Renze Lou

Phone: (86) 177 6721 6840 Email: marionojump0722@gmail.com Personal Website

EDUCATION

Zhejiang University City College (ZUCC), Hangzhou, China Sept 2018-Jul 2022
Bachelor of Engineering in Software Engineering GPA: 3.88/4.0 Average Score: 89.79/100 Rank: 2/66

SELECTED PUBLICATIONS

(* indicates equal contribution)

1. Yutong Wang*, **Renze Lou***, Kai Zhang*, Maoyan Chen and Yujiu Yang. MORE: A Metric Learning Based Framework for Open-Domain Relation Extraction. In ICASSP 2021.

   - Accomplish the whole paper writing and the rebuttal letter.
   - Study intensively in related fields, improve our methodology with VAT strategy, gain a better performance.
   - Lead and design all the experiments, complete most of the codes (90%).

2. Weicheng Ma*, Kai Zhang*, **Renze Lou**, Lili Wang and Soroush Vosoughi. Contributions of Transformer Attention Heads in Multi- and Cross-lingual Tasks. In ACL 2021 (**Long Oral**).

   - Participate in collecting massive NLP corpus and help design experiment settings.
   - Handle the NER task of our experiments. Help refine our algorithm according to the observation on head-masked distribution and some other unexpected results.

3. Weicheng Ma*, **Renze Lou***, Kai Zhang, Lili Wang and Soroush Vosoughi. GradTS: A Gradient-Based Automatic Auxiliary Task Selection Method Based on Transformer Networks. In EMNLP 2021.

   - Write the details of experiments, participate designing the paper layout, provide the materials used in paper.
   - Think up several intuitive control experiments to improve our experimental design.
   - Lead all the experiments, complete most of our codes and reproduce the strong closed-source baseline.

RESEARCH EXPERIENCE

Remote Research Intern. Minds, Machines and Society Group. Dartmouth College
Advisor: Soroush Vosoughi, Google Scholar Oct 2020-Aug 2021

Remote Research Intern. Intelligent Computing Laboratory. Tsinghua University
Advisor: Yujiu Yang, Research Gate Jun 2020-Oct 2020

Research Assistant. Institute of Artificial Intelligence. Zhejiang University City College
Advisors: Lin Sun, Google Scholar and Minghui Wu, Google Scholar May 2020-Mar 2021

WORK EXPERIENCE

Hangzhou Maixiang Health Technology Co., Ltd *Zhejiang, China*
Jul 2021- Sept 2021 *NLP Algorithm Intern*

Research on the Application of Traditional Chinese Medical Knowledge Graph:

- Communicate with doctors of traditional Chinese medicine, capture their needs.
- Investigate related works and employ python to crawl data on the internet.
- Utilize our algorithm (the one accepted by ICASSP 2021) to detect noisy samples in clinical data.

PROJECTS

1. Research on Open Relation Extraction in Medical Therapic Recording *Leader*; May 2021- May 2022
National Innovation Training Program for College Students (202113021002) 15,000 RMB

2. Research on Named Entity Recognition in Social Media *Member*; May 2021-May 2022
National Innovation Training Program for College Students (202113021003) 15,000 RMB

SERVICES

External Reviewer: EMNLP 2021; IJCAI 2021; JIFS.

Leadership: Co-Founder of CCAi (the first institution for Artificial Intelligence at ZUCC).

Teaching Asistance: Deep Learning Application Development; Object-oriented Programming.

SKILLS

Programming: Python; LaTeX; Java; C/C++; Matlab.

Tools: Pytorch; Transformers (huggingface); Keras; Scikit-learn.

ZHEJIANG UNIVERSITY
City College Student's Academic Records

Website: http://zzfw.zucc.edu.cn
S/N: ZUCCANMCJMNLKJBMMJPD

Name:Lou Renze	College/Dept.:School of Computer & Computing Science	Speciality:Software Engineering	Student ID:31801316
Sex:Male	Birthday:22/07/2000 Birth Place：Zhejiang	Entrance Date:11/09/2018 Graduation Date:	Years of Program:4Years

Academic Year 2018-2019			Object-Oriented Programming	3.0	91	Software Testing Principiple & Practice	3.0	88			
Courses(1st) Term	***Cr**	***Sc**	Physical Education	1.0	84	Software Project Management Principle & Practice	3.0	83			
Programming Foundation and Experiment	4.5	95	Comprehensive Course Design Ⅰ For Software Engineering Mojor	2.0	B	Software Requirement Analysis Principle & Practice	3.0	90			
College English (III)	4.0	87	Data Structure & Algorithm	3.0	98						
Principles of Marxism	3.0	80	Ideological and Political Theory and Social Practice	1.0	P						
Introduction to Software Engineering	2.0	94	A Concise History of Modern & Contemporary China	2.0	91						
Moral Cultivation and Legal Knowledge	3.0	82	**Courses(2nd) Term**	***Cr**	***Sc**						
Physical Education	1.0	83	Principles of Operating System	3.0	89						
Calculus(Ⅰ)	4.5	96	Lab. of Operating System Principles	1.0	B						
Linear Algebra	3.0	94	Introduction to Big Data	2.0	94						
Courses(2nd) Term	***Cr**	***Sc**	Audio-Visual And Oral English	2.0	82						
University Physics(Ⅰ)	4.0	100	Probability & Mathematical Statistics (A)	3.0	97						
Experiment of College Physics	1.5	94	Computer Networks	3.0	86						
College English (IV)	4.0	91	Lab. of Computer Networks	1.0	B						
History and Application of Laser	2.0	91	Database Principles	4.0	75						
Introduction to Discrete Mathematics	2.0	97	Mathematical Modeling	3.0	70						
Physical Education	1.0	92	Academic Writing & Documentation by Word and LaTeX	2.0	86						
Basics of Data Structure	3.0	91	**Academic Year 2020-2021**								
Introduction to Digital Logic and Sensors	3.0	95	**Courses(1st) Term**	***Cr**	***Sc**						
Calculus(II)A	5.5	98	Logic & Eloquence	2.0	91						
Academic Year 2019-2020			Introduction To Artificial Intelligence	3.0	92						
Courses(1st) Term	***Cr**	***Sc**	Software Engineering	3.0	86						
Innovate And Entrepreneurship	2.0	85	Comprehensive Course Design Ⅱ For Software Engineering Major	3.0	B						
University Physics(Ⅱ)	2.0	93	Software Architecture Principle & Practice	3.0	91						
Invitation To Public Speaking In English-Mastering The Basic	2.0	95	Deep Learning Application Development	3.0	96						
Computer System	4.0	96	Introduction to Data Mining	3.0	97						
Principle of Computer Systems	1.0	B	**Courses(2nd) Term**	***Cr**	***Sc**				Overall GPA:3.88/4.0(89.79/100)		
Military Training & National Defense Education	1.0	93	Big Data Computing	3.0	85						
Mao Zedong Thought & Theoretical System of Socialism With Characteristics	4.0	91	Computer Vision	3.0	93						
Credits Required for Graduation：165			Credits Obtained：138.00						Degree Granted：		

浙大城市学院 成绩考核专用章

*Cr=Credits;*Sc=Score;
Three grade systems are used simultaneously in Zhejiang University City College,specifically as follows:
1.The percentage system:Above 60 is passing,100 is full mark；
2.Five degree grading:Excellent(A),Good(B),Fair(C),Passing(D),Failed(F)；
3.Two degree grading:Passing(P),Failed(F)；
4.■Indicate courses undertaken at other univeristies,the grades which met requirements of ZUCC are recorded as "Pass"。

Director of Academic Affairs Office: 颜晖

Registrar: Chen Peiyu

Date Issued:08/22/2021

Contributions of Transformer Attention Heads in Multi- and Cross-lingual Tasks

Weicheng Ma[1*], Kai Zhang[2*†], Renze Lou[3†], Lili Wang[1], and Soroush Vosoughi[4]

[1,4]Department of Computer Science, Dartmouth College
[2]Department of Computer Science and Technology, Tsinghua University
[3]Department of Computer Science, Zhejiang University City College
[1]{first.last}.gr@dartmouth.edu
[2]drogozhang@gmail.com
[3]marionojump0722@gmail.com
[4]soroush.vosoughi@dartmouth.edu

arXiv:2108.08375v1 [cs.CL] 18 Aug 2021

Abstract

This paper studies the relative importance of attention heads in Transformer-based models to aid their interpretability in cross-lingual and multi-lingual tasks. Prior research has found that only a few attention heads are important in each mono-lingual Natural Language Processing (NLP) task and pruning the remaining heads leads to comparable or improved performance of the model. However, the impact of pruning attention heads is not yet clear in cross-lingual and multi-lingual tasks. Through extensive experiments, we show that (1) pruning a number of attention heads in a multilingual Transformer-based model has, in general, positive effects on its performance in cross-lingual and multi-lingual tasks and (2) the attention heads to be pruned can be ranked using gradients and identified with a few trial experiments. Our experiments focus on sequence labeling tasks, with potential applicability on other cross-lingual and multi-lingual tasks. For comprehensiveness, we examine two pre-trained multi-lingual models, namely multi-lingual BERT (mBERT) and XLM-R, on three tasks across 9 languages each. We also discuss the validity of our findings and their extensibility to truly resource-scarce languages and other task settings.

1 Introduction

Prior research on mono-lingual Transformer-based (Vaswani et al., 2017) models reveals that a subset of their attention heads makes key contributions to each task, and the models perform comparably well (Voita et al., 2019; Michel et al., 2019) or even better (Kovaleva et al., 2019) with the remaining heads pruned [1]. While multi-lingual Transformer-based models, e.g. mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020), are widely applied in cross-lingual and multi-lingual NLP tasks [2] (Wang et al., 2019; Keung et al., 2019; Eskander et al., 2020), no attempt has been made to extend the findings on the aforementioned mono-lingual research to this context. In this paper, we explore the roles of attention heads in cross-lingual and multi-lingual tasks for two reasons. First, better understanding and interpretability of Transformer-based models leads to efficient model designs and parameter tuning. Second, head-pruning makes Transformer-based models more applicable to truly resource-scarce languages if it does not negatively affect model performance significantly.

The biggest challenge we face when studying the roles of attention heads in cross-lingual and multi-lingual tasks is locating the heads to prune. Existing research has shown that each attention head is specialized to extract a collection of linguistic features, e.g., the middle layers of BERT mainly extract syntactic features (Vig and Belinkov, 2019; Hewitt and Manning, 2019) and the fourth head on the fifth layer of BERT greatly contributes to the coreference resolution task (Clark et al., 2019). Thus, we hypothesize that important feature extractors for a task should be shared across languages and the remaining heads can be pruned. We evaluate two approaches used to rank attention heads, the first of which is layer-wise relevance propagation (LRP, Ding et al. (2017)). Voita et al. (2019) interpreted the adaptation of LRP in Transformer-based models on machine translation. Motivated by Feng et al. (2018) and Serrano and Smith (2019), we design a second ranking method based on gradients since the gradients on each attention head

[*] Equal contribution.

[†] Work done when interning at the Minds, Machines, and Society Lab at Dartmouth College.

[1] We regard single-source machine translation as a monolingual task since the inputs to the models are mono-lingual.

[2] We define a cross-lingual task as a task whose test set is in a different language from its training set. A multi-lingual task is a task whose training set is multi-lingual and the languages of its test set belong to the languages of the training set.

reflect its contribution to the predictions.

We study the effects of pruning attention heads on three sequence labeling tasks, namely part-of-speech tagging (POS), named entity recognition (NER), and slot filling (SF). We focus on sequence labeling tasks since they are more difficult to annotate than document- or sentence-level classification datasets and require more treatment in cross-lingual and multi-lingual research. We choose POS and NER datasets in 9 languages, where English (EN), Chinese (ZH), and Arabic (AR) are candidate source languages. The MultiAtis++ corpus (Xu et al., 2020) is used in the SF evaluations with EN as the source language. We do not include syntactic chunking and semantic role labeling tasks due to lack of availability of manually written and annotated corpora. In these experiments, we rank attention heads based only on the source language(s) to ensure the extensibility of the learned knowledge to cross-lingual tasks and resource-poor languages.

In our preliminary experiments comparing the gradient-based method and LRP, the average F1 score improvements on NER with mBERT are 0.69 (cross-lingual) and 0.24 (multi-lingual) for LRP and 0.81 (cross-lingual) and 0.31 (multi-lingual) for the gradient-based method, though both methods rank attention heads similarly.

Thus we choose the gradient-based method to rank attention heads in all our experiments.

Our evaluations confirm that only a subset of attention heads in each Transformer-based model makes key contributions to each cross-lingual or multi-lingual task and that these heads are shared across languages.

Performance of models generally drop when the highest-ranked or randomly selected heads are pruned, validating the head rankings generated by our gradient-based method. We also observe performance improvements on tasks with multiple source languages by pruning attention heads. Our findings potentially apply to truly resource-scarce languages since we show that the models perform better with attention heads pruned when fewer training instances are available in the target languages.

The contributions of this paper are three-fold:

- We explore the roles of attention heads in multi-lingual Transformer-based models and find that pruning certain heads leads to comparable or better performance in cross-lingual and multi-lingual sequence labeling tasks.
- We adapt a gradient-based method to locate attention heads that can be pruned without exhaustive experiments on all possible combinations.
- We show the correctness, robustness, and extensibility of the findings and our head ranking method under a wide range of settings through comprehensive experiments.

LC	Language Family	Training Size	
		POS	NER
EN	IE, Germanic	12,543	14,987
DE	IE, Germanic	13,814	12,705
NL	IE, Germanic	12,264	15,806
AR	Afro-Asiatic, Semitic	6,075	1,329
HE	Afro-Asiatic, Semitic	5,241	2,785
ZH	Sino-Tibetan	3,997	20,905
JA	Japanese	7,027	800
UR	IE, Indic	4,043	289,741
FA	IE, Iranian	4,798	18,463

Table 1: Details of POS and NER datasets in our experiments. LC refers to language code. Training size denotes the number of training instances.

2 Datasets

We use human-written and manually annotated datasets in experiments to avoid noise from machine translation and automatic label projection.

We choose POS and NER datasets in 9 languages, namely EN, ZH, AR, Hebrew (HE), Japanese (JA), Persian (FA), German (DE), Dutch (NL), and Urdu (UR). As Table 1 shows, these languages fall in diverse language families and the datasets are very different in size. EN, ZH, and AR are used as candidate source languages since they are resource-rich in many NLP tasks. Our POS datasets are all from Universal Dependencies (UD) v2.7 [3]. These datasets are labeled with a common label set containing 17 POS tags.

For NER, we use NL, EN, and DE datasets from CoNLL-2002 and 2003 challenges (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003). Additionally, we use the People's Daily dataset [4], iob2corpus [5], AQMAR (Mohit et al., 2012), ArmanPerosNERCorpus (Poostchi et al., 2016), MK-PUCIT (Kanwal et al., 2020), and a news-based NER dataset (Mordecai and Elhadad, 2012) for the languages CN, JA, AR, FA, UR, and

[3] http://universaldependencies.org/

[4] http://github.com/OYE93/Chinese-NLP-Corpus/tree/master/NER/People'sDaily

[5] http://github.com/Hironsan/IOB2Corpus

HE, respectively. Since the NER datasets are individually constructed in each language, their label sets do not fully agree. As there are four NE types (PER, ORG, LOC, MISC) in the three source-language datasets, we merge other NE types into the MISC class to allow cross-lingual evaluations.

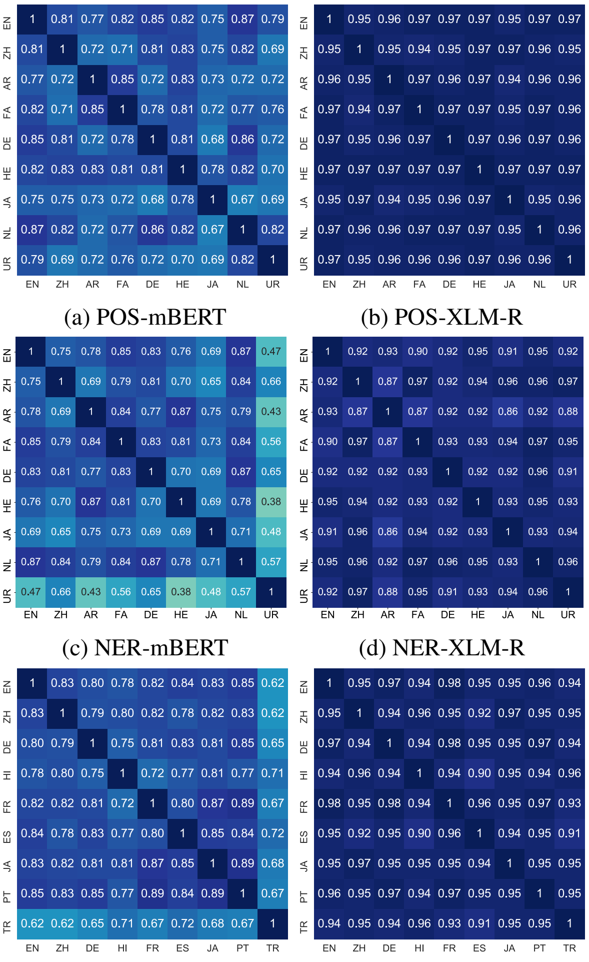
We evaluate SF models on MultiAtis++ with EN as the source language and Spanish (ES), Portuguese (PT), DE, French (FR), ZH, JA, Hindi (HI), and Turkish (TR) as target languages. There are 71 slot types in the TR dataset, 75 in the HI dataset, and 84 in the other datasets. We do not use the intent labels in our evaluations since we study only sequence labeling tasks. Thus our results are not directly comparable with Xu et al. (2020).

3 Methodology

Here, we introduce the gradient-based method we use in the experiments to rank the attention heads. Feng et al. (2018) claim that gradients measure the importance of features to predictions. Since each head functions similarly as a standalone feature extractor in a Transformer-based model, we use gradients to approximate the importance of the feature set extracted by each head and rank the heads accordingly. Michel et al. (2019) determine importance of heads with accumulated gradients at each head in a training epoch. Different from their approach, we fine-tune the model on the training set and rank the heads using gradients on the development set to ensure that the head importance rankings are not significantly correlated with the training instances in one source language.

Specifically, our method generates head rankings for each language in three steps:
(1) We fine-tune a Transformer-based model on a mono-lingual task for three epochs.
(2) We re-run the fine-tuned model on the development partition of the dataset with back-propagation but not parameter updates to obtain gradients.
(3) We sum up the absolute gradients on each head, layer-wise normalize the accumulated gradients, and scale them into the range [0, 1] globally.

We show Spearman's rank correlation coefficients (Spearman's ρ) between head rankings of each language pair generated by our method on POS, NER, and SF in Figure 1. The highest-ranked heads largely overlap in all three tasks, while the rankings of unimportant heads vary more in mBERT than XLM-R.



	EN	ZH	AR	FA	DE	HE	JA	NL	UR
EN	1	0.81	0.77	0.82	0.85	0.82	0.75	0.87	0.79
ZH	0.81	1	0.72	0.71	0.81	0.83	0.75	0.82	0.69
AR	0.77	0.72	1	0.85	0.72	0.83	0.73	0.72	0.72
FA	0.82	0.71	0.85	1	0.78	0.81	0.72	0.77	0.76
DE	0.85	0.81	0.72	0.78	1	0.81	0.68	0.86	0.72
HE	0.82	0.83	0.83	0.81	0.81	1	0.78	0.82	0.70
JA	0.75	0.75	0.73	0.72	0.68	0.78	1	0.67	0.69
NL	0.87	0.82	0.72	0.77	0.86	0.82	0.67	1	0.82
UR	0.79	0.69	0.72	0.76	0.72	0.70	0.69	0.82	1

(a) POS-mBERT

	EN	ZH	AR	FA	DE	HE	JA	NL	UR
EN	1	0.95	0.96	0.97	0.97	0.97	0.95	0.97	0.97
ZH	0.95	1	0.95	0.94	0.95	0.97	0.97	0.96	0.95
AR	0.96	0.95	1	0.97	0.96	0.97	0.94	0.96	0.96
FA	0.97	0.94	0.97	1	0.97	0.97	0.95	0.97	0.96
DE	0.97	0.95	0.96	0.97	1	0.97	0.96	0.97	0.96
HE	0.97	0.97	0.97	0.97	0.97	1	0.97	0.97	0.97
JA	0.95	0.97	0.94	0.95	0.96	0.97	1	0.95	0.96
NL	0.97	0.96	0.96	0.97	0.97	0.97	0.95	1	0.96
UR	0.97	0.95	0.96	0.96	0.96	0.97	0.96	0.96	1

(b) POS-XLM-R

	EN	ZH	AR	FA	DE	HE	JA	NL	UR
EN	1	0.75	0.78	0.85	0.83	0.76	0.69	0.87	0.47
ZH	0.75	1	0.69	0.79	0.81	0.70	0.65	0.84	0.66
AR	0.78	0.69	1	0.84	0.77	0.87	0.75	0.79	0.43
FA	0.85	0.79	0.84	1	0.83	0.81	0.73	0.84	0.56
DE	0.83	0.81	0.77	0.83	1	0.70	0.69	0.87	0.65
HE	0.76	0.70	0.87	0.81	0.70	1	0.69	0.78	0.38
JA	0.69	0.65	0.75	0.73	0.69	0.69	1	0.71	0.48
NL	0.87	0.84	0.79	0.84	0.87	0.78	0.71	1	0.57
UR	0.47	0.66	0.43	0.56	0.65	0.38	0.48	0.57	1

(c) NER-mBERT

	EN	ZH	AR	FA	DE	HE	JA	NL	UR
EN	1	0.92	0.93	0.90	0.92	0.95	0.91	0.95	0.92
ZH	0.92	1	0.87	0.97	0.92	0.94	0.96	0.96	0.97
AR	0.93	0.87	1	0.87	0.92	0.92	0.86	0.92	0.88
FA	0.90	0.97	0.87	1	0.93	0.93	0.94	0.97	0.95
DE	0.92	0.92	0.92	0.93	1	0.92	0.92	0.96	0.91
HE	0.95	0.94	0.92	0.93	0.92	1	0.93	0.95	0.93
JA	0.91	0.96	0.86	0.94	0.92	0.93	1	0.93	0.94
NL	0.95	0.96	0.92	0.97	0.96	0.95	0.93	1	0.96
UR	0.92	0.97	0.88	0.95	0.91	0.93	0.94	0.96	1

(d) NER-XLM-R

	EN	ZH	DE	HI	FR	ES	JA	PT	TR
EN	1	0.83	0.80	0.78	0.82	0.84	0.83	0.85	0.62
ZH	0.83	1	0.79	0.80	0.82	0.78	0.82	0.83	0.62
DE	0.80	0.79	1	0.75	0.81	0.83	0.81	0.85	0.65
HI	0.78	0.80	0.75	1	0.72	0.77	0.81	0.77	0.71
FR	0.82	0.82	0.81	0.72	1	0.80	0.87	0.89	0.67
ES	0.84	0.78	0.83	0.77	0.80	1	0.85	0.84	0.72
JA	0.83	0.82	0.81	0.81	0.87	0.85	1	0.89	0.68
PT	0.85	0.83	0.85	0.77	0.89	0.84	0.89	1	0.67
TR	0.62	0.62	0.65	0.71	0.67	0.72	0.68	0.67	1

(e) SF-mBERT

	EN	ZH	DE	HI	FR	ES	JA	PT	TR
EN	1	0.95	0.97	0.94	0.98	0.95	0.95	0.96	0.94
ZH	0.95	1	0.94	0.96	0.95	0.92	0.97	0.95	0.95
DE	0.97	0.94	1	0.94	0.98	0.95	0.95	0.97	0.94
HI	0.94	0.96	0.94	1	0.94	0.90	0.95	0.94	0.96
FR	0.98	0.95	0.98	0.94	1	0.96	0.95	0.97	0.93
ES	0.95	0.92	0.95	0.90	0.96	1	0.94	0.95	0.91
JA	0.95	0.97	0.95	0.95	0.95	0.94	1	0.95	0.95
PT	0.96	0.95	0.97	0.94	0.97	0.95	0.95	1	0.95
TR	0.94	0.95	0.94	0.96	0.93	0.91	0.95	0.95	1

(f) SF-XLM-R

Figure 1: Spearman's ρ of head ranking matrices between languages in the POS, NER, and SF tasks. Darker colors indicate higher correlations.

After ranking the attention heads, we fine-tune the model, with the lowest-ranked head in the source language pruned. We keep increasing the number of heads to prune until it reaches a pre-set limit or when the performance starts to drop. We limit the number of trials to 12 since the models mostly show improved performance within 12 attempts [6].

4 Experiments and Analysis

This section displays and explains experimental results on cross-lingual and multi-lingual POS, NER, and SF tasks. Training sets in target languages are not used to train the model under the cross-lingual setting. Our experiments are based on the Huggingface (Wolf et al., 2020) implementations of mBERT

[6] On average 7.52 and 6.58 heads are pruned for POS, 7.54 and 7.28 heads for NER, and 6.19 and 6.31 heads for SF, respectively in mBERT and XLM-R models.

MORE: A METRIC LEARNING BASED FRAMEWORK FOR OPEN-DOMAIN RELATION EXTRACTION

Yutong Wang[1*], *Renze Lou*[3*], *Kai Zhang*[2*], *Mao Yan Chen*[1], *Yujiu Yang*[1†]

[1]Tsinghua Shenzhen International Graduate School, Tsinghua University
[2]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[3]Department of Computer Science, Zhejiang University City College, Hangzhou, China

ABSTRACT

Open relation extraction (OpenRE) is a task of extracting relation schemes from open-domain corpora. Most existing OpenRE methods either do not fully benefit from high-quality labeled corpora or can not learn semantic representation directly, affecting downstream clustering efficiency. To address these problems, in this work, we propose a novel learning framework named MORE (**M**etric learning-based **O**pen **R**elation **E**xtraction). The framework utilizes deep metric learning to obtain rich supervision signals from labeled data and drive the neural model to learn semantic relational representation directly. Experiments result in two real-world datasets show that our method outperforms other state-of-the-art baselines. Our source code is available on Github[1].

Index Terms— Open-domain, relation extraction, deep metric learning

1. INTRODUCTION

Relation extraction (RE) is an important NLP task that aims to detect and categorize semantic relations between entities. Moreover, it is the key ingredient in many applications, such as knowledge graph construction [1], information retrieval [2], and logic reasoning [3]. However, with the rapid emergence of novel knowledge, the corresponding types of relations in open-domain corpora are also increasing, which is challenging for RE to handle. Thus, OpenRE is proposed to solve this problem [4], which aims at extracting relational schemes from the open-domain corpus without predefined relation types.

Existing OpenRE methods are divided into two main categories: tagging-based and clustering-based. The tagging-based methods formulate OpenRE as a sequence labeling problem [4, 5], but these methods often extract surface forms and are difficult to be utilized for downstream tasks. Meanwhile, many efforts are devoted to exploring clustering-based methods that cluster semantic patterns into certain relation types, such as [6, 7], yet those schemes are laborious and time-consuming due to the high dependence on rich features. Recently, neural networks began to be exploited in clustering-based OpenRE tasks to alleviate the above issues. For example, Hu et al. [8] utilize a neural model to capture self-supervision signals and detect novel instances in an open scene. Gao et al. [9] propose a siamese network, which accumulates novel types with its few-shot instances. Besides these bootstrapping methods, another supervised scheme learn the similarity metrics from labeled instances and further transfer the relational knowledge to open-domain corpora, namely Relational Siamese Networks (RSNs) [10]. However, RSNs target at learning a similarity classifier rather than building relational representations directly. Thus this may affect the speed and efficiency of downstream clustering.

To address this issue, in this paper, we propose a new supervised learning scheme, which applies deep metric learning to clustering-based OpenRE and build semantic embeddings directly. From our insight view, the essential objective of the cluster-based OpenRE algorithm is to distinguish relational texts and detect the novel classes. Thus, learning semantic representations from sentences is crucial for downstream clustering. Furthermore, metric learning algorithms aim to learn the representations and acquire semantic space directly. Therefore, it is rational to take metric learning into count. However, most prevailing deep metric learning methods, such as triplet loss [11], N-pair-mc [12], or Proxy-NCA [13], always suffer from poor supervision signals from the limited number of data points, which may harm the performance of novel relation detection. Inspired by [14], we take Ranked List Loss (RLL) as our choice, which can capture set-based supervision signals and gain richer information. In addition, we design virtual adversarial training to enhance model's robustness and deal with the noise in open scenes. Our experiments demonstrate that MORE can learn more semantic representations and achieve state-of-the-art results on real-world datasets.

To sum up, the main contributions of us are as follows:

1. To the best of our knowledge, we are the first to adopt deep metric learning in OpenRE tasks and propose a new

[*] Indicates equal contribution.

[†] Corresponding author(yang.yujiu@sz.tsinghua.edu.cn)

[1] `https://github.com/RenzeLou/MORE`

learning framework MORE that can learn semantic representations and be used for downstream clustering directly. Meanwhile, we also design virtual adversarial training on our model to smooth the semantic space.

2. Experiments illustrate that the proposed MORE achieves state-of-the-art performance on real-world datasets, even if the imbalance distribution presents in the test set. Moreover, the visual analysis also demonstrates its excellent ability of representation learning.

2. METHODOLOGY

As shown in Figure 1, the proposed framework MORE exploits the neural encoders to extract relational representations and use them to calculate the Ranked List Loss (RLL). Besides, we set virtual adversarial training to smooth the semantic space.

2.1. Neural Encoders

As a vital component of our method, neural encoder aim at learning semantic representations of relation types. In this paper, we have experimented with two different encoders.

CNN Following [10], we take the CNN encoder as our first choice, which includes an embedding layer followed by a one-dimensional convolutional layer and a max-pooling layer. The model setting we used is the same as [10].

BERT Inspired by SelfORE [8], which exploits the pre-trained language model, we also choose BERT [15] as our encoder and follow the operation proposed by [16] to fit our OpenRE task better. Specifically, for a sentence $\mathcal{S} = \{s_1, .., s_T\}$, where s indicates the token and T is the length of $\mathcal{S}$. We insert four special tokens before and after each entity mentioned in a sentence and get a new sequence:

$$\mathcal{S} = [s_1, .., [E1_{start}], s_p, .., s_q, [E1_{end}], .., [E2_{start}], s_k, .., s_l, [E2_{end}], .., s_T] \tag{1}$$

We use this sequence as the input of BERT, and we concatenate the last hidden state of BERT's outputs corresponding to $[E1_{start}], [E2_{start}]$, take it as our relational representation.

After obtaining the relational representations, we then use a fully-connected layer to map representations into hidden states. Next, we apply L_2 normalization on every hidden state and use Euclidean distance to measure the similarity among them.

2.2. Ranked List Loss

We utilize deep metric learning to optimize the semantic space after defining the distance metric between representations in normalized Euclidean space. Unlike most other metric learning methods, such as triplet loss [11], N-pair-mc [12], which are limited from point-based or pair-based information. Ranked List Loss (RLL) explores the set-based similarity structure from the training batch, and obtain richer supervision signals. For an anchor selected from the training batch, RLL rank the similarity of all the same type (positive) points before the different categories (negative) points and preserve a margin between them.

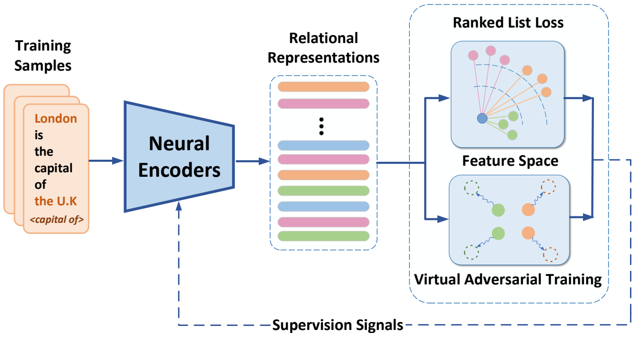


Fig. 1. Overall architecture of MORE.

To be specific, given a set of normalized relation representations $\mathcal{R} = \{r_1, .., r_n\}$, where n indicates the total number of labeled sentences. We sample $m(m \leq n)$ instances of c types randomly from $\mathcal{R}$ and group them into a batch $\mathcal{B} = \{r_k, .., r_{k+l}\}$. Intuitively, given an instance(anchor) r_i in $\mathcal{B}$, we expect the positive points in $\mathcal{B}$ can be gathered together while those negative points are the opposite. Then, we calculate the following formula:

$$\mathcal{L}(r_i, \mathcal{B}; f) = \sum_{r_j \in \mathcal{B}, j \neq i} [(1 - y_{ij})[\alpha_N - d_{ij}]_+ + y_{ij}[d_{ij} - \alpha_P]_+] \tag{2}$$

where f is model parameters, y indicates the relation type, $y_{ij} = 1$ if r_j is a positive point, and $y_{ij} = 0$ otherwise. d_{ij} denotes the Euclidean distance between two points. α_P, α_N represent the positive and negative boundary respectively, $[.]_+$ denote the hinge function. As shown in Figure 2, those positive instances outside the α_P will be pulled closer, while those negative points within the α_N will be pushed farther. The remaining uninformative points that already meet our objective will not be taken into count because of the hinge function.

More concisely, for r_i in $\mathcal{B}$, let's define $\mathcal{L}_P(r_i, \mathcal{B}; f)$ as the total loss of all informative positive points, and $\mathcal{L}_N(r_i, \mathcal{B}; f)$ is the sum of all informative negative samples loss, the optimization objective function can be summarized as below:

$$\mathcal{L}_{RLL}(\mathcal{B}; f) = \sum_{r_i \in \mathcal{B}} [(1 - \lambda)\mathcal{L}_P(r_i, \mathcal{B}; f) + \lambda \mathcal{L}_N(r_i, \mathcal{B}; f)] \tag{3}$$

Here, the λ is used to control the balance between $\mathcal{L}_P(r_i, \mathcal{B}; f)$ and $\mathcal{L}_N(r_i, \mathcal{B}; f)$, which we set to 0.5 practically.

Usually, given r_i as a anchor, numerous informative negative points can be found in $\mathcal{B}$. To deal with the magnitude difference laying in the negative loss, we follow [14], weighting the negative examples according to the values of their loss:

$$w_{ij} = \exp[T_n * (\alpha - d_{ij})], r_j \in \mathcal{R}_i^{\mathcal{N}} \tag{4}$$

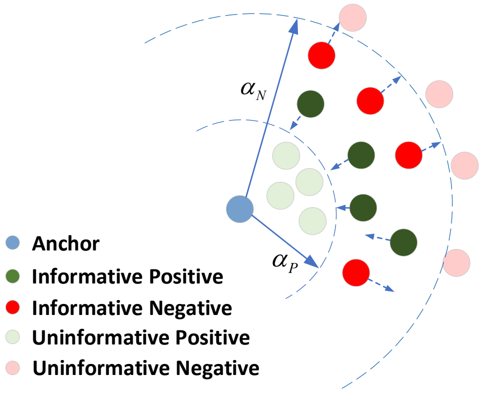


Fig. 2. Illustration of the ranked list loss.

Where $\mathcal{R}_i^{\mathcal{N}}$ represents the set of informative negative samples of r_i, and T_n indicates the temperature which controls the degree of weighting these samples. For example, when T_n is 0, every instance will be treated equally. And setting T_n to $+\infty$ when devoting almost all attention to the hardest sample.

Consequently, the $\mathcal{L}_N(r_i, \mathcal{B}; f)$ in (3) can be updated as:

$$\mathcal{L}_N(r_i, \mathcal{B}; f) = \sum_{r_j \in \mathcal{R}_i^{\mathcal{N}}} [\frac{w_{ij}}{\sum_{r_j \in \mathcal{R}_i^{\mathcal{N}}} w_{ij}} (\alpha_N - d_{ij})] \qquad (5)$$

After the model retrieve the supervision signals of $\mathcal{B}$, we sample the next m instances from $\mathcal{R}$ and build a new group iteratively.

2.3. Virtual adversarial training

To alleviate the diversity and noise present in the text of open scenarios, we set virtual adversarial training (VAT) [17] to smooth the semantic space and enhance the model's robustness.

Specifically, for a given sentence S and its original representation r, we first generate a normalized perturbation on the word embeddings within S randomly and take this disturbed embeddings as the input of encoder to build a new representation $\tilde{r}$. Next, we calculate the Euclidean distance between r and $\tilde{r}$ and take the gradient g of the distance. Then, we regard ϵ times normalized g as the worst-case perturbation ξ, where ϵ is a small decimal we set to 0.02 in all our experiments. Finally, we use ξ to disturb the representation build by the model and penalize our model with the VAT loss:

$$\mathcal{L}_{adv}(\mathcal{B}; f) = \frac{1}{m} \sum_{i=1}^{m} D(F(S_i; f), F(S_i + \xi_i; f)) \qquad (6)$$

$F(S_i + \xi_i; f)$ denotes the distributed representation encoded by the neural model, while $F(S_i; f)$ is the original one and D calculates the Euclidean distance between two representations. Intuitively, we expect any representation r_i in $\mathcal{B}$ is stable as possible under such worst-case perturbations.

Thence, the final objective loss function can be written as:

$$\mathcal{L}(\mathcal{B}; f) = \mathcal{L}_{RLL}(\mathcal{B}; f) + \beta \mathcal{L}_{adv}(\mathcal{B}; f) \qquad (7)$$

Where β is a factor that indicates the weight of virtual adversarial training, and we set it to 1, same as [17].

3. EXPERIMENT

In this section, we conduct some experiments on real-world RE datasets to show the effectiveness of our model.

3.1. Experiment Setting

Datasets We perform our experiments on two datasets. The first one is FewRel [18], derived from Wikipedia and annotated by crowd workers. Different from most other datasets, the entity pair of each instance in FewRel is unique, which makes the model unable to obtain shortcuts by memorizing the entities. Following [10], we choose 64 relations as the train set and randomly select 16 relations with 1600 instances as the test set; the remaining sentences are validation set. The other one is NYT+FB [19][20], which is built by distant supervision. To fit the supervision setting, we divide the original dataset again and generate NYT+FB-sup. Usually, the relations which occur frequently are common categories, and those relations with rare instances are insufficient to be regarded as novel types. Therefore, we select relations with the number of instances between 20 and 2000 as novel relations. We finally obtain 72 novel relations equally divided between the test and validation set, leaving 190 relations as the train set, which contain both common and extremely rare types to simulate an unbalanced real environment.

Evaluation Metrics We adopt B^3 F_1 score [21] as our metrics, which is widely used in previous works [10, 8, 7, 22]. F_1 calculates the harmonic mean of precision and recall, its value is more affected by the lower one, which can fairly demonstrate the performance of the model.

Implementation Details In all our experiments, we choose Adam [23] for our optimization. We fix the learning rate with 0.003 and 0.00001 on CNN and BERT respectively. For the content of a batch $\mathcal{B}$, we set the number of relation types c to 24, each type with 10 instances. For the hyperparameter α_P, α_N, we set 0.8 and 1.2 separately, thus preserving a margin of 0.4 between these two boundaries. The temperature factor T_n in this work is 10. On FewRel, we choose K-Means [24] as our downstream clustering algorithm and set the number of clusters with 16. And on NYT+FB-sup, to deal with the imbalance, we choose Mean-Shift [25] instead of K-means, which can automatically find clusters based on spatial density.

3.2. Result

We compare with four baselines [10, 8, 22, 7] on two datasets. All these models are evaluated on the test set to show their

GradTS: A Gradient-Based Automatic Auxiliary Task Selection Method Based on Transformer Networks

Anonymous EMNLP submission

Abstract

A key problem in multi-task learning (MTL) research is how to select high-quality auxiliary tasks automatically. This paper presents GradTS, an automatic auxiliary task selection method based on gradient calculation in Transformer-based models. Compared to AUTOSEM, a strong baseline method, GradTS improves the performance of MT-DNN with a bert-base-cased backend model, from 0.35% to 28.94% on 8 natural language understanding (NLU) tasks in the GLUE benchmarks. GradTS is also time-saving since (1) its gradient calculations are based on single-task experiments and (2) the gradients are re-used without additional experiments when the candidate task set changes. On the 8 GLUE classification tasks, for example, GradTS costs on average 21.32% less time than AUTOSEM with comparable GPU consumption. Further, we show the robustness of GradTS across various task settings and model selections, e.g. mixed objectives among candidate tasks. The efficiency and efficacy of GradTS in these case studies illustrate its general applicability in MTL research without requiring manual task filtering or costly parameter tuning.

1 Introduction

MTL (Caruana, 1997) is widely used in NLU research to improve the performance of machine learning (ML) models by enlarging the training data size with datapoints related to the primary tasks. However, its efficacy is largely affected by the selection of auxiliary tasks. The auxiliary task selection problem is addressed mainly under two settings. The first setting treats each task as a whole. For example, Bingel and Søgaard (2017) assess task relatedness by exhaustive experiments in all task pairs. Nonetheless, high pairwise task correlations are often not decisive features for choosing auxiliary tasks. Glover and Hokamp (2019) train a policy for task selection through counterfactual estimation, but their learned policy brings improvements only to one out of nine tasks on GLUE benchmarks (Wang et al., 2019b). The second setting subsamples training instances from auxiliary tasks, e.g. with Bayesian optimization (Ruder and Plank, 2017), but these methods are time- and resource-consuming due to their reliance on multi-task experiments involving all the candidate tasks. AUTOSEM (Guo et al., 2019) combines the two settings into one method, selecting candidate tasks with Thompson sampling and deciding the ratio with which to draw training instances from the selected tasks via a Gaussian Process. Despite the higher quality of the auxiliary task sets it generates, AUTOSEM is still costly, similar to Ruder and Plank (2017).

To design a better-performing and less costly auxiliary task selection method, we take advantage of the characteristics of Transformer networks (Vaswani et al., 2017). Prior research reveals that in a Transformer-based model, each attention head attends on specialized linguistic features (Clark et al., 2019; Voita et al., 2019; Mareček and Rosa, 2019; Lin et al., 2019; Vig and Belinkov, 2019; Kovaleva et al., 2019). Since important linguistic features strongly correlate with the goals of tasks, we further hypothesize that a good auxiliary task shares key linguistic features with the primary task. Thus, we address the auxiliary task selection problem by maximizing the overlap of important heads in a Transformer-based model between primary and auxiliary tasks. As Michel et al. (2019) claim, the importance of attention heads to a task can be approximated by the absolute gradients accumulated at each head. We design our auxiliary task selection method, GradTS, accordingly, by ranking the importance of attention heads for each individual task and modeling the correlation between each pair of tasks with their head rankings. By greedily selecting the tasks most closely related to the primary task, GradTS constructs auxiliary task sets through trial experiments (GradTS-trial). GradTS also enables task subsampling to further optimize auxiliary task sets. To achieve this goal, we design another setting of GradTS (GradTS-fg) that first assesses the correlations between the primary task and each training instance in an auxiliary task selected by GradTS-trial and then filters the training instances via thresholding.

We assess the strength of GradTS via MTL evaluations on 8 GLUE classification tasks. We use AUTOSEM and AUTOSEM-p1 [1] as our baselines since AUTOSEM is among the most advanced auxiliary task selection methods in the NLP field and because it features both task selection and task subsampling. For consistency, we use the bert-base-cased model as the backend model of GradTS, AUTOSEM, and the MTL framework. Results show that GradTS-trial produces better auxiliary task sets than AUTOSEM-p1 in all 8 GLUE tasks while costing on average 6.73% less time. In experiments with task subsampling, GradTS-fg again shows superior strength to AUTOSEM on all 8 tasks while costing 21.32% less time. These results strongly support the efficacy and efficiency of GradTS.

In addition to the main experiments, we compare GradTS to multiple intuitive auxiliary task selections to show its high performance. We also conduct case studies to show that GradTS is effective and robust on difficult tasks or candidate tasks with mixed objectives, e.g. classification, regression, and sequence labeling tasks. These findings reflect the general applicability of GradTS in various task settings. In comparison, auxiliary task sets produced by AUTOSEM and AUTOSEM-p1 are often not optimal in these complicated scenes. Further, GradTS reuses the head rankings when the candidate task set grows larger, which makes it even more time- and resource-efficient than existing methods.

The contributions of this paper are three-fold:

- we propose GradTS, an automatic auxiliary task selection method based on gradient calculation in pre-trained Transformer-based models;
- we illustrate the efficacy and efficiency of GradTS through comprehensive MTL evaluations; and
- we show, through case studies, the superior capability and robustness of GradTS to complicated candidate task settings compared to both AUTOSEM and auxiliary task selections based on human intuition.

2 Datasets

Following Guo et al. (2019), we use the 8 classification tasks in GLUE benchmarks (Wang et al., 2019b), namely CoLA, MRPC, MNLI, QNLI, QQP, RTE, SST-2, and WNLI, in our main experiments. [2] We apply the standard split of these datasets as Wang et al. (2019b) describe. [3]

We also use one regression and three sequence labeling tasks in our case studies about the efficacy of GradTS on candidate tasks with mixed training objectives. These tasks include STSB from GLUE benchmarks, Part-of-Speech tagging (POS) from Universal Dependencies [4], Named Entity Recognition (NER) from CoNLL-2003 challenges (Tjong Kim Sang and De Meulder, 2003), and Syntactic Chunking (SC) from CoNLL-2000 shared tasks (Tjong Kim Sang and Buchholz, 2000). The official data split of all these datasets is applied.

Additionally, we introduce MELD and Dyadic-MELD datasets (Poria et al., 2019) to verify the applicability of GradTS to tasks that are difficult for its backend model. While these two tasks are multi-modal emotion recognition tasks, we use only the textual data in the experiments. The MELD and Dyadic-MELD datasets are annotated with 7 emotion labels. The bert-base-cased model achieves F-1 scores less than 50% on both tasks, lower than its performance on most GLUE classification tasks.

Section A in the supplementary material (SM) describes the objectives, number of classes, and training data size of the 14 aforementioned tasks. We evaluate both accuracy and F-1 scores for MRPC and QQP, accuracy for QNLI, RTE, SST-2, MNLI [5] and WNLI, Matthew's correlation coefficient (MCC) for CoLA, Pearson's correlation coefficient and Spearman's correlation coefficient for STSB, and F-1 score for POS, NER, SC, MELD, and Dyadic-MELD tasks.

3 Methodology

We design GradTS based on the hypothesis that better auxiliary tasks share more important linguistic

[1] We refer to the AUTOSEM method without task subsampling as AUTOSEM-p1.

[2] We do not use SuperGLUE benchmarks (Wang et al., 2019a) to be consistent with the baseline method.

[3] We report scores on the development set of GLUE tasks due to the submission quota limit. We sample 10% training instances of each GLUE task with a random seed of 42 for choosing thresholds and selecting auxiliary tasks.

[4] https://universaldependencies.org/

[5] We report accuracy scores separately on the matched and mismatched splits of MNLI.

features with the primary task. Since each attention head in a Transformer-based model functions similarly as a standalone feature extractor on a specialized set of features, we approximate the important feature set of each task by the heads contributing the most to the task. As the key feature sets are task-specific, GradTS does not require multi-task experiments to rank auxiliary tasks given a primary task. This makes GradTS a time- and resource-economic method especially when the set of candidate auxiliary tasks is large or growing.

GradTS consists of three successive modules responsible for (1) ranking attention heads for a task based on their contributions, (2) ranking auxiliary tasks based on inter-task correlations, and (3) finalizing the auxiliary task sets, respectively.

3.1 Attention Head Ranking Module

We estimate the importance of attention heads to a task using the absolute gradients accumulated at each head, following Michel et al. (2019). Specifically, we achieve the goal in four steps: (1) We fine-tune a pre-trained Transformer-based model on a task. (2) We repeat the fine-tuning step on the training set of the task with the fine-tuned model, without updating parameters, to get gradients of the model. (3) We sum up the absolute gradients accumulated at each attention head. (4) We layer-wise normalize the accumulated gradients and scale the gradients to the range [0, 1] globally to represent the importance of each head for the given task.

In practice, we use pre-trained bert-base-cased model as the backend of GradTS and we fine-tune the model for three epochs before starting to accumulate gradients on each head [6]. The fine-tuning stage prior to gradient accumulation is designed to avoid large gradients on unimportant heads when the model is exposed to a downstream task for the first time. Figure S1 in SM visualizes the head importance matrix of the bert-base-cased model on MRPC, as an example.

3.2 Auxiliary Task Ranking Module

Given a primary task, we rank each candidate auxiliary task by the correlation between its head ranking matrix and that of the primary task. As Puth et al. (2015) suggest, we use Kendall's rank correlation coefficients (Kendall's τ) since the importance scores of heads seldom result in a tie, based on our observations. The task correlations for the 8 GLUE classification tasks are shown in Figure S2 in SM.

While the rankings of auxiliary tasks produced by GradTS are intuitive in some cases, e.g. the three natural language inference (NLI) tasks are good auxiliary tasks for each other, the correlation scores between many seemingly unrelated tasks, e.g. WNLI and CoLA, are also high. This reveals the difficulty of manually designing auxiliary task sets since the factors affecting the appropriateness of auxiliary tasks are multi-faceted, e.g. text lengths and label distributions. As a result, designing automatic methods for selecting auxiliary tasks makes up a crucial part of MTL research, especially at a time when candidate auxiliary tasks are rapidly growing both larger in amount and more complex.

3.3 Auxiliary Task Selection Module

After obtaining the rankings of candidate auxiliary tasks for each primary task, we finalize the auxiliary task selection process through trial experiments. We also study the potential of GradTS to subsample the selected auxiliary tasks. Our experiments show that with one additional fine-tuning pass of its backend model on the individual tasks, GradTS produces subsampled auxiliary training sets higher in quality than the task-level selections.

We introduce the two settings of GradTS to select tasks from the task correlations as follows:

[Task-level Trial-based] We select auxiliary tasks greedily under this setting. Starting from the most closely-correlated task to a primary task, we keep adding tasks to the auxiliary task set and run MTL evaluations on the primary task and all the chosen auxiliary tasks. GradTS stops adding new tasks when the evaluation score starts to decrease on the validation set we leave for parameter tuning and finalizes the auxiliary task set with the tasks chosen at the previous step.

[Instance-level] We re-run the base model of GradTS on all the individual tasks once, with gradient calculation but not parameter updates. For each instance, we take the absolute value of its gradients on all the attention heads, layer-wise normalize the gradients, and scale the numbers to the range [0, 1]. Then we calculate and record the correlation score between the normalized gradient matrix and the head ranking matrix of each candidate auxiliary task. Last, we use a threshold to select auxiliary training instances from tasks chosen by the task-level trial-based method to form a subsampled aux-

[6] Our preliminary experiments show that fine-tuning the backend model for three to seven epochs at the warm up stage does not have much effect on the predictions of GradTS.

A Unified Representation Learning Strategy for Open Relation Extraction with Ranked List Loss

Renze Lou[1]*, **Fan Zhang**[2]*, **Xiaowei Zhou**[3], **Yutong Wang**[4], **Minghui Wu**[1] and **Lin Sun**[1]†
[1]Department of Computer Science, Zhejiang University City College, Hangzhou, China
[2]Faculty of Economics, Hitotsubashi University, Tokyo, Japan
[3]Zhejiang Qianyue Digital Technology Co., Ltd, China
[4]Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, China
marionojump0722@gmail.com;zhangoutstanding@hotmail.com;
tinycs@126.com;wangyt19@mails.tsinghua.edu.cn;{mhwu,sunl}@zucc.edu.cn

Abstract

Open Relation Extraction (OpenRE), aiming to extract relational facts from open-domain corpora, is a sub-task of Relation Extraction and a crucial upstream process for many other NLP tasks. However, various previous clustering-based OpenRE strategies either confine themselves to unsupervised paradigms or can not directly build a unified relational semantic space, hence impacting down-stream clustering. In this paper, we propose a novel supervised learning framework named MORE-RLL (**M**etric learning-based **O**pen **R**elation **E**xtraction with **R**anked **L**ist **L**oss) to construct a semantic metric space by utilizing Ranked List Loss to discover new relational facts. Experiments on real-world datasets show that MORE-RLL can achieve excellent performance compared with previous state-of-the-art methods, demonstrating the capability of MORE-RLL in unified semantic representation learning and novel relational fact detection.

1 Introduction

Relation Extraction (RE) aims to extract pre-defined relational facts from plain text (e.g., *"Mary gave birth to Keller in 1989s."*, RE can extract ***"gave_birth_to"*** between two named entities *"Mary"* and *"Keller"*). It is an important task that can structure a large amount of text data. Therefore, it can benefit for unstructured text data storing and the procedure of many other down-stream NLP tasks or applications, such as knowledge graph construction (Suchanek et al., 2007), information retrieval (Xiong et al., 2017), and logic reasoning (Socher et al., 2013). Nevertheless, with the rapid development of social media and human civilization, novel relationships and new knowledge in open-domain text data are also increasing. Accordingly, the relation types in the open-domain corpora may not be pre-defined, which is hard for RE to handle. To meet the rapid emergence of such novel knowledge, OpenRE emerged as the times required (Banko et al., 2007). The goal of OpenRE is to detect novel relational facts from open-domain datasets. It is a crucial task for updating the human knowledge base and the study of human civilization.

Existing OpenRE methods are divided into two main categories: tagging-based and clustering-based. The tagging-based strategies treat OpenRE as a sequence labeling problem (Banko et al., 2007; Banko and Etzioni, 2008). Still, these methods often extract surface forms that can not be utilized for down-stream tasks (i.e., some sequences have the same relational semantic type, but their phrases generated from tagging-based methods are different because of overly-specific). Comparatively, clustering-based methods aim to identify the rich semantic features in the text, then cluster them into certain relation types. Recently, many efforts have been devoted to exploring clustering-based methods, such as (Yao et al., 2012; Elsahar et al., 2017). Yet, those schemes are laborious and time-consuming because of the high dependence on well-designed features created by hand.

Profited from the substantial improvement of computing power in recent years, neural networks begin to be exploited in clustering-based OpenRE tasks to alleviate the above issues, such as (Simon et al., 2019; Hu et al., 2020; Gao et al., 2020). Even so, these strategies confine themselves to unsupervised

[*] Equal contribution
[†] Corresponding author

©2021 China National Conference on Computational Linguistics
Published under Creative Commons Attribution 4.0 International License

or self-supervised paradigms and can not fully benefit from current high-quality human-labeled corpora. Although several unconventional works have gained phenomenal performance, such as (Zhang et al., 2021), the reliance on the extra knowledge for these strategies make it hard for us to compare with. Besides, another supervised scheme learned the similarity metrics from labeled instances and further transferred the relational knowledge to the open-domain scene, namely Relational Siamese Networks (RSNs) (Wu et al., 2019). However, RSNs target learning a similarity classifier rather than building relational representations directly. Thereby, this may impact the efficiency and effect of down-stream clustering.

In order to address these issues, we propose a novel supervised learning framework via a clustering-based scheme driving neural encoder to build rich semantic representation directly. From our insight view, the essential target of the clustering-based OpenRE algorithm is to construct a reasonable semantic space on the open-domain corpora, where all different relational facts can be distinguished clearly. Therefore, the learning of semantic representation is a fundamental part of the whole task. It can not only extend the functionality of the neural encoder (i.e., the semantic space construction ability of the neural model can be used in other scenes, such as classification, etc.) but also bring benefits to downstream clustering.

As a result, we pay attention to the unified semantic representation learning ability of neural encoders. Specifically, we employ deep metric learning to drive the neural encoder to build a distinguishable semantic space on open-domain datasets. However, most prevailing deep metric learning methods, such as triplet loss (Hoffer and Ailon, 2015), N-pair-mc (Sohn, 2016), or Proxy-NCA (Movshovitz-Attias et al., 2017), always bring low yield due to the poor supervision signals from the limited number of training data points. Inspired by (Wang et al., 2019), we chose Ranked List Loss (RLL) instead, which can capture set-based rich supervision signals. Meanwhile, RLL can preserve a better intraclass similarity structure within a hypersphere than other set-based schemes, hence constructing a more desirable semantic space.

Additionally, considering that the open scene corpora is usually full of noise, hence directly transferring knowledge may not be an ideal choice. To enhance the model's robustness, we also design virtual adversarial training for our semantic space construction algorithm. Experiments demonstrate that MORE-RLL can build more distinguishable semantic representations and obtain excellent performances on real-world datasets.

To sum up, the main contributions of this work are as follows:

- We propose a novel clustering-based OpenRE framework, namely MORE-RLL. The MORE-RLL combines deep metric learning and neural encoder to build a unified relational semantic space to discriminate samples rather than utilize an additional classification layer. Thus, it can handle the enormous undefined relation types in the open-domain corpora and facilitate down-stream clustering to discover valuable novel types. Meanwhile, we adopt a Ranked List Loss to gain more prosperous supervision signals and construct a more desirable semantic space than other prevailing metric learning losses.
- Considering the noise and bias present in the text of open scenarios, we also design virtual adversarial training to enhance the robustness of MORE-RLL instead of directly transferring the knowledge that comes from clean RE datasets to the open-domain corpora.
- Experiments illustrate that the proposed MORE-RLL achieves state-of-the-art performance on real-world datasets, even if the imbalance distribution presents in the test set. Moreover, the visual analysis also demonstrates its excellent ability of relational representation learning and novel knowledge detecting.

2 Methodology

In this section, we will introduce our framework in detail. As shown in Figure 1, we exploit a neural encoder to extract relational representations from a batch of training samples. These sentence-level representations can be taken as relational semantic vectors, indicating the relative locations of facts in the semantic feature space. Then, we use them to calculate the Ranked List Loss (RLL) and gain rich