathcal{L}(\mathbf{R}) = \sum_{l \in L} \sum_{l' \in \{L_U - L\}} \max[0, 0.1 - \text{sc}(\mathbf{R}, l) + \text{sc}(\mathbf{R}, l')], \quad (3)$$

where L_U denotes the complete set of labels of all images in the training data, and l' is a label that does not appear in the image \mathbf{R} . Minimizing $\mathcal{L}(\mathbf{R})$ is equivalent to increasing $\text{sc}(\mathbf{R}, l)$ and decreasing $\text{sc}(\mathbf{R}, l')$ simultaneously, which forces the labels in L to approach the image \mathbf{R} and keep other labels in $\{L_U - L\}$ away from it in the common latent space. After training, we use $f_{\text{lin}}(E(l))$ as the multi-modal context embedding of l , which is the projection of l in the common latent space.

The cosine similarity of the multi-modal context embeddings can reflect the similarity of object words in both visual and linguistic context. On the one hand, the training loss $\mathcal{L}(\mathbf{R})$ makes the labels in similar images (i.e., with a similar visual context) close to each other in the common latent space. On the other hand, we have already incorporated the linguistic context into the training process at the beginning by initializing E with the pretrained GloVe embedding.

3.3 Object Grounding

The object grounding module grounds an object word to its corresponding object region in the image. In the proposed method, we leverage this module to (1) ground the object word w_o^* to its corresponding object region \mathbf{r}_o^* in the original image and (2) build the extended object region dictionary. Next, we elaborate how we achieve the above two goals respectively.

Ground w_o^* to \mathbf{r}_o^* . Given the object word w_o^* , we explore two kinds of strategies to find its corresponding object region \mathbf{r}_o^* in the object representation \mathbf{R} of the original image. The first kind of strategy requires ground-truth bounding boxes of the image that are manually annotated. We first pick out the ground-truth bounding boxes with the object category corresponding to w_o^* , denoted as \mathbf{B} , and then identify \mathbf{r}_o^* as follows:

$$\mathbf{r}_o^* = \{\mathbf{r} \in \mathbf{R} | \text{IoU}(\mathbf{r}, \mathbf{b}) > T\}, \quad (\mathbf{r}, \mathbf{b}) \in \mathbf{R} \times \mathbf{B}, \quad (4)$$

where $T \in [0.0, 1.0]$ is the threshold value of IoU. The second kind of strategy requires no manual efforts and is more general. It leverages the object categories of object regions in \mathbf{R} , which are output by the object detector. Specifically, we regard all the object regions in \mathbf{R} with the object category corresponding to w_o^* as the object region \mathbf{r}_o^* .

Build the Extended Object Region Dictionary. The structure of the extended object region dictionary is shown in Fig. 2. Each key corresponds to an extended object word $w_e \in W_{ext}$, and its value consists of a series of object regions corresponding to w_e from different unpaired images. Each object region \mathbf{r}_e in the value is obtained by grounding w_e in the object representation \mathbf{R} of an unpaired image.

There are also two kinds of strategies for grounding w_e to \mathbf{r}_e . The first needs ground-truth bounding boxes of the image, while the second leverages the object categories with their confidence scores output by the object detector. In the first kind of strategy, we identify \mathbf{r}_e as follows:

$$\mathbf{r}_e = \underset{\mathbf{r} \in \mathbf{R}}{\operatorname{argmax}} \text{IoU}(\mathbf{r}, \mathbf{b}), \quad (\mathbf{r}, \mathbf{b}) \in \mathbf{R} \times \mathbf{B}, \quad (5)$$

where \mathbf{B} denotes the ground-truth bounding boxes with the object category corresponding to w_e . In the second kind of strategy, we first find out all the object regions with the object category corresponding to w_e in the object representation \mathbf{R} , and then pick the one with the highest confidence score as \mathbf{r}_e .

Note that the grounding of the object word w_o^* and that of the extended object words in W_{ext} are slightly different, which makes the replacement more precise and thus improves the quality of the new generated image-text pair. When grounding w_o^* to \mathbf{r}_o^* , we adopt a relatively loose screening condition to find out all the object regions possibly corresponding to w_o^* in the original image, which means that the notation \mathbf{r}_o^* may represent multiple object regions instead of only one. Since the object detector may output multiple object regions that largely overlap for the same object in an image, this loosely grounding can guarantee all of them can be completely removed in the replacement. When building the extended object region dictionary, we ground each extended object word $w_e \in W_{ext}$ to only the most accurate object region in an unpaired image. In this way, when we replace \mathbf{r}_o^* with \mathbf{r}_e^* , we guarantee the object region \mathbf{r}_e^* added into the object representation exactly contains the extended object.

3.4 Context-Aware Replacement for Automatic Data Generation

Given an image-text pair from the original training dataset \mathcal{D}_o , we generate a new image-text pair of the extended object in the context-aware replacement. We replace the object word w_o^* in the original caption by the extended object word w_e^* , and replace the object region \mathbf{r}_o^* corresponding to w_o^* in the object representation of the original image by the extended object region \mathbf{r}_e^* corresponding to w_e^* . We describe the context-aware replacement in Algorithm 1.

To ensure the replacement result is meaningful, we need to find the region-word pair (\mathbf{r}_e^*, w_e^*) with the most similar context to the region-word pair (\mathbf{r}_o^*, w_o^*) . First, we extract the corresponding row of w_o^* from the context rank table, and select the most top-ranked extended object word in the row as w_e^* . Then, we take w_e^* as the key to retrieve its corresponding value from the extended object region dictionary, and randomly select an object region as \mathbf{r}_e^* from a series of object regions in the value. Note that we do not perform the replacement if there

Algorithm 1. Context-Aware Replacement (CAR)**Input:**

- 1: An image-text pair (\mathbf{R} , S) containing a region-word pair (\mathbf{r}_o^* , w_o^*);
- 2: Context rank table CRT;
- 3: Extended object region dictionary EORD.

Output:

- 4: A new image-text pair (\mathbf{R}' , S').
- 5: $\text{CRT}(w_o^*) \leftarrow$ corresponding row of w_o^* in CRT
- 6: **if** $W_{ext} \cap \text{CRT}(w_o^*) \neq \emptyset$ **then**
- 7: $w_e^* \leftarrow$ top-ranked element in $W_{ext} \cap \text{CRT}(w_o^*)$
- 8: EORD(w_e^*) \leftarrow the value of key w_e^* in EORD
- 9: $\mathbf{r}_e^* \leftarrow$ retrieve an object region from EORD(w_e^*)
- 10: $S' \leftarrow$ in S , replace w_o^* by w_e^*
- 11: $\mathbf{R}' \leftarrow$ in \mathbf{R} , replace \mathbf{r}_o^* by \mathbf{r}_e^*
- 12: **return** (\mathbf{R}' , S')
- 13: **else**
- 14: do not perform the replacement
- 15: **end if**

is no extended object word in the corresponding row of w_o^* in the context rank table, which means we can not find a replacement similar enough to the object in the original image-text pair in both visual and linguistic context.

For each image-text pair in \mathcal{D}_o , we perform the context-aware replacement to generate a new image-text pair. Finally, we gather all the new image-text pairs, and combine them with \mathcal{D}_o to obtain an extended training dataset \mathcal{D}_e . Comparing with training on \mathcal{D}_o , the additional computation cost of training on \mathcal{D}_e is empirically sub-linear, since each image-text pair in \mathcal{D}_o yields at most one new image-text pair (sometimes the replacement will not be successfully performed as mentioned above). This indicates that our method can scale up on datasets with different sizes.

4 Experiments

4.1 Experimental Setup

Dataset. For the convenience of comparing with previous works, we evaluate our method on the held-out MSCOCO dataset, a widely-used benchmark [3] for image captioning on objects not in the original training dataset. The dataset consists of four splits: *training*, *validation*, *test* and *rest*. Follow the previous work [3], we employ a subset of MSCOCO [6] training set as the training split, which excludes all the image-text pairs containing at least one of the eight objects: *bottle*, *bus*, *couch*, *microwave*, *pizza*, *racket*, *suitcase*, *zebra*. The eight objects are used as the extended objects in this setting. We use 50% of MSCOCO validation set as the validation split, and set aside the other 50% for the test split. We take the excluded part in MSCOCO training set as the rest split.

Table 1. Performance (%) on held-out MSCOCO test split.

Model	bottle	bus	couch	microwave	pizza	racket	suitcase	zebra	Avg. F1	CIDEr	METEOR	SPICE
DCC [3]	4.6	29.8	45.9	28.1	64.6	52.2	13.2	79.9	39.8	59.1	21.0	13.4
NOC [14]	14.9	69.0	43.8	37.9	66.5	65.9	28.1	88.7	51.8	–	20.7	–
Base+T4 [1]	16.3	67.8	48.2	29.7	77.2	57.1	49.9	85.7	54.0	77.9	23.3	15.9
KGA-CGM [8]	26.4	54.2	42.1	50.9	70.8	75.3	25.6	90.7	54.5	–	22.2	14.6
LSTM-C [18]	29.7	74.4	38.8	27.8	68.2	70.3	44.8	91.4	55.7	–	23.0	–
DNOC [16]	33.0	76.9	54.0	46.6	75.8	33.0	59.5	84.6	57.9	–	21.6	–
NBT [7]	14.0	74.8	42.8	63.7	74.4	19.0	44.5	92.0	53.2	84.0	23.9	16.6
LSTM-P [5]	28.7	75.5	47.1	51.5	81.9	47.1	62.6	93.0	60.9	88.3	23.4	16.6
CAR	29.4	75.7	49.7	56.0	73.5	18.7	50.3	94.4	56.0	101.9	26.1	19.3
CAR + T2	37.4	78.5	52.2	58.7	76.6	39.2	56.1	94.5	61.7	100.1	25.8	19.2

In the experiment, we use the training split as the original training dataset \mathcal{D}_o (351134 image-text pairs), and generate 302179 new image-text pairs to obtain the extended training dataset \mathcal{D}_e (653313 image-text pairs). We perform the validation and testing on the corresponding splits respectively. There is no overlap of data between model training and evaluation.

Evaluation. On the one hand, we evaluate the captioning performance on automatic metrics. On the other hand, We also compute F1-score for the eight extended objects respectively. For an image in the test split, we regard it as a true positive example of an extended object only if its generated caption and at least one of its ground-truth captions both mention the object.

Implementation Details. For each image, we take a pretrained Faster R-CNN [11] as the object detector to extract 36 object regions as its object representation. This is aligned with the strong baselines DNOC and NBT which also use the Faster R-CNN feature. Additionally, considering the generality, we perform object grounding based on the output of Faster R-CNN, instead of ground-truth bounding boxes (We discuss the difference in Sect. 4.5). In the context rank table, we set the value of K to 20.

4.2 Comparison with SOTA Methods

We compare our method context-aware replacement (abbreviated as CAR) with state-of-the-art methods in Table 1. We can see that our method CAR achieves comparable average F1-score (Avg. F1) of extended objects compared to the SOTA methods, which shows that our approach successfully generates captions for extended objects. By using the constrained beam search [1] (CAR + T2), our method achieves the best average F1-score (61.7%) while maintaining decent captioning performance. However, the results on F1-score can only reflect that the generated caption correctly mentions the corresponding object word of the extend object that appears in the image. We should also focus on the overall captioning performance. We observe that CAR significantly outperforms all the SOTA methods on automatic metrics. Particularly, CAR improves over the competitive baseline LSTM-P by 13.6% on CIDEr, 2.7% on METEOR and 2.7% on SPICE. This indicates that the new generated training data is high-quality enough for training a caption model to generate natural and fluent captions.

Table 2. Human evaluation (%) on a sampled subset of held-out MSCOCO test split. The notation “both” means the judgement holds in both criteria.

Judgement	CAR vs. UpDn			CAR vs. NBT		
	object coverage	consistency	both	object coverage	consistency	both
CAR is better	69.67 ± 0.02	43.00 ± 0.03	38.33 ± 0.06	46.33 ± 0.06	37.67 ± 0.11	23.00 ± 0.02
UpDn/NBT is better	9.00 ± 0.09	25.33 ± 0.00	7.00 ± 0.09	24.33 ± 0.16	35.67 ± 0.11	16.67 ± 0.04
two models are equal	21.33 ± 0.10	29.33 ± 0.04	13.67 ± 0.11	31.67 ± 0.02	26.67 ± 0.39	13.00 ± 0.08

4.3 Human Evaluation

To complement the automatic metrics, we re-implement the strong baseline NBT [7], and perform the human evaluation on a sampled subset of the held-out MSCOCO test split to compare our method CAR with it. We also take the UpDn model [2] for comparison. For each image, we generate three captions with the compared models respectively, and randomly shuffle them to avoid potential bias. We ask three human evaluators to compare the generated captions in pair.

Evaluation Criteria. Given two captions generated by different models for the same image, the evaluators make a judgement about which one is better in two aspects respectively. The first is *object coverage*. This criterion reflects how well the caption covers the objects in the image. If the image contains an extended object, we also tell the evaluators to focus more on it. The second is *consistency*. It measures how consistent the caption is with the image content.

Evaluation Results. We report the results of human evaluation in Table 2. Comparing with both UpDn and NBT, our method CAR generates more captions which are better on either object coverage or consistency. Considering the two criteria simultaneously, our approach also outperforms the other methods.

4.4 Qualitative Examples

As shown in Fig. 3, our method CAR describes the extended objects in all the examples while the other methods not, which verifies its effectiveness of incorporating the extended objects into the caption generation. The red bounding box in an image indicates the object region with the largest attention weight when CAR generate the highlighted word. We observe that red bounding boxes fit well with the extended objects in the images, which reflects that our method really learns to ground the extended objects in the images correctly.

4.5 Discussion

Ablation Study. We compare our method CAR with its two variants in Table 3a: 1) UpDn [2]. It represents an UpDn-style caption model which is trained only on the original training dataset. 2) General Replacement. Besides the original training dataset, it also generates new training data for extended objects by the proposed replacement mechanism to assist the model training. However, it

Table 3. Discussion on the ablation study of our approach CAR and the effectiveness of using the ground-truth bounding boxes.

(a) Ablation study to demonstrate contributions from “replacement (R)” and “context-aware (CA)” in CAR.

Model	R	CA	Avg. F1
UpDn			0.0
General Replacement	✓		48.4
CAR	✓	✓	56.0

(b) Performance on held-out MSCOCO test split without/with leveraging the ground-truth bounding boxes.

Model	Avg. F1	CIDEr	METEOR	SPICE
CAR	56.0	101.9	26.1	19.3
CAR + bbox	56.6	102.2	26.3	19.4



UpDn: a wooden bench sitting next to a green bench
NBT: a green park bench sitting on a sidewalk
CAR: a bottle is sitting on a green bench



UpDn: a man walking down a street next to a traffic light
NBT: a man is sitting on a bike in the street
CAR: a man walking down the street with a suitcase



UpDn: a truck is parked next to a herd of sheep
NBT: a group of sheep standing next to each other
CAR: a bus is parked next to a group of sheep



UpDn: a woman standing on a sidewalk holding a kite
NBT: a woman is standing on a tennis court
CAR: a woman in a hat is holding a racket

Fig. 3. Qualitative examples of captions generated by different methods.

does not consider the visual context and linguistic context, and just randomly selects an extended object as the replacement.

First, by adding the “replacement (R)”, general replacement performs much better than UpDn on average F1-score, while UpDn cannot generate captions for any extended object (Avg. F1 is 0.0%). This validates the effectiveness of the proposed replacement mechanism on describing extended objects. Second, by further adding the “context-aware (CA)”, CAR increases 7.6% on average F1-score. This indicates that it is necessary to ensure that the replacement result is meaningful and complies with common knowledge, which improves the quality of generated training data and thus is beneficial to model training.

Using Ground-Truth Bounding Boxes. As shown in Table 3, the performance of our method is further boosted by leveraging the ground-truth bounding boxes (CAR + bbox) to perform the object grounding. This is reasonable since better grounding will lead to more precise replacement and thus improve the quality of generated training data.

5 Conclusion

In this paper, we propose an object-extensible training framework based on a general replacement mechanism, which focuses on the training data generation of extended objects and is compatible with any UpDn-style caption model. It paves a new data-driven way to generate captions for extended objects. To ensure that the generated data is meaningful and complies with common knowledge,

we introduce the multi-modal context embedding to make the replacement process aware of both visual context and linguistic context. It guarantees that the generated object representation is coherent in visual context and the generated caption is smooth and fluent in linguistic context. Extensive experiments conducted on held-out MSCOCO shows that our method outperforms the SOTA methods in both automatic and human evaluation.

References

1. Anderson, P., Fernando, B., Johnson, M., Gould, S.: Guided open vocabulary image captioning with constrained beam search. In: EMNLP, pp. 936–945 (2017)
2. Anderson, P., et al.: Bottom-up and top-down attention for image captioning and visual question answering. In: CVPR, pp. 6077–6086 (2018)
3. Anne Hendricks, L., Venugopalan, S., Rohrbach, M., Mooney, R., Saenko, K., Darrell, T.: Deep compositional captioning: describing novel object categories without paired training data. In: CVPR, pp. 1–10 (2016)
4. Huang, L., Wang, W., Chen, J., Wei, X.Y.: Attention on attention for image captioning. In: ICCV, pp. 4634–4643 (2019)
5. Li, Y., Yao, T., Pan, Y., Chao, H., Mei, T.: Pointing novel objects in image captioning. In: CVPR, pp. 12497–12506 (2019)
6. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
7. Lu, J., Yang, J., Batra, D., Parikh, D.: Neural baby talk. In: CVPR, pp. 7219–7228 (2018)
8. Mogadala, A., Bista, U., Xie, L., Rettinger, A.: Describing natural images containing novel objects with knowledge guided assistance. In: ACM Multimedia (2017)
9. Pan, Y., Yao, T., Li, Y., Mei, T.: X-linear attention networks for image captioning. In: CVPR, pp. 10971–10980 (2020)
10. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: EMNLP, pp. 1532–1543 (2014)
11. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NeurIPS, pp. 91–99 (2015)
12. Rennie, S.J., Marcheret, E., Mroueh, Y., Ross, J., Goel, V.: Self-critical sequence training for image captioning. In: CVPR, pp. 7008–7024 (2017)
13. Shi, Z., Zhou, X., Qiu, X., Zhu, X.: Improving image captioning with better use of caption. In: ACL, pp. 7454–7464 (2020)
14. Venugopalan, S., Anne Hendricks, L., Rohrbach, M., Mooney, R., Darrell, T., Saenko, K.: Captioning images with diverse objects. In: CVPR, pp. 5753–5761 (2017)
15. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: a neural image caption generator. In: CVPR, pp. 3156–3164 (2015)
16. Wu, Y., Zhu, L., Jiang, L., Yang, Y.: Decoupled novel object captioner. In: ACM Multimedia, pp. 1029–1037 (2018)
17. Xu, K., et al.: Show, attend and tell: Neural image caption generation with visual attention. In: ICML, pp. 2048–2057 (2015)
18. Yao, T., Pan, Y., Li, Y., Mei, T.: Incorporating copying mechanism in image captioning for learning novel objects. In: CVPR, pp. 6580–6588 (2017)
19. Zhao, S., Sharma, P., Levinboim, T., Soricut, R.: Informative image captioning with external sources of information. In: ACL, pp. 6485–6494 (2019)



Relation-Aware Multi-hop Reasoning for Visual Dialog

Yao Zhao^{1,2}, Lu Chen^{1,2(✉)}, and Kai Yu^{1,2(✉)}

¹ X-LANCE Lab, Department of Computer Science and Engineering,
MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University,
Shanghai, China

{zy410397555, chenluszz, kai.yu}@sjtu.edu.cn

² State Key Lab of Media Convergence Production Technology and Systems,
Beijing, China

Abstract. Visual dialog is a multi-modal task that requires a dialog agent to answer a series of progressive questions grounded in an image. In this paper, we propose Relation-aware Multi-hop Reasoning Network (i.e. R2N for short) for visual dialog tasks, which can perform multi-hop reasoning during visual co-reference resolution process in a recurrent way. At each hop, in order to fully understand the visual scene in the image, a Relation-aware Graph Attention Network is used, which encodes each image into graphs with multi-type inter-object relations via a graph attention mechanism. Moreover, we find that the auxiliary clustering mechanism on answer candidates is conducive to model’s performance. We evaluate R2N on VisDial v1.0 dataset. Experimental results on the VisDial v1.0 dataset demonstrate that the proposed model is effective and outperforms compared models.

1 Introduction

Multi-modal researches have increasingly drawn more interest, particularly across computer vision and natural language processing. Researchers have achieved inspiring progress in various vision-language tasks, including visual relation detection [12], image captioning [17], and visual question answering (VQA) [1]. Apart from these, visual dialog [4] is another vision-language task introduced. It can be regarded as the generalization of VQA, in which the agent needs to answer a list of questions, or rather, to make a multi-round dialog, with a certain image as the visual information and a dialog history as the contextual information.

Inspired by the pattern of how humans think in a real dialog, it is necessary to conduct a multi-hop reasoning process to answer the question in visual dialog. Who is asked a question about an image will firstly review the dialog history to collect some fundamental semantic information about pronouns and some other co-references mentioned in the question and then scan the given image to ground it so as to gain visual information. After integrating information from the two

sides, comprehensive visual-semantic information is generated and then is able to recall more detailed semantic information from the dialog history in his mind to update the comprehensive information and examine the image again based on previous information. This pattern follows a recurrent way.

Besides this, due to a significant semantic gap between image and natural language, we believe it is necessary to introduce additional information to build a bridge between the two. For example, given an image of a tennis player holding a racket, existing models may recognize the player and the racket, but not the semantic interaction and geometric position relationship between the two. Therefore, it is difficult to answer questions such as “Is the player swinging his racket?” or “Where is the tennis ball?”. Because they require the visual dialog system to recognize not only the objects (“player”, “racket”), but also the semantics about actions (“swing”, “hold”) and locations (“left”, “above”) in corresponding image, question and dialog history.

To address the two problems, we propose the model **Relation-aware Multi-hop Reasoning Network**, or **R2N** for short, which comprise the Recurrent Attention Encoder as the core module for multi-hop reasoning so as to simulate the thinking patterns of natural people. It acquires information from both a textual encoder and an image features encoder simultaneously and conducts a reasoning process among the question, the dialog history, and the image in order to utilize abundant latent information from the two aspects. On the other hand, with relation-aware graph attention networks, the image features encoder can extract relation features between visual objects, in order to explicitly introduce relation information into image encoding process. Therefore, R2N can handle this multi-hop reasoning task which can act like a refiner to resolve higher-order interactions, including semantic and geometric interactions, between the representation of question and dialog history and the features of image areas iteratively.

Our main contributions are fourfold:

- We introduce a recurrent model structure to visual dialog task to execute a multi-hop reasoning process and meanwhile guarantee information interactions among modalities refined in each hop.
- We introduce the explicit relation features of visual objects into the task and our ablation experiments and results indicate that they facilitate the model to understand relations between visual objects.
- We use an auxiliary clustering mechanism on answer candidates list and our experiments show that it helps to improve the NDCG (Normalized Discounted Cumulative Gain) [3, 4] score.
- The experimental results show that our proposed R2N outperforms previous models and achieves the best results among compared models.

2 Related Work

2.1 Visual Dialog

Visual dialog (VisDial) is a task that requires a dialog agent to answer a series of questions grounded in an image. Different from visual question answering(VQA), the series of questions should capture a temporally-semantic context from a dialog history and utilize visually-grounded information. The task was proposed by [4] accompanied by a corresponding dataset VisDial. Attention-based approaches were previously proposed to address this challenge. The work [9] proposed Neural Module Network [9] to resolve visual coreference with specific modules. DAN [7] consists of two attention modules to retrieve the history to clarify ambiguous questions and perform visual grounding via attention mechanism.

2.2 Multi-hop Reasoning

Nevertheless, these are single-hop approaches, whose ability of reasoning is limited and they neglect latent information of the interactions among the question, the dialog history, and the image. Several researchers also investigate multi-hop reasoning approaches. For example, work [5] propose ReDAN to infer the answer progressively through multiple reasoning steps. The semantic representation is updated based on the image and the previous dialog history, and a recurrently-refined representation is used for further reasoning in the subsequent steps. Work [2] propose DMRM which has a dual-channel to capture the question- and history-aware image features and the question- and image-aware dialog history features by a multi-hop reasoning process in each channel.

However, these models basically either omits relations between visual objects or merely allows for implicit relations. Due to the heterogeneity between textual modality and visual modality, it is necessary to introduce explicit relations, like spatial relation and semantic relation.

3 Background

In this section, we introduce relation-aware graph attention networks (RGATs) for the graphs with labeled edges, which are the basis of our proposed model.

RGAT is a special type of networks that operates on graph-structured data with attention mechanisms. Given a graph $G = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} and \mathcal{E} are the set of nodes x_i and the set of edges e_{ij} , respectively. Each edge has an edge type τ_e ¹, which represents the relation between x_i and x_j . $\mathcal{N}(x_i)$ denotes the nodes which are directly connected by x_i . $\mathcal{N}^+(x_i)$ is the set including x_i and all its direct neighbors. we have $\mathcal{N}^+(x_i) = \mathcal{N}(x_i) \cup \{x_i\}$.

Each node x_i in the graph has an initial feature $\mathbf{h}_i^0 \in \mathbb{R}^d$, where d is the feature dimension. The representation of each node is iteratively updated by the graph attention operation. At the l -th step, each node x_i aggregates context

¹ For brevity, we use τ_e to denote $\tau_{e_{ij}}$.

information by attending over its neighbors and itself. The updated representation \mathbf{h}_i^l is calculated by the weighted average of the connected nodes:

$$\mathbf{h}_i^l = \sigma \left(\sum_{x_j \in \mathcal{N}^+(x_i)} \alpha_{ij}^l (\mathbf{W}_0^l \mathbf{h}_j^{l-1} + \boldsymbol{\gamma}_e) \right), \quad (1)$$

where $\sigma(\cdot)$ is a nonlinear activation function, e.g. ReLU, and the attention coefficient α_{ij}^l is calculated as:

$$\alpha_{ij}^l = \text{softmax}_j \left((\mathbf{W}_1^l \mathbf{h}_i^{l-1})^T (\mathbf{W}_2^l \mathbf{h}_j^{l-1} + \boldsymbol{\beta}_e) \right), \quad (2)$$

where \mathbf{W}_0^l , \mathbf{W}_1^l and $\mathbf{W}_2^l \in \mathbb{R}^{d \times d}$ are learnable parameters for projections. $\boldsymbol{\gamma}_e$ and $\boldsymbol{\beta}_e$ are bias terms for edge type or relation τ_e .

For brevity, we use RGAT(\cdot) to denote the function shown in Eq. (1), i.e.

$$\mathbf{h}_i^l = \text{RGAT} \left(\mathbf{h}_i^{l-1}, \{(\mathbf{h}_j^{l-1}, \tau_e) | x_j \in \mathcal{N}^+(x_i)\} \right). \quad (3)$$

When each edge doesn't have the edge type, or we don't take the edge type into consideration, the bias items $\boldsymbol{\gamma}_e$ and $\boldsymbol{\beta}_e$ will be omitted in Eq. (1) and Eq. (2). RGAT will degenerate into vanilla graph attention network (GAT), which is denoted by GAT(\cdot). Accordingly, Eq. (3) is revised as follows,

$$\mathbf{h}_i^l = \text{GAT} \left(\mathbf{h}_i^{l-1}, \{\mathbf{h}_j^{l-1} | x_j \in \mathcal{N}^+(x_i)\} \right). \quad (4)$$

4 Proposed Model

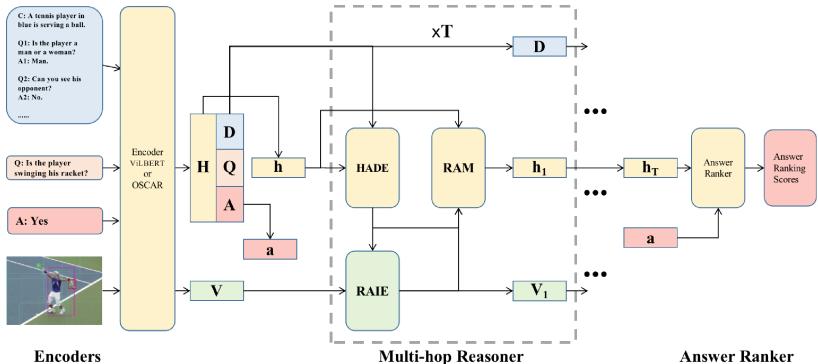


Fig. 1. Model architecture overview. From left to right are Encoders for both modalities, the Multi-hop Reasoner (MR) and the Answer Ranker (AR). The MR will do reasoning process for T times, where T is a hyper-parameter. Bolded letters like \mathbf{H} , \mathbf{D} , \mathbf{Q} , \mathbf{A} etc. denote feature tensors, and their details will be explained in following paragraphs.

Our proposed model can be divided into three parts: Encoders (**Enc**), Multi-hop Reasoner (**MR**) and Answer Ranker (**AR**). In which, the Multi-hop Reasoner consists of three parts also: History Aware Dialog Encoder (**HADE**), Relation Aware Image Encoder (**RAIE**) and Recurrent Attention Manager (**RAM**). Figure 1 illustrates the overview of R2N.

4.1 Encoders

Cross-Modal Encoders. Visual-linguistic multi-modal pretrained models have significantly improved the performance in visual-textual tasks. ViLBERT [13], which is the original one, uses a two-stream structure to model visual and textual streams separately and then use cross-attention mechanism to do alignment and fusion on the two streams. We use it as one encoder alternative. OSCAR [11] uses objects' tags as additional information to facilitate modality fusion. We use it as our second alternative. Both of them output a sequence containing visual features and textual features.

Outputs of Encoders. For brevity, we use $\mathbf{O} = \{\mathbf{H}, \mathbf{V}\}$ to denote the outputs of encoders, where $\mathbf{H} = \{\mathbf{h}^i\}_{i=1}^M \in \mathbb{R}^{M \times d_{hidden}}$ is textual features and $\mathbf{V} = \{\mathbf{v}^j\}_{j=1}^N \in \mathbb{R}^{N \times d_{hidden}}$ is visual features, and M is the number of textual features while N is the number of visual features. Then pooled features are gained for both modalities with linear layers and are denoted by \mathbf{h} (textual) and \mathbf{v} (visual). With mask matrices, we can split \mathbf{H} 's rows into sections corresponding to the question (**Q**), the dialog history (**D**) and an answer candidate (**A**).

4.2 Multi-hop Reasoner

Multi-hop reasoner is comprised of three sub-modules - History Aware Dialog Encoder (**HADE**), Relation Aware Image Encoder (**RAIE**) and Recurrent Attention Manager (**RAM**). History Aware Dialog Encoder extracts dialog history relating to the current query, and Relation Aware Image Encoder captures object features as well as relations between objects based on dialog history and question. Recurrent Attention Manager introduces the explicit multi-hop reasoning process in reference resolution and carries out the modal fusion operation.

History-Aware Dialog Encoder. In this section, we describe the history-aware dialog encoder (**HADE**). Similar to the REFER module from [7], relevant history is extracted by attention mechanism. We then use pooled textual representation \mathbf{h} as query key to obtain the useful information \mathbf{h}_c from dialogue context,

$$\mathbf{h}_c = \text{GAT}(\mathbf{h}, \mathbf{D}). \quad (5)$$

where \mathbf{h} represents the contextual vector and \mathbf{D} is the dialog history matrix, each of its column \mathbf{d}_i is one token's representation in dialog history.

Finally, we obtain the history-aware question representation $\hat{\mathbf{h}}$ by:

$$\hat{\mathbf{h}} = \mathbf{h}_c \oplus \mathbf{h}. \quad (6)$$

For brevity, we denote above process as the following function:

$$\hat{\mathbf{h}} = \text{HADE}(\mathbf{h}, \mathbf{D}). \quad (7)$$

Relation-Aware Image Encoder. Relation-aware Image Encoder (RAIE) aims to extract object features and capture relations between objects from the image. Inspired by the works [12] and [10], we define three kinds of relations: *semantic* relation, *spatial* relation and *implicit* relation. An example is shown in Fig. 2.

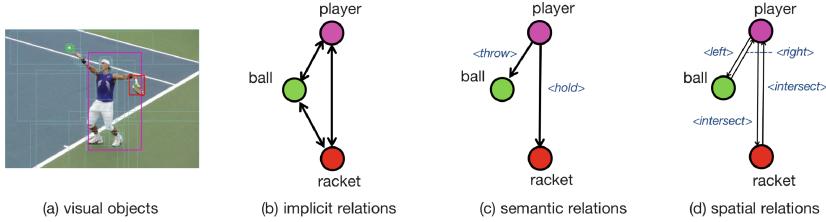


Fig. 2. Visual objects are boxed in (a), the features of each visual object in (a) consist of size, position, region feature and spatial feature. (b-d) illustrates three types of relation graph attention networks, where the nodes are corresponding to visual objects shown in (a) and the edges are relations. Implicit relations imply a complete graph while semantic relations and spatial relations are not. For semantic relations, it means no predicate relations between two visual objects; for spatial relations, it means the two visual objects are too distant to form any spatial relations.

Graph Initialization

For each visual object o_i , we concatenate its feature \mathbf{v}_i with the output of HADE, namely $\hat{\mathbf{h}}$, to obtain the object's initial representation $\hat{\mathbf{o}}_i$ in each graph. We anticipate that textual information will be integrated into visual scene graphs via this method. Specifically, this process can be represented as follows:

$$\hat{\mathbf{o}}_i = \hat{\mathbf{h}} \oplus \mathbf{v}_i, \quad (8)$$

where \oplus means a concatenation operation.

Semantic Relation

We use τ_e^{sem} to denote the semantic relation between object o_i and object o_j . For instance, in Fig. 2(c), the directed relation between *player* and *racket* is *hold*. The task to extract such directional relation can be formulated as a classification task [10, 16] and we implement a multi-label classifier to note semantic relations between image regions with totally 15 labels including a special label “*none*”. With the relation information, we can first build a semantic relation

graph $G^{sem} = (\mathcal{O}, \mathcal{E}^{sem})$, and then obtain the representation \mathbf{g}_i^{sem} of each node o_i using RGAT(\cdot) function shown in Eq. (3),

$$\mathbf{g}_i^{sem} = \text{RGAT}(\hat{\mathbf{o}}_i, \{(\hat{\mathbf{o}}_j, \tau_e^{sem}) | o_j \in \mathcal{N}_{sem}^+(o_i)\}), \quad (9)$$

where $\mathcal{N}_{sem}^+(o_i)$ is the set including o_i itself and all its direct neighbors in G^{sem} .

Spatial Relation

Spatial relation τ_e^{spa} is a type of relation which indicates the relative position, distance, intersection and some other spatial features of two objects. As shown in Fig. 2(d), the directed spatial relation between *player* and *ball* is *left*, because the ball is in left of the player in Fig. 2(a). Similarly, with the spatial relation information, we can first build a spatial relation graph $G^{spa} = (\mathcal{O}, \mathcal{E}^{spa})$, and then obtain the representation \mathbf{g}_i^{spa} of each node o_i ,

$$\mathbf{g}_i^{spa} = \text{RGAT}(\hat{\mathbf{o}}_i, \{(\hat{\mathbf{o}}_j, \tau_e^{spa}) | o_j \in \mathcal{N}_{spa}^+(o_i)\}), \quad (10)$$

where $\mathcal{N}_{spa}^+(o_i)$ is the set including o_i itself and all neighbors in G^{spa} . It's notable that the parameters of RGAT(\cdot) in Eq. (10) is different from that in Eq. (9).

Implicit Relation

To model relations between nodes that are not directly connected, we build a third graph $G^{imp} = (\mathcal{O}, \mathcal{E}^{imp})$. In this graph, we assume that there is an implicit relation τ_e^{imp} between every pair of nodes o_i and o_j , i.e., G^{imp} is a fully connected graph without labels for edges. Using GAT(\cdot) function shown in Eq. (4), we can obtain the representation \mathbf{g}_i^{imp} of each node o_i ,

$$\mathbf{g}_i^{imp} = \text{GAT}(\hat{\mathbf{o}}_i, \{\hat{\mathbf{o}}_j | o_j \in \mathcal{O}\}). \quad (11)$$

With three representations \mathbf{g}_i^{sem} , \mathbf{g}_i^{spa} and \mathbf{g}_i^{imp} , we can obtain the representation \mathbf{g}_i of each object o_i as follows,

$$\mathbf{g}_i = \mathbf{W} \left(\mathbf{g}_i^{imp} \oplus \mathbf{g}_i^{spa} \oplus \mathbf{g}_i^{sem} \right), \quad (12)$$

where \mathbf{W} is a projection matrix. The representation \mathbf{g}_i contains not only image information, but also dialogue context and question information. For brevity, we denote above process as the following function,

$$\mathbf{g}_i = \text{RAIE} \left(\hat{\mathbf{h}}_c, \{\mathbf{v}_j | o_j \in \mathcal{O}\}, G^{sem}, G^{spa}, G^{imp} \right). \quad (13)$$

Recurrent Attention Manager

Recurrent Attention Manager (RAM) maintains a global state \mathbf{x}_t and updates it with fused multi-modal information \mathbf{h}_t using LSTM, i.e.,

$$\mathbf{x}_t = \text{LSTMCell}(\mathbf{h}_t, \mathbf{x}_{t-1}). \quad (14)$$

We take the pooled textual representation \mathbf{h} as the initial state \mathbf{x}_0 . The fused multi-modal information \mathbf{h}_t is calculated with three steps. The first step is to obtain useful information from dialogue context \mathbf{D} ,

$$\hat{\mathbf{h}} = \text{HADE}(\mathbf{x}_{t-1}, \mathbf{D}), \quad (15)$$

the second step is to fuse text information $\hat{\mathbf{h}}$ with image information using relation-aware image encoder RAIE(\cdot), i.e.,

$$\hat{\mathbf{v}} = \text{RAIE}\left(\hat{\mathbf{h}}, \{\mathbf{v}_j | o_j \in \mathcal{O}\}, G^{sem}, G^{spa}, G^{imp}\right). \quad (16)$$

And finally the third step is the Multi-modal factorized bilinear pooling (**MFP**) function [18]:

$$\text{MFP}(\mathbf{a}, \mathbf{b}) = \text{SumPooling}(\mathbf{U}^T \mathbf{a} \oplus \mathbf{V}^T \mathbf{b}, k), \quad (17)$$

$$\mathbf{h}_t = \text{MFP}(\hat{\mathbf{h}}, \hat{\mathbf{v}}), \quad (18)$$

where the operation $\text{SumPooling}(\mathbf{x}, k)$ means using a one-dimensional non-overlapped window with the size k to perform sum pooling over \mathbf{x} . MFP is adopted because of its well performance in the VQA task [1, 18].

At each iteration, the global state contains more information and makes the history encoding process more specific, thus the module can perform reasoning with much longer dependency. The final output will be \mathbf{h}_T , T is a hyper-parameter that controls total runs of context updating.

4.3 Answer Ranker

Answer Cluster. Traditional methods calculate the similarity between a certain answer candidate and the output of contextual representation encoder directly. We attempt to first perform a clustering operation to get several clustering centers instead. We anticipate that each clustering center represents a group of semantically-similar answers so that these answers can be ranked with higher scores or lower scores simultaneously. In this way, the model can gain a better score on NDCG, which is indicated in the Experiments Sect. 5.3.

In details, we use a conventional clustering algorithm Gaussian Mixture Model (**GMM**) to achieve our goal. Given an answer candidate, GMM can output its corresponding clustering center, which has the same dimension and shape as the answer candidate. Specifically, we denote above process as the following function,

$$\hat{\mathbf{a}}_i = \text{GMM}(\mathbf{a}_i), \quad (19)$$

where the $\hat{\mathbf{a}}_i$ represents the clustering center of \mathbf{a}_i and \mathbf{a}_i is the pooling feature of the answer candidate from the sequence feature \mathbf{A} .

Scores Ranker. Given the candidate answer list and the corresponding clustering centers, and final contextual representation \mathbf{h}_T , answer ranker computes the probability of choosing each answer by:

$$\mathbf{p} = \text{softmax}(\mathbf{E}_a \mathbf{W}_e \mathbf{h}_T), \quad (20)$$

where \mathbf{W}_e is a learnable parameter and the i th row of \mathbf{E}_a , \mathbf{e}_i , is the representation of i th answer candidate, which is calculated as follows,

$$\mathbf{e}_i = \hat{\mathbf{a}}_i + \mathbf{a}_i. \quad (21)$$

Here \mathbf{p} represents a probability distribution over candidate answers. Finally, they are ranked in descending order by these probability values.

5 Experiment

Table 1. Retrieval performance on two major metrics on dataset VisDial v1.0 - Mean Rank of human Response (MRR), Normalized Discounted Cumulative Gain (NDCG), and other metrics. The higher the better for NDCG, MRR and R@k, while the lower the better for Mean rank. * indicates the result is implemented by ourselves.

Name	Dense	Cluster	MRR↑	NDCG↑	R@1↑	R@5↑	R@10↑	Mean↓
Attention-based Methods								
DAN [7]			63.20	57.59	49.63	79.75	89.35	4.30
DAN*		✓	61.37	60.21	47.23	77.79	85.92	4.77
Multi-hop reasoning methods								
ReDAN [5]			53.74	64.47	42.45	64.68	75.68	6.64
Graph-based methods								
DualVD [6]			63.23	56.32	49.25	80.23	89.70	4.11
FGA [15]			63.70	52.10	49.58	80.97	88.55	4.51
Pretrained models								
ViLBERT [13]			67.50	63.87	53.85	84.68	93.25	3.32
ViLBERT [13]	✓		50.74	74.47	37.95	64.13	80.00	6.28
OSCAR* [11]			67.37	66.21	50.22	84.09	92.13	3.42
OSCAR* [11]	✓	✓	55.23	71.94	44.09	73.21	81.34	6.03
(ours)								
R2N + ViLBERT			67.92	64.77	53.91	84.77	93.38	3.30
R2N + ViLBERT	✓	✓	54.29	74.63	43.13	73.86	80.03	5.97
R2N + OSCAR				68.01	66.38	53.21	84.36	93.41
R2N + OSCAR	✓	✓	55.29	72.82	43.93	73.00	80.96	6.05

Table 2. Ablation Study on RGAT. The evaluation metric is MRR. Results are from the model R2N + ViLBERT

w/o RGAT	Implicit only	Semantic only	Spatial only	All
67.54	67.63	67.82	67.71	67.92

5.1 VisDial Dataset and Metrics

We evaluate our proposed model on the VisDial v1.0 dataset [4]. For each question, the dialog agent is given 100 candidate answers. Four kinds of evaluation metrics are used for retrieval performance: (1) MRR (Mean Rank of human Response), (2) R@k (the existence of human response in top-k ranked), (3) NDCG (Normalized Discounted Cumulative Gain) [3,4], (4) Mean, mean rank of all cases.

Previous work [14] found that the dense annotations in VisDial can lead to better performance on NDCG metrics but may hurt MRRs, which highlights

Table 3. Ablation Study on the number of multi-hops on R2N + ViLBERT

No. hops	1	2	3	4
MRR	67.59	67.83	67.92	67.97
Time sec./iter	8.34	11.23	14.09	16.94

a trade-off between them. It is due to dense annotations not correlating well with ground-truth answers to questions. Based on this, we firstly train R2N on the train split of VisDial, and then fine-tune it on dense annotations. Therefore, models that can perform better on either MRR or NDCG are obtained respectively.

5.2 Implementation Details

Relation Aware Image Encoder. We use regions of interest (ROI) extracted from VinVL [19] for OSCAR. Each image consists of 1-100 ROI(s), and every pairs of ROIs are connected with directional relations as aforementioned. These relations are extracted from a relation encoder, which is a simple classifier whose inputs are features from two certain ROIs and output is a relation label.

Graph Attention Networks. In HADE module, we use multi-head attention with 8 heads and 2 layers in GAT, and the hidden size is 512. In RAIE module, multi-head attention with 4 heads and 2 layers is used, in RGAT.

Training Details. We minimize the cross-entropy loss in training and use Adam [8]. Due to the restriction of GPU memory, we can not train R2N on the whole candidate list with a size of 100, and hence we sample one positive example and two negative examples in every turns. However, we have to fit the GMM with a full list of answer candidates. Depending on this, every several rounds, we cancel the sampling mechanism and input all 100 answers for training.

5.3 Quantitative Results

Baselines. We compare our proposed approach with several models shown in Table 1. Among them, DAN executes the single-hop reasoning with cross modality, simply using cross-modal representation as the context vector. ReDAN utilizes dual attention mechanism as well. It is a similar multi-hop reasoning model to ours, but it neglects relations between visual objects. Moreover, two pretrained models, i.e. ViLBERT and OSCAR are compared as well because our method is based on them. We also tried other Visual-Linguistic Pretrained models but they cannot achieve better results so we didn't list them above.

Main Results. R2N with ViLBERT outperforms all other approaches on NDCG with the value of 74.63, in which dense annotations and clustering mechanism are used for fine-tuning. Higher NDCG means more semantically-correct

answers but not only the ground truth answer are ranked higher. Without the two tricks above, R2N with OSCAR can achieve the best result on MRR, 68.01, which indicates that the ground truth answer is ranked higher on average. In the two cases, our model also achieves comparable results on other metrics. We also collect results of original OSCAR [11], so that we can compare the results between original OSCAR and R2N + OSCAR and find that R2N can improve OSCAR’s result. In addition, we analyse the effectiveness of answer candidate cluster method on DAN and as shown it improves the result on NDCG.

Results with Different Types of Relations. To validate the effectiveness of relation-aware reasoning, we equip R2N with RGAT of varied relations. To facilitate comparison, we test the model without RGAT as the baseline temporarily in this part, accompanying with each single type of relations adopted individually for other three models. Model with all the three types of relations is evaluated for comparison as well.

Results in Table 2 indicate that all types of relations take effect depending on the MRRs. Especially, the semantic relation has the greatest influence among them, since the RGAT with semantic relation has fewer edges. This is because predicate relations are sparser in comparison with spatial relations and implicit relations and therefore it has more information according to the entropy theory. Furthermore, the model with all three types of relations achieves the best MRR with 0.38 improved, which implies different types of relations have a combined effect on feature extraction of visual objects.

Results with Different Numbers of Hop. Table 3 demonstrates the influence of the number of multi-hops on the performance. It can be seen that an increase in the number of multi-hops can improve the performance on MRR. Although increasing the number can improve the performance of the model, it also significantly increases the training time and memory footprint, so we only take the model with a hop count of 3 as the final model. Model with over 4 hops will meet the out-of-memory error on our servers.

6 Conclusion

We introduce R2N for visual dialog task. R2N executes multi-hop reasoning to resolve visual co-reference and ground textual information on image features. It also takes relation-aware visual information into account. The experimental results show that R2N outperforms various baselines in this task.

Acknowledgements. This work has been supported by Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), State Key Laboratory of Media Convergence Production Technology and Systems Project (No. SKLMCPTS2020003) and Startup Fund for Youngman Research at SJTU (SFYR at SJTU).

References

1. Antol, S., et al.: VQA: visual question answering. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2425–2433 (2015)

2. Chen, F., Meng, F., Xu, J., Li, P., Xu, B., Zhou, J.: DMRM: a dual-channel multi-hop reasoning model for visual dialog. In: Proceedings of the AAAI, vol. 34, pp. 7504–7511 (2020)
3. Clarke, C.L., et al.: Novelty and diversity in information retrieval evaluation. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 659–666 (2008)
4. Das, A., et al.: Visual dialog. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 326–335 (2017)
5. Gan, Z., Cheng, Y., Kholy, A.E., Li, L., Liu, J., Gao, J.: Multi-step reasoning via recurrent dual attention for visual dialog. arXiv preprint [arXiv:1902.00579](https://arxiv.org/abs/1902.00579) (2019)
6. Jiang, X., et al.: Dualvd: an adaptive dual encoding model for deep visual understanding in visual dialogue. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 11125–11132 (2020)
7. Kang, G.C., Lim, J., Zhang, B.T.: Dual attention networks for visual reference resolution in visual dialog. arXiv preprint [arXiv:1902.09368](https://arxiv.org/abs/1902.09368) (2019)
8. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
9. Kottur, S., Moura, J.M., Parikh, D., Batra, D., Rohrbach, M.: Visual coreference resolution in visual dialog using neural module networks. In: Proceedings of ECCV, pp. 153–169 (2018)
10. Li, L., Gan, Z., Cheng, Y., Liu, J.: Relation-aware graph attention network for visual question answering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10313–10322 (2019)
11. Li, X., et al.: OSCAR: object-semantics aligned pre-training for vision-language tasks. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12375, pp. 121–137. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58577-8_8
12. Lu, C., Krishna, R., Bernstein, M., Fei-Fei, L.: Visual relationship detection with language priors. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 852–869. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_51
13. Lu, J., Batra, D., Parikh, D., Lee, S.: Vilbert: pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. arXiv preprint [arXiv:1908.02265](https://arxiv.org/abs/1908.02265) (2019)
14. Murahari, V., Batra, D., Parikh, D., Das, A.: Large-scale pretraining for visual dialog: a simple state-of-the-art baseline. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12363, pp. 336–352. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58523-5_20
15. Schwartz, I., Yu, S., Hazan, T., Schwing, A.G.: Factor graph attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2039–2048 (2019)
16. Yao, T., Pan, Y., Li, Y., Mei, T.: Exploring visual relationship for image captioning. In: Proceedings of ECCV, pp. 684–699 (2018)
17. You, Q., Jin, H., Wang, Z., Fang, C., Luo, J.: Image captioning with semantic attention. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4651–4659 (2016)
18. Yu, Z., Yu, J., Fan, J., Tao, D.: Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1821–1830 (2017)
19. Zhang, P., et al.: Vinvl: making visual representations matter in vision-language models. arXiv preprint [arXiv:2101.00529](https://arxiv.org/abs/2101.00529) (2021)



Multi-modal Sarcasm Detection Based on Contrastive Attention Mechanism

Xiaoqiang Zhang, Ying Chen^(✉), and Guangyuan Li

China Agricultural University, Beijing, China

{xqzhang, chenying, liguangy}@cau.edu.cn

Abstract. In the past decade, sarcasm detection has been intensively conducted in a textual scenario. With the popularization of video communication, the analysis in multi-modal scenarios has received much attention in recent years. Therefore, multi-modal sarcasm detection, which aims at detecting sarcasm in video conversations, becomes increasingly hot in both the natural language processing community and the multi-modal analysis community. In this paper, considering that sarcasm is often conveyed through incongruity between modalities (e.g., text expressing a compliment while acoustic tone indicating a grumble), we construct a Contrastive-Attention-based Sarcasm Detection (ConAttSD) model, which uses an inter-modality contrastive attention mechanism to extract several contrastive features for an utterance. A contrastive feature represents the incongruity of information between two modalities. Our experiments on MUStARD, a benchmark multi-modal sarcasm dataset, demonstrate the effectiveness of the proposed ConAttSD model.

Keywords: Sarcasm detection · Multi-modal analysis · Contrastive attention

1 Introduction

Sarcasm is a form of communication in which the speaker intends to communicate a contradictory situation or the opposite meaning of what is literally said. Understanding sarcasm often uses a highly complex structure of multi-modal signals [1]. For example, we employ three communicative modalities in a coordinated manner to convey our intentions: language (spoken words), vision (gestures), and audio (voice). Therefore, it is important to do multi-modal sarcasm detection that recognizes sarcasm in videos, where the three modalities are present.

According to the studies of multi-modal affective analysis (i.e., sentiment detection, emotion detection), there are dual dynamics in human communication: intra-modality dynamics and inter-modality dynamics [2]. Intra-modality dynamics refer to dynamics within each modality, and inter-modality dynamics refer to dynamics between modalities. Sarcasm detection needs the dual dynamics to find incongruity which is either from multiple modalities in an utterance or from the context of an utterance [1]. For example, there are two sarcasm cases in Fig. 1. The sarcasm in Fig. 1(a) is conveyed by an inter-modality incongruity, where the text indicates a compliment while the facial

expressions show a disagreement. The sarcasm in Fig. 1(b) is expressed by a textual incongruity, where the sentence including “ugly” and “itchy” is indicative of negative sentiment, while the sentence including “favorite” gives a positive sentiment.

Although several studies [1, 3] have investigated multi-modal sarcasm detection, there are remaining several unsolved research problems. One crucially problem lies in the modeling of the incongruity between modalities in an utterance. Inter-modality incongruity often plays an important role in sarcasm detection, e.g., the sarcasm in Fig. 1(a) is noticed through the contrast between the language modality and the visual modality. However, to our best knowledge, the feature extraction of the inter-modality incongruity has not yet been explored in multi-modal analysis. Therefore, more efforts are required for multi-modal sarcasm detection.

Utterance:

(a) Inter-modal incongruity

SHELDON :

I'm listening to you snore. I'm wondering how I'll ever sleep without it.

- **Text:** suggests neutral
- **Audio:** grumbled tone
- **Video:** gloomy face



(b) Intra-modal incongruity

LEONARD :

Why? Because I got an **ugly, itchy** sweater, and my brother got a car?
No, I was her **favorite**.



Fig. 1. Incongruity in sarcasm

To address the issue, we propose a Contrastive-Attention-based Sarcasm Detection (ConAttSD) model, which utilizes a contrastive attention mechanism [4, 5] to extract inter-modality incongruent information for multi-modal sarcasm detection. As shown in Fig. 1(a), in the feature space of the focused utterance, the features learned from the vision and audio should be similar, whereas the features learned from the text and vision should be different. Thus, we design an inter-modality contrastive attention mechanism, which produces opponent attention weights for a directed bi-modal variant (e.g., *text* → *audio*), and then generate a contrastive feature to represent the incongruity between the two modalities.

Overall, our main contributions can be summarized as follows:

- We design a Contrastive-Attention-based Sarcasm Detection (ConAttSD) model to detect sarcasm in video conversations.

- We propose an inter-modality contrastive attention mechanism to extract contrastive features to represent the incongruity between modalities. These contrastive features can effectively facilitate the detection of sarcasm.
- Experiments on MUStARD (a benchmark multi-modal sarcasm dataset) demonstrate the effectiveness of our ConAttSD model for multi-modal sarcasm detection.

2 Related Work

2.1 Sarcasm Detection

The studies of sarcasm detection in the textual scenario have been intensively carried out for many years. In general, text-based sarcasm detection can be divided into rule-based, statistical-machine-learning-based, and deep-learning-based.

Rule-based sarcasm detection [6, 7] mainly aims to detection sentiment polarity inconsistency using different rules. Sentiment polarity inconsistency refers to two sentiments that are contradictory in their polarity (i.e., negative vs. positive). For example, the sarcasm in Fig. 1(b) is expressed by sentiment polarity inconsistency. Sarcasm detection based on statistical machine learning mainly focused on feature extraction. In general, features are extracted from two perspectives [8]: the characteristics of sarcasm expressions in different-level texts (i.e., special symbols, morphology, syntax) and sentiment polarity inconsistency. In recent years, sarcasm detection based on deep learning has been explored, which uses different deep neural networks (DNNs) to extract various types of textual information. [9] used a bidirectional recurrent neural network to extract the representation of contextual information. [10] utilized multiple pre-trained models (involving emotion, sentiment, personality, etc.) to help feature extraction. [11] used two complementary adversarial learning methods to improve sarcasm detection. [12] pre-trained the BERT model to extract representations with more emotional information to help sarcasm detection.

Recently, there are concerns about sarcasm detection in multi-modal scenarios. [1] provided MUStARD, a sarcasm dataset on video conversations, and [3] assigned affective labels (sentiment and emotion) to each utterance in MUStARD. Moreover, [3] treated an utterance and its historical context as a whole by concatenating operations and then used two attention mechanisms (i.e., inter-segment inter-modal attention and intra-segment inter-modal attention) to model inter-modality dynamics. In this paper, according to the characteristics of sarcasm expressions in videos, we focus on the extracting of incongruent information between modalities in an utterance.

2.2 Multi-modal Affective Analysis

Multi-modal affective analysis is an important task in the multi-modal analysis community, which generalizes text-based sentiment detection or emotion detection to videos. The challenge in the multi-modal affective analysis is how to effectively model inter-modality dynamics. In general, the inter-modality modeling methods are designed for two scenarios: monologue and dialogue.

In a monologue scenario, the modeling of multi-modal interactive information mostly focuses on the exaction of sequential context information along the time axis. [2]

proposed a tensor fusion network that can capture the interactive information of any modal combination. Subsequently, the network was strengthened so that the multi-modal sequence information that changes over time can be dynamically obtained [13, 14]. [15] proposed a sequence-to-sequence translation model to extract multi-modal interaction information during the translation from one modality to another modality. [16] used Transformers [17] to model each modality and used its multi-head attention mechanism to capture multi-modal interaction information.

In a dialogue scenario, the affective state of a speaker is the result of the interaction of multiple factors (e.g., contextual information, previous affective state, the expression styles of the speaker). Therefore, in addition to the sequential context information, [18] used memory networks to separately model the contextual information of different speakers in dialogue. [19] used the GRU network [20] to separately model the emotions of different speakers. [21, 22] used graph convolutional networks to simultaneously model the emotions of different speakers. [23] proposed two-layer Transformers, which uses Transformers to extract and modulate intra-modality information. In this paper, besides the modeling of sequential contexts and speakers in a dialogue, we mainly deal with inter-modality incongruity in an utterance, which is a specific research issue for multi-modal sarcasm.

3 Methodology

3.1 Overview

In this section, we describe our proposed ConAttSD model for multi-modal sarcasm detection. Suppose that a conversation has proceeded for t turns so far with the utterance sequence $S_i = \{u_1, u_2, \dots, u_i\}$, the i -th utterance u_i is ready to be tested, and the other utterances are its historical context. The goal of our multi-modal sarcasm detection is to assign a binary label (1: sarcasm; 0: no sarcasm) to u_i conditioned on u_i and its historical context.

Specifically, as illustrated in Fig. 2, our ConAttSD model comprises two encoders (sequential context encoder and contrastive-attention-based encoder) and one decoder. The sequential context encoder dynamically captures intra-modality influence transmitted along with the conversation, and the contrastive-attention-based encoder extracts incongruent information between modalities in u_i by an inter-modality contrastive attention mechanism. Then, a linear decoder assigns a sarcasm label to u_i according to its representation.

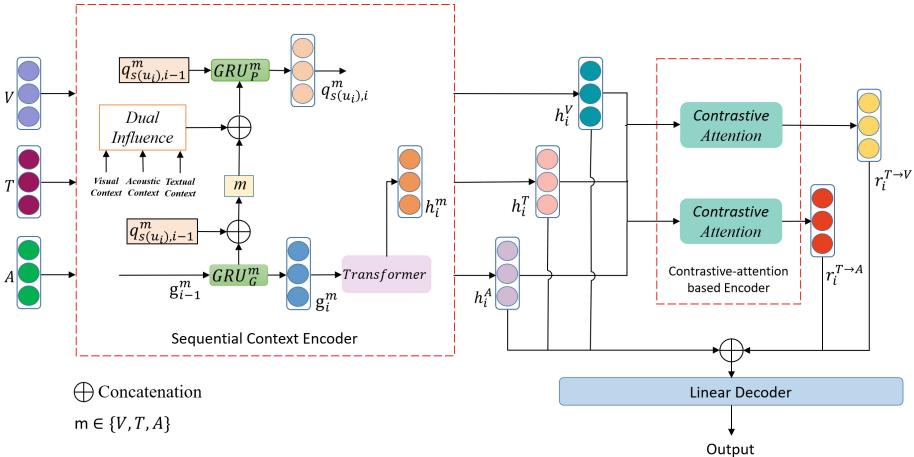


Fig. 2. The overview of our multi-modal sarcasm detection

3.2 Utterance Representation

Each utterance is represented by three vectors: a textual feature T for linguistic content (text), an acoustic feature A for acoustic characteristics (audio), and a visual feature V for visual information (vision).

Textual Feature Extraction

Textual feature T is generated by a pre-trained BERT model [24] and the dimension is 768 ($d_t = 768$).

Acoustic Feature Extraction

The library Librosa [25] is used to extract acoustic features, including Mel-frequency cepstral coefficients (MFCCs), pitch tracking and voiced/unvoiced segmenting features, peak slope parameters, and maxima dispersion quotients. The average of the acoustic features for the focused utterance is used as acoustic feature A , and the dimension is 298 ($d_a = 298$).

Visual Feature Extraction

A pre-trained Resnet-152 [26] is used to extract a visual feature for a visual frame. The average of visual features of all frames in the focused utterance is used as visual feature V , and the dimension is 2048 ($d_v = 2048$).

3.3 Sequential Context Encoder

Since conversations are highly sequential in nature and contextual information flows along a sequence of utterances, we propose a sequential context encoder to model the inter-modality influence between these utterances. As shown in Fig. 2, there are two sub-encoders used in the sequential context encoder: a GRU-based encoder which uses

GRU [20] to extract sequential context information, and a Transformer-based encoder which applies Transformers [17] to the vector output from the GRU-based encoder.

In the GRU-based encoder, we adopt the inter-modal influence modeling proposed by [27] for multi-modal sentiment detection. For utterance u_i , global state g_i^m and speaker state $q_{s(u_i),i}^m$ interact together to represent both utterance u_i and speaker $s(u_i)$ (i.e., the speaker of the utterance u_i), respectively. Notice that a global state g_i^m is actually a kind of sequential context information for utterance u_i . Formally, both the global state and the speaker state are iteratively updated as follows.

$$g_i^m = GRU_G^m \left((u_i^m \oplus q_{s(u_i),i-1}^m), g_{i-1}^m \right) \quad (1)$$

$$q_{s(u_i),i}^m = GRU_P^m \left((u_i^m \oplus c_i^m), q_{s(u_i),i-1}^m \right) \quad (2)$$

where $m \in \{T, A, V\}$ is the modality, g_i^m denotes the global state representation of i -th turn utterance for the modality m , $q_{s(u_i),i}^m$ denotes the speaker state representation at i -th turn utterance for the modality m , c_i^m is the context representation of i -th utterance using a dual influence network, which includes both intra-modal and inter-modal information.

In the Transformer-based encoder, we extract more effective sequential context information using Transformers, which have shown superior performance in capturing long-range dependency than RNN models. A Transformer is composed of a stack of B identical blocks, and each block has two sub-layers (including a multi-head self-attention mechanism and a Multi-Layer Perceptron) with a residual connection, as shown in Eq. 3. In this paper, we use a Transformer to capture the dependency inside the global state g_i^m ($m \in \{T, A, V\}$) and output a sequential context vector h_i^m for utterance u_i , as illuminated in Fig. 2.

$$y = \text{LayerNorm}(x + \text{Sublayer}(x)) \quad (3)$$

3.4 Contrastive-Attention-Based Encoder

To extract the incongruent information between multiple modalities for sarcasm detection, we propose an inter-modality contrastive attention mechanism that applies the contrastive attention mechanism to the three sequential context vectors (h_i^T , h_i^A and h_i^V) outputted from the sequential context encoder.

Contrastive Attention

The contrastive attention mechanism, which attempts to capture irrelevant or less relevant parts between two vectors, was proposed by [4] for person re-identification (a computer vision task), and then used for text summarization (an NLP task) by [5]. In fact, the contrastive attention mechanism is transformed from the self-attention mechanism used in Transformers.

Specifically, given three input vectors Q, K and V, the self-attention mechanism is defined by Eq. 4, and the contrastive attention mechanism is defined by Eq. 5–7. First, the attention weights a_c is calculated by Eq. 5. Then, the opponent attention weights a_o is obtained through the opponent function applied on a_c followed by the softmax function,

as shown in Eq. 6. Compared to the conventional attention weights a_c which capture the most relevant part between Q and K , the opponent attention weights a_o focuses on their irrelevant parts. Lastly, a contrastive vector is generated by Eq. 7, which is the weighted sum of elements of V and the weights are the opponent attention weights.

$$y = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (4)$$

$$a_c = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \quad (5)$$

$$a_o = \text{softmax}(1 - a_c) \quad (6)$$

$$r = a_o V \quad (7)$$

where Q, K, V are queries, keys and values, respectively, and d_k is the dimension of K .

Inter-modality Contrastive Attention

In an inter-modality contrastive attention, a directed bi-modal variant (e.g., $T \rightarrow A$: $Q = h_i^T$ and $K = h_i^A$) is used as its inputs, and an inter-modality contrastive vector (e.g., $r_i^{T \rightarrow A}$) is generated for utterance u_i . First, opponent attention weights are learned by Eq. 6. Then, the opponent attention weights are applied to a modality (e.g., $V = h_i^A$) to produce the inter-modality contrastive vector by Eq. 7.

As shown in Fig. 2, our contrastive-attention-based encoder takes the textual modality as the anchor modality, and generates two directed bi-modal variants (i.e., $T \rightarrow A$ and $T \rightarrow V$) as input for two inter-modality contrastive attentions, respectively. Then, two inter-modal contrastive vectors (i.e., $r_i^{T \rightarrow A}$ and $r_i^{T \rightarrow V}$) are generated, where an inter-modal contrastive vector (e.g., $r_i^{T \rightarrow A}$) represents the incongruity in its corresponding input bi-modal variant (e.g., $T \rightarrow A$). Thus, through the two inter-modality contrastive attentions, text can be effectively contrasted with information from audio and vision, respectively.

3.5 Linear Decoder

As illuminated in Fig. 2, the three sequential context vectors from the sequential context encoder and the two inter-modal contrastive vectors from the contrastive-attention-based encoder are concatenated as Eq. 8. Then, the final vector β_i is feed to a softmax classifier to obtain the sarcasm label of utterance u_i .

$$\beta_i = [h_i^T, h_i^A, h_i^V, r_i^{T \rightarrow A}, r_i^{T \rightarrow V}] \quad (8)$$

4 Experiment

4.1 Setup

Datasets

In our experiment, we use a benchmark Multimodal Sarcasm Detection Dataset (MUSTARD) provided by [1] for multi-modal sarcasm detection. The dataset was collected from 4 popular TV Series: Friends, the Big Bang Theory, the Golden Girls, and Sarcasmaholics Anonymous. There are totally 690 samples (i.e., conversations) with an even number of sarcastic and non-sarcastic samples, and the utterances in each sample consist of three modalities: vision, audio, and text.

Moreover, there are two experimental setups to use MUStARD for the sarcasm detection [1]: speaker-dependent and speaker-independent. Compared to the speaker-dependent scenario, the speaker-independent setup is more challenging because it prevents the detection model using registered speaker’s specific information and requires the model with a higher degree of generalization. Since it is often a case that a unregistered speaker appears in a video conversation, we work on the sarcasm detection with the speaker-independent setup in this paper. Specifically, the speaker-independent dataset split is used in our experiment, where videos from The Big Bang Theory, The Golden Girls, and Sarcasmaholics Anonymous are served as the training set and videos from Friends are used as the testing set. Moreover, the performances are evaluated by three metrics: precision (P), recall (R), and F1-score (F1).

Baselines

We compare our proposed approach with the following two models using different encoders.

- Two-attention-based encoder: This is a multi-task learning system developed by [3] to detect simultaneously detect sarcasm, sentiments, and emotions. For each modality, the focused utterance and its historical context utterances are concatenated as an input utterance. Then, two attention mechanisms (i.e., inter-segment inter-modal attention and intra-segment inter-modal Attention) are applied to model inter-modality dynamics for the input utterance. Lastly, a multi-task learning framework is used based on inter-modality interactive information.
- GRU-based encoder: This is a pure sarcasm detection system, which concatenates three global states from our GRU-based encoder (i.e., g_i^T , g_i^A and g_i^V) to detect sarcasm in the focused utterance.

Implementation Details

While training, we use the Adam optimizer [28] to update all hyper-parameters. Each training batch contains 64 conversations, and the learning rate is set to 0.0001. For GRU, to reduce over-fitting, dropout [29] is applied, and it is set 0.5. The size of the hidden representations is 150. For Transformers, the number of blocks B is set to 3, and the number of heads is 6.

4.2 Results and Analysis

Model Comparison

We first compare our ConAttSD model with the two baseline models, and list the performances in Table 1.

Table 1. Sarcasm detection results of different models.

	P	R	F1
Two-attention-based encoder [3]	71.51	71.35	70.46
GRU-based encoder	71.69	70.90	70.82
Sequential Context Encoder (GRU + Transformer)	72.32	72.32	72.26
ConAttSD (GRU + Transformer + Contrastive Attention)	74.46	74.01	73.97

Table 2. Sarcasm detection results of the sequential context encoder using different modalities.

		P	R	F1
Uni-modal	T	53.38	53.39	53.39
	A	67.99	67.23	66.52
	V	71.42	71.19	71.19
Multi-modal	T + A	60.38	60.45	60.35
	T + V	70.42	70.34	70.35
	A + V	73.17	72.03	71.89
	T + A + V	72.32	72.32	72.26

Table 3. Sarcasm detection results of the contrastive-attention-based encoder using different bi-modal variants.

	P	R	F1
$T \rightarrow A$	71.24	71.19	71.07
$A \rightarrow T$	71.51	70.90	70.85
$T \rightarrow V$	70.29	70.06	70.06
$V \rightarrow T$	72.19	72.97	72.58
$A \rightarrow V$	70.66	70.34	70.33
$V \rightarrow A$	70.91	70.62	70.62
Optimal: $T \rightarrow A + T \rightarrow V$	74.46	74.01	73.97

In Table 1, ConAttSD significantly outperforms the two baseline systems. E.g., compared to the best baseline system (i.e., the GRU-based encoder), the F1 score of ConAttSD rises by 3.51%. Specifically, the F1 score rises by 1.44% (from 70.82% to 72.26%) through incorporating the Transformer-based encoder and furthermore increases 1.71% (from 72.26% to 73.97%) by adding the contrastive-attention-based encoder. This indicates that our Transformer-based encoder and contrastive-attention-based encoder can effectively extract sequential context information and the inter-modality incongruent information.

Moreover, from Table 1, we observe that although the baseline model, the two-attention-based encoder, adopts a multi-task learning framework to use richer label information (sentiment, emotion, and sarcasm), its performance is still not comparable to the one of ConAttSD whose input is only sarcasm labels. This indicates that how to effectively combine affective information and sarcasm for multi-modal analysis needs more investigation.

Modality Comparison

To further explore the effects of the three modalities for our ConAttSD model, we perform an in-depth analysis of the sequential context encoder and the contrastive-attention-based encoder with different modalities, respectively.

First, we evaluate our sequential context encoder with all possible inputs: uni-modal variants (i.e., T , A , and V), bi-modal variants (i.e., $T + A$, $T + V$, and $A + V$), and a tri-modal variant (i.e., $T + A + V$), the performances are shown in Table 2. In Table 2, the model with the visual modality achieves the best performance among the unimodal variants. Furthermore, the addition of acoustic modality (i.e., $A + V$) slightly improves the uni-modal baseline (from 71.19% to 71.89% in F1 scores). Finally, the tri-modal variant achieves the best performance (72.26% in F1 scores).

Then, based on the optimal sequential context encoder whose input is $T + A + V$, we evaluate our contrastive-attention-based encoder with all possible inputs. Notice that the inter-modality contrastive attention requires exactly two modalities, and any directed bi-modal variant (e.g., $A \rightarrow T$, $A \rightarrow V$) can serve as input to the inter-modality contrastive attention. The performances are listed in Table 3. In Table 3, the model input with either $V \rightarrow T$ or $T \rightarrow A$ achieves good performances among these directed bi-modal variants. E.g., the F1 score is 72.58% for $V \rightarrow T$ and 71.07% for $T \rightarrow A$. This indicates that texts that look seemingly straightforward is noticed to contain sarcasm only when vocal tonality and facial expressions are taken into account. After searching all possible combinations of the directed bi-modal variants, we find that the model with the two directed bi-modal variants (i.e., $T \rightarrow V$ and $T \rightarrow A$) achieves the best performance (73.97% in F1 scores).

5 Conclusion

In this paper, we propose a novel Contrastive-Attention-based Sarcasm Detection (ConAttSD) model for multi-modal sarcasm detection. Experimental results indicate the capability of our ConAttSD in capturing inter-modal incongruent information by inter-modality contrastive attention. In the future, we would like to investigate the combination of sarcasm and affective information for multi-modal analysis.

References

1. Castro, S., Hazarika, D., Pérez-Rosas, V., et al.: Towards multimodal sarcasm detection (an obviously perfect paper). In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4619–4629 (2019)
2. Zadeh, A., Chen, M., Poria, S., et al.: Tensor fusion network for multimodal sentiment analysis. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 1103–1114 (2017)
3. Firdaus, M., Chauhan, H., Ekbal, A., et al.: MEISD: a multimodal multi-label emotion, intensity and sentiment dialogue dataset for emotion recognition and sentiment analysis in conversations. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 4441–4453 (2020)
4. Song, C., Huang, Y., Ouyang, W., et al.: Mask-guided contrastive attention model for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1179–1188 (2018)
5. Duan, X., Yu, H., Yin, M., et al.: Contrastive attention mechanism for abstractive sentence summarization. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pp. 3044–3053 (2019)
6. Riloff, E., Qadir, A., Surve, P., et al.: Sarcasm as contrast between a positive sentiment and negative situation. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 704–714 (2013)
7. Maynard, D.G., Greenwood, M.A.: Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis. In: Proceedings of the 9th Conference on Language Resources and Evaluation, pp. 4238–4243 (2014)
8. Van Hee, C.: Can Machines Sense Irony?: Exploring Automatic Irony Detection on Social Media. Ghent University (2017)
9. Zhang, M., Zhang, Y., Fu, G.: Tweet sarcasm detection using deep neural network. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: technical papers. pp. 2449–2460 (2016)
10. Poria, S., Cambria, E., Hazarika, D., et al.: A deeper look into sarcastic tweets using deep convolutional neural networks. In: Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers, pp. 1601–1612 (2016)
11. Zhang, Q., Du, J., Xu, R.: Sarcasm detection based on adversarial learning. Beijing Da Xue Xue Bao **55**(1), 29–36 (2019)
12. Babanejad, N., Davoudi, H., An, A., et al.: Affective and contextual embedding for sarcasm detection. In: Proceedings of the 28th International Conference on Computational Linguistics. pp. 225–243 (2020)
13. Zadeh, A., Liang, P.P., Mazumder, N., et al.: Memory fusion network for multi-view sequential learning. Proc. AAAI Conf. Artif. Intell. **32**(1) (2018)
14. Zadeh, A.B., Liang, P.P., Poria, S., et al.: Multimodal language analysis in the wild: CMU-mosei dataset and interpretable dynamic fusion graph. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (vol. 1: Long Papers). pp. 2236–2246 (2018)
15. Pham, H., Liang, P.P., Manzini, T., et al.: Found in translation: learning robust joint representations by cyclic translations between modalities. Proc. AAAI Conf. Artif. Intell. **33**(1), 6892–6899 (2019)
16. Tsai, Y.H.H., Bai, S., Liang, P.P., et al.: Multimodal transformer for unaligned multimodal language sequences. In Proceedings of the Conference, Association for Computational Linguistics, Meeting, NIH Public Access, 2019. p. 6558 (2019)

17. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. In: NIPS, pp. 5998–6008 (2017)
18. Hazarika, D., Poria, S., Mihalcea, R., et al.: ICON: interactive conversational memory network for multimodal emotion detection. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2594–2604 (2018)
19. Majumder, N., Poria, S., Hazarika, D., et al.: DialogueRNN: an attentive rnn for emotion detection in conversations. Proc. AAAI Conf. Artif. Intell. **33**(1), 6818–6825 (2019)
20. Chung, J., Gulcehre, C., Cho, K.H., et al.: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. CoRR (2014). <https://arxiv.org/abs/1412.3555>
21. Ghosal, D., Majumder, N., Poria, S., et al.: DialogueGCN: a graph convolutional neural network for emotion recognition in conversation. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pp. 154–164 (2019)
22. Ishiwatari, T., Yasuda, Y., Miyazaki, T., et al.: Relation-aware graph attention networks with relational position encodings for emotion recognition in conversations. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pp. 7360–7370 (2020)
23. Delbrouck, J.B., Tits, N., Dupont, S.: Modulated fusion using transformer for linguistic-acoustic emotion recognition. In: Proceedings of the First International Workshop on Natural Language Processing Beyond Text, 20 November 2020, pp. 1–10 (2020)
24. Devlin, J., Chang, M.W., Lee, K., et al.: BERT: pre-training of deep bidirectional transformers for language understanding (2018)
25. McFee, B., McVicar, M., Balke, S., et al.: librosa/librosa: 0.6.2 (2018)
26. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
27. Zhang, D., Zhang, W., Li, S., et al.: Modeling both intra-and inter-modal influence for real-time emotion detection in conversations. In: Proceedings of ACM Multimedia, pp. 503–511 (2020)
28. Kingma, D.P., Ba, J.: ADAM: a method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR)
29. Srivastava, N., Hinton, G., Krizhevsky, A., et al.: Dropout: a simple way to prevent neural networks from overfitting. In: Journal of Machine Learning Research, pp. 1929–1958 (2014)

Author Index

- Bai, Xingyu II-519
Bai, Xu I-557
Bai, Xuefeng I-104
Bajaj, Prateek II-323
Bao, Hui II-235
Bao, Jianzhu II-472
Bao, Jie II-275
Bao, Junwei I-439, II-124
Bao, Lujia II-345
Bao, Xinrui II-235
Bharti, Taroon I-786
Bing, Lidong II-548
Bo, Lin I-53
Bo, Ruipeng II-496

Cao, Ruisheng I-505
Cao, Shuai II-219
Castle, Steffen II-317
Chao, Rui II-219
Chao, Wenhan II-3
Chen, Boxing I-104
Chen, Chunquan II-345
Chen, Jiahao II-434
Chen, Jiangjie I-353
Chen, Jiaze I-389
Chen, Jingqiang I-116
Chen, Ke-jia I-116
Chen, Liyi I-664
Chen, Lu I-505, I-810
Chen, Nan I-607
Chen, Ruijun II-111
Chen, Shaowei I-664, I-734
Chen, Shuai II-496
Chen, Sichao II-365
Chen, Wei I-401
Chen, Wenliang I-607
Chen, Xiaoliang II-248
Chen, Xiaoshuai II-345
Chen, Xiaoying I-275
Chen, Xinyu I-239
Chen, Yidong I-65, I-652
Chen, Ying I-822
Chen, Yubo I-338
Chen, Yufeng I-92, I-365

Chen, Zhi I-505
Cheng, Biao I-401
Cheng, Liying II-548
Chi, Zewen II-179
Chu, Xiaomin I-15
Cui, Edward I-786
Cui, Zhen II-163

Dai, Anan I-544
Dai, Lu II-59
Dai, Wen II-496
Dai, Xudong I-300, II-509
Dang, Jianwu I-493
Dang, Kai I-734
Dey, Sreya II-323
Di, Hui I-92
Dong, Mengxing I-427
Du, Haowei I-676
Du, Yajun II-248
Duan, Nan I-401, I-786

Fa, Lingling II-509
Fan, Xiaoming I-631
Fan, Yaxin I-15
Feng, Chong II-400
Feng, Lin I-300
Fu, Biao I-65
Fu, Yingwen I-210

Gan, Zhen I-338
Gao, Jing I-196
Gao, Sheng II-43
Gao, Yan I-413
Gao, Zeyu II-235
Geng, Yuanling II-579
Ghosal, Tirthankar II-590
Gong, Yeyun I-401
Gong, Yifei II-354
Gu, Chengmin II-509
Guo, Dongjie II-509
Guo, Weigang II-519

Han, Ruifang I-709
Han, Ting II-206

- Han, Yujiao I-709
Hao, Tianyong II-139
He, Kai II-235
He, Pan II-603
He, Ruidan II-548
He, Ruifang I-493
He, Xiaodong I-439, II-124
He, Xiaofeng I-325
He, Xingchen I-652
He, Yancheng I-583
He, Zheyu II-434
Hong, Yu I-427
Hou, Boyu II-422
Hou, Lei I-27
Hu, Baotian II-365
Hu, Changjian I-365
Hu, Hai II-412
Hu, Mengyang I-53
Hu, Nan I-325
Hu, Renfen II-412
Hu, Xiaoxuan I-196
Hu, Xinshuo II-365
Hua, Xinyu II-496
Huang, Canan I-377
Huang, Chongxuan II-206
Huang, Degen I-129
Huang, Haoyang I-79, I-786
Huang, Heyan I-141, II-179, II-400
Huang, Hui I-92
Huang, Jun II-422
Huang, Kaiyu I-129
Huang, Minlie II-206
Huang, Peijie I-517
Huang, Rongtao I-427
Huang, Weihao I-544
Huang, Xuanjing II-86, II-548
Huang, Yongjie II-151
Inkpen, Diana II-303
Jain, Shipra II-323
Ji, Lei I-786
Jia, Xu I-467
Jia, Yuxiang II-219
Jiang, Changjian II-548
Jiang, Di II-43
Jiang, Feng I-15
Jiang, Lei I-557
Jiang, Shengyi I-210
Jiang, Yufan II-388
Jiang, Zhihua I-570
Jin, Lifeng II-484
Ju, Jiangzhou II-163
Ju, Qi II-519
Kaur, Prabsimran II-590
Kohli, Guneet Singh II-590
Kong, Fang I-40, I-312, I-480
Kong, Jun I-224
Lan, Man I-688, II-377, II-447
Lee, Lap-Kei II-139
Li, Bangqi II-98
Li, Bochao I-455
Li, Changliang I-377
Li, Dongfang II-365
Li, Guangyuan I-822
Li, Jiuyi I-129
Li, Juan I-413
Li, Juanzi I-27
Li, Junyi II-412
Li, Lei I-389
Li, Mu II-388
Li, Peifeng I-15, I-239, I-262, II-73
Li, Peng II-422
Li, Qizhi II-248
Li, Shuqun I-746, II-579
Li, Shuxin I-709
Li, Si II-43
Li, Sujian II-614
Li, Weikang II-345
Li, Xiangyang II-614
Li, Xiaoyong Y. I-251
Li, Xianyong II-248
Li, Xuan II-538
Li, Xuanyu II-235
Li, Yafu I-104
Li, Yaliang I-196
Li, Yang II-422
Li, Yaru II-354
Li, Yinzi II-548
Li, Yuming II-530
Li, Yuncong I-583
Li, Yunxin II-365
Li, Zheng II-614
Li, Zhengyan II-86
Li, Zhimin II-614
Li, Zhoujun I-79
Li, Zhucong I-338
Lian, Yixin II-206

- Liang, Jiahui I-439, II-124
 Liang, Wuyan I-773
 Lin, Hongfei I-619, I-746, II-579
 Lin, Nankai I-210
 Lin, Xiaotian I-210
 Ling, Xiao I-664
 Liu, Baoju I-27
 Liu, Hai II-139
 Liu, Huan I-129
 Liu, Huanyu I-493
 Liu, Jian I-92
 Liu, Jie I-664, I-734
 Liu, Junpeng I-129
 Liu, Kang I-338
 Liu, Mingtong I-365
 Liu, Peng I-300, II-509
 Liu, Pengyuan I-53, II-13, II-569
 Liu, Qin II-548
 Liu, Shen I-688
 Liu, Shengping I-338
 Liu, Shilei I-455
 Liu, Shudong I-517
 Liu, Wei I-183
 Liu, Weijie II-519
 Liu, Xiang I-709
 Liu, Xiao II-400
 Liu, Ximing II-206
 Liu, Yaduo II-460
 Liu, Yufei I-183
 Liu, Zhenghao II-275
 Liu, Zhengtao I-413
 Liu, Zhiyuan II-275
 Liu, Zihao I-288
 Lu, Xiaojing II-412
 Lu, Xiaolei II-27
 Lu, Xin II-260
 Lucaci, Diana II-303
 Luo, Weihua I-104
 Luo, Yichao II-86
 Luo, Yingfeng I-377
 Luo, Zhiyong I-709
 Luo, Zhunchen II-3
 Lv, Rongrong II-496
 Lv, Xin I-27
 Ma, Meirong I-688, II-377
 Ma, Shuming I-79
 Ma, Weifeng II-354
 Ma, Yutuan II-219
 Mao, Xian-Ling II-179
 Mao, Yuting I-53
 Meng, Yao I-365
 Miao, Qingliang I-505
 Miao, Xin I-570
 Mo, Yijun II-59
 Ni, Bin II-27
 Ni, Yuan I-169
 Ni, Zixin I-480
 Niu, Changyong II-219
 Niu, Yuchen I-758
 Nong, Wei I-325
 Ouchi, Kazushige I-92
 Pan, Jiaxin I-467
 Pan, Xiang II-412
 Pang, Shiguan I-531, I-544
 Peng, Boci II-98
 Peng, Huailiang I-557
 Peng, Min I-467
 Peng, Wei II-206
 Pourvali, Mohsen II-317
 Qi, Chao II-345
 Qian, Jin I-427
 Qin, Bing I-595, II-260
 Qin, Wentao II-288, II-337
 Qiu, Minghui II-422
 Qiu, Xipeng II-86
 Rana, Prashant Singh II-590
 Rao, Dongning I-570
 Ren, Feiliang I-455, II-460
 Rohit Kumar, A. S. II-323
 Schwarzenberg, Robert II-317
 Shao, Yifeng II-98
 Shen, Chenhui II-548
 Shen, Xinyao I-353
 Shen, Zhexu I-746
 Shi, Ge II-400
 Shi, Shumin I-141
 Shi, Xiaodong I-65
 Shi, Yafei I-338
 Si, Luo II-548
 Si, Yuke I-493
 Singh, Muskaan II-590
 Song, Dawei I-700
 Song, Linfeng II-484
 Su, Jianlin II-412
 Su, Wentao I-53

- Sui, Zhifang I-786
 Sun, Changlong II-548
 Sun, Fanshu II-179
 Sun, Haipeng I-439
 Sun, Jingyi II-472
 Sun, Si II-275
 Sun, Xichen II-3
 Sun, Xinghai II-345
 Sun, Yansong I-746
 Sun, Yuxiang II-377
 Takanabu, Ryuichi II-206
 Tan, Conghui II-43
 Tan, Jianying I-517
 Tan, Xin I-3
 Tang, Bixia I-544
 Tang, Chengguang II-422
 Tang, Jin I-413
 Tang, Jizhi I-676
 Tian, Dawei I-65
 Tong, Yiqi I-65, II-27
 Wan, Dazhen II-206
 Wan, Jing I-338
 Wan, Juncheng I-79
 Wang, Bang II-59
 Wang, Bin II-163
 Wang, Bingquan I-664
 Wang, Bo II-400
 Wang, Chengyu II-422
 Wang, Chenxu I-631
 Wang, Cunxiang I-758
 Wang, Danqing I-389
 Wang, Fang I-583
 Wang, Haocheng I-493
 Wang, Huizhen I-288
 Wang, Jin I-224, II-111
 Wang, Jinfeng I-40
 Wang, Liang II-73
 Wang, Liudi II-354
 Wang, Longbiao I-493
 Wang, Ming II-422
 Wang, Qiqi I-262
 Wang, Rui II-472
 Wang, Ruifang I-493
 Wang, Shanpeng I-664
 Wang, Sheng I-275
 Wang, Shuheng I-141, II-538
 Wang, Xiaoling I-169
 Wang, Xiaolong II-365
 Wang, Yan II-603
 Wang, Yashen I-643
 Wang, Yidong II-530
 Wang, Yifan II-124
 Wang, Zichun II-98
 Wei, Binghao I-377
 Wei, Furu I-79
 Wei, Guoao II-412
 Wei, Zhongyu I-401, II-548
 Wen, Ji-Rong I-196
 Wong, Derek I-104
 Wong, Leung-Pun II-139
 Wu, Chunhua I-183
 Wu, Jialun II-235
 Wu, Jiaming I-619
 Wu, Jipeng II-472
 Wu, Lifang II-400
 Wu, Shuangzhi II-388
 Wu, Taiqiang II-519
 Wu, Xianze I-389
 Wu, Yang I-595
 Wu, Yi II-569
 Wu, Yike I-798
 Wu, Youzheng I-439, II-124
 Wu, Yuanbin I-688
 Xia, Maojin II-530
 Xia, Qiaolin I-786
 Xia, Qingrong I-607
 Xia, Yu I-312
 Xiang, Lu II-193
 Xiang, Wei II-59
 Xiao, Jinghui I-27
 Xiao, Tong I-377
 Xiao, Yanghua I-353
 Xie, Guotong I-169
 Xie, Jiaying I-734
 Xie, Jin I-401
 Xiong, Chenyan II-275
 Xiong, Deyi I-365
 Xiong, Shengwu I-275
 Xu, Bo I-619
 Xu, Chen I-377
 Xu, Heng-Da II-179
 Xu, Huilin II-412
 Xu, Jinan I-92, I-365
 Xu, Ke I-53
 Xu, Kun II-484
 Xu, Liang II-412, II-434
 Xu, Mingzhou I-104

- Xu, Qian II-43
 Xu, Ruifeng II-472
 Xu, Sheng I-239, II-73
 Xu, Weijie II-377
 Xu, Xiaolong I-773
 Xu, Yajing II-43
 Xu, Yige II-86
 Xu, Ziyun II-422
 Xue, Yun I-531, I-544
- Yan, Zehao I-531, I-544
 Yang, Caihua II-472
 Yang, Erguang I-365
 Yang, Jian I-79
 Yang, Liang I-619, I-746, II-579
 Yang, Meng II-151
 Yang, Ni II-151
 Yang, Shimin II-614
 Yang, Shuangji I-325
 Yang, Xuefeng II-519
 Yang, Yan II-377
 Yang, Yujiu II-519
 Yang, Zhenyu II-412
 Yang, Zinong I-116
 Yao, Bolun I-401
 Yin, Cunxiang I-583
 Yin, Shujuan II-460
 Yin, Xuemeng II-235
 Yin, Zimo II-538
 Ying, Jiahao II-377
 Yu, Dong II-13, II-484
 Yu, Kai I-505, I-810
 Yu, Yong I-79
 Yuan, Changsen II-400
 Yuan, Chenyang II-412
 Yuan, Hao I-688, II-377
 Yuan, Hu II-412
 Yuan, Jian II-548
 Yuan, Xiaojie I-798
 Yuan, Youliang I-467
- Zan, Hongying II-219
 Zeng, Jiali II-388
 Zeng, Jingjie I-746
 Zeng, Xin I-377
 Zhang, Baoli I-338
 Zhang, Bosen II-43
 Zhang, Chen I-700
 Zhang, Dongdong I-79, I-786
 Zhang, Dongmei I-183
- Zhang, Fan II-579
 Zhang, Guocheng I-652
 Zhang, Haichuan II-235
 Zhang, Hengtong I-196
 Zhang, Hu II-13
 Zhang, Hualin I-517
 Zhang, Jiaxin I-664
 Zhang, Jun II-377
 Zhang, Longhui II-460
 Zhang, Longyin I-3, I-40
 Zhang, Min I-607
 Zhang, Mingming I-709
 Zhang, Qi II-86
 Zhang, Sanle II-13
 Zhang, Shaokang I-557
 Zhang, Shaowu II-579
 Zhang, Shuonan II-548
 Zhang, Taolin I-325
 Zhang, Wei II-538
 Zhang, Weicong II-219
 Zhang, Weiqi I-721
 Zhang, Wenzheng I-664
 Zhang, Xiaoqiang I-822
 Zhang, Xuanwei II-412
 Zhang, Xuejie I-224, II-111
 Zhang, Yadong II-447
 Zhang, Yan I-288
 Zhang, Yanru II-603
 Zhang, Ying I-798
 Zhang, Yue I-104, I-758
 Zhang, Yujie I-365
 Zhang, Yujing I-251
 Zhang, Yulai II-354
 Zhang, Yushi II-377
 Zhang, Zhenyu I-595
 Zhang, Zhirui I-104
 Zhang, Zhiyang I-721
 Zhao, Chongshuai I-300, II-509
 Zhao, Dongyan I-676, II-288, II-337
 Zhao, Jun I-338
- Zhao, Tiejun I-439
 Zhao, Wayne Xin I-196
 Zhao, Weixiang II-260
 Zhao, Xiaofeng I-455, II-460
 Zhao, Yang II-193
 Zhao, Yanyan I-595, II-260
 Zhao, Yao I-810
 Zhao, Yu II-365
 Zhao, Zhe II-519

- Zhao, Zhilin I-709
Zheng, Boyuan I-758
Zheng, Derong II-538
Zheng, Xuling I-652
Zhong, Sheng-hua I-583
Zhong, Yuyanzhen I-721
Zhou, Bartuer I-401
Zhou, Bowen I-439, II-124
Zhou, Feng I-251
Zhou, Gongxue II-354
Zhou, Guodong I-3
Zhou, Hao I-389
Zhou, Ming I-786
Zhou, Xiabing I-607
Zhou, Xinya II-43
Zhou, Yicheng II-86
Zhou, Yongwei I-439
- Zhou, Yu II-193
Zhu, Jianchao I-688, II-377
Zhu, Jingbo I-288, I-377
Zhu, Jingjing II-179
Zhu, Junnan II-193
Zhu, Juyi I-721
Zhu, Ming I-65
Zhu, Qiaoming I-15, I-239, II-73
Zhu, Qinglin II-472
Zhu, Su I-505
Zhu, Wei I-155, I-169
Zhu, Xiaozhi II-139
Zhu, Yujin II-434
Zhu, Zhanbiao I-517
Zong, Chengqing II-193
Zou, Bowei I-427