Improving Relation Extraction with Relational Paraphrase Sentences

Junjie Yu¹, Tong Zhu¹, Wenliang Chen¹, Wei Zhang², Min Zhang¹

Institute of Artificial Intelligence, School of Computer Science and Technology,

Soochow University, China¹

Alibaba Group, China²

{jjyu,tzhu7}@stu.suda.edu.cn, {wlchen,minzhang}@suda.edu.cn, lantu.zw@alibaba-inc.com

Abstract

Supervised models for Relation Extraction (RE) typically require human-annotated training data. Due to the limited size, the human-annotated data is usually incapable of covering diverse relation expressions, which could limit the performance of RE. To increase the coverage of relation expressions, we may enlarge the labeled data by hiring annotators or applying Distant Supervision (DS). However, the human-annotated data is costly and non-scalable while the distantly supervised data contains many noises. In this paper, we propose an alternative approach to improve RE systems via enriching diverse expressions by relational paraphrase sentences. Based on an existing labeled data, we first automatically build a task-specific paraphrase data. Then, we propose a novel model to learn the information of diverse relation expressions. In our model, we try to capture this information on the paraphrases via a joint learning framework. Finally, we conduct experiments on a widely used dataset and the experimental results show that our approach is effective to improve the performance on relation extraction, even compared with a strong baseline.

1 Introduction

Relation Extraction (RE) is an important task in Information Extraction, which identifies semantic relations between entities in text (Zelenko et al., 2003; Zhou et al., 2005; Mintz et al., 2009). The task becomes a typical classification problem if an entity pair in a text is given. In recent years, supervised models have achieved great progress on this task with the help of a massive amount of manually annotated high-quality data (Zeng et al., 2014; dos Santos et al., 2015; Miwa and Bansal, 2016; Zhang et al., 2017).

However, diverse expressions for a same semantic relation are difficult to be fully covered by humanannotated data. For example, sentences (1) "*Steve Jobs* co-founded *Apple Computer*."; (2) "*Steve Jobs* was the co-founder of *Apple Computer*."; and (3) "*Steve Jobs* started *Apple Computer* with Wozniak." express the same semantic relation between person "*Steve Jobs*" and company "*Apple Computer*" in different wording. Generally, it is difficult for the supervised model trained on sentence (1) and (2) to recognize the semantic relation in sentence (3).

To solve the above challenge, we can use two possible solutions. The first solution is to hire annotators to label more data. While the human-annotated data is reliable, it is costly and non-scalable, with regard to both time and money. The second one is to adopt the Distant Supervision (DS) mechanism to automatically build a large-scale labeled data (Mintz et al., 2009). However, with the strong assumption that all sentences containing two given entities in a relation triple express the same relation, DS may result in the severe wrong labeling problem. In this paper, we use an alternative solution that uses a paraphrase data which collects sentences conveying the same meaning in different wording. In the literature, there exist many paraphrase datasets, such as Simple Wikipedia (Kauchak, 2013), Twitter URL corpus (Lan et al., 2017), and Para-NMT (Wieting and Gimpel, 2018). However, these general paraphrase datasets do

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Licence details: http://creativecommons.org/licenses/by/4.0/.



Figure 1: Framework of building the relational paraphrase data. EN=English, CN=Chinese.



Figure 2: An example from our ReP data. #1 is a human-annotated sentence, and #2-4 are paraphrase sentences. Blue words with underlines mean different clues for relation "org:founded_by" between two entities.

not have explicit clues for entities and relations. Our preliminary experimental results show that using such paraphrase datasets harms the performance of relation extraction. Therefore, it is difficult to learn useful information for relation extraction from the general paraphrase data.

In this paper, we propose to automatically build a task-specific paraphrase data which has the explicit clues instead of using general paraphrase datasets for relation extraction. Motivated by the recent success of deep neural networks in machine translation (Luong et al., 2015; Wu et al., 2016; Vaswani et al., 2017), we adopt more than one Neural Machine Translation (NMT) systems to generate possible paraphrases via back-translation for each sentence in an existing RE data. The back-translation is the procedure in which a system translates a sentence into another language, and then translates back to the original language. However, we can not convey the annotations of entities during back-translation since word alignment information is unavailable. To solve this problem, we design a contextual similarity based method to align entities between the human-annotated sentences and corresponding paraphrase sentences. We combine the human-annotated sentences with these paraphrase sentences as our new training data, named as Relational Paraphrase (ReP) Data.

Then, we propose a joint learning framework to train a relation extractor on the ReP data. Though back-translation is a convenient way for us to generate paraphrase sentences, there is some noise due to the wrong translation. In order to reduce the effect of the noise in the ReP data, we propose a multiinstance learning module to model multiple paraphrase sentences. To build a strong baseline, we choose BERT's fine-tuning mechanism to encode sentences and train the relation extractor (Devlin et al., 2018).

In summary, we make following contributions:

- We build a Relational Paraphrase (ReP) data explicitly expressing the information of entities and their relations. In the ReP data, there are 204,372 paraphrase sentences and 68,124 human-annotated sentences.
- We propose a joint learning framework to train a new relation extractor on the ReP data. To reduce the effect of the noise, we propose several multi-instance learning strategies to model paraphrase sentences.

To the best of our knowledge, the ReP data is the first task-specific paraphrase data for RE. Experimental results on a widely used RE data show that our approach can effectively improve the performance as compared with the strong baseline.

2 Relational Paraphrase Data

In this section, we describe how to build the Relational Paraphrase (ReP) data, which is a task-specific paraphrase data for RE. As shown in Figure 1, we build the ReP data by generating paraphrase sentences for human-annotated sentences from an existing RE data. In this way, our ReP data contains two parts: ReP-GOLD and ReP-AUTO. ReP-GOLD is the original training set of the existing RE data and ReP-AUTO is the auto-generated paraphrase data. An example from the ReP data is shown in Figure 2.

Data Split	Train				Dev	Test
Data Split	# Sen	# Sen1	# Fact	# Sen1/Fact	# Sen	# Sen
Gold-Annotated (TACRED)	68,124	13,012	8,190	1.6	22,631	15,509
Auto-Generated	204,372	39,036	8,190	4.8	-	-

Table 1: Statistics of the RE data used in the experiments. # Sen: number of all sentences. # Sen1: number of sentences excluding sentences labeled with no_relation. # Fact: number of relation facts (excluding no_relation). # Sen1/Fact: average number of supporting sentences for each relation fact.

2.1 Human Annotated Data

In this paper, we take a widely used relation extraction data: TACRED (Zhang et al., 2017), which contains about 105k sentences in total. There are 41 pre-defined relation types (e.g., "person:city_of_birth", "organization:founded_by") and a special type no_relation. In each sentence, two entities and one relation are labeled by human. The statistics of TACRED are shown in Table 1, where TACRED contains the training, development, and test sets. From the table, we can find that although the training set contains more than 60k sentences, the number of sentences with meaningful relation types (not no_relation) is small (about 13k). And the average number of sentences for each relation fact (an individual triple <head entity, tail entity, relation type>) is less than 2. Hence, generating paraphrase expressions for each labeled sentence is expected to enrich the annotated data.

2.2 Generating Relational Paraphrase Sentences

We introduce Neural Machine Translation (NMT) technology with back-translation to help us automatically generate possible paraphrase sentences. Back-translation is an effective method to augment the parallel training corpus in NMT (Fadaee et al., 2017; Edunov et al., 2018). In this procedure, there are two challenges: (1) How to guarantee the variety of paraphrase sentences; (2) How to label the entities in paraphrase sentences. For the first challenge, we view each NMT system as an individual knowledge base which translates a sentence in its own way. Hence, we take more than one public NMT systems to perform back-translation on the training set of TACRED. As the NMT systems provide end-to-end translations, entities in sentences may be replaced by other words after back-translation. As shown in Figure 2, the head entity "all basotho convention" has been translated into "wholly basotho", "all basoto conference", and "all basoto congress" by three NMT systems, respectively. Thus, we propose to do entity alignment for the second challenge. To do entity alignment, there are two possible solutions. One is to do preprocessing on input sentences before translation. Another is to do postprocessing on translated sentences.

We tried the preprocessing solution where two tags (#ENTITY1# and #ENTITY2#) are used to replace the entities in an input sentence before back-translation. We expected that the tags would be kept unchanged during back-translation. However, this method did not work well since the tags are often changed. Moreover, the meaning of the sentence is changed in some degree after replacing the entities by the tags that affects the performance of NMT systems. In our solution, we perform back-translation on the original sentences and then propose a contextual similarity based method to conduct entity alignment.

2.2.1 Back-Translation

The back-translation is a procedure that first translates a sentence from a source language into a target language, then translates it back to the source language. In this paper, we use English (EN) as the source language and Chinese (CN) as the target language.

To perform back-translation, we adopt three public NMT systems: Google Translation¹, Baidu Translation² and Xiaoniu Translation³. We use the online service of these three NMT systems to do backtranslation for sentences in TACRED. As a result, we can obtain three paraphrase sentences for one

¹https://translate.google.com/

²https://fanyi.baidu.com/

³http://www.niutrans.com/



Para.	Acc.	78.0%			
	Diff.	Yes	No		
Entity	Prop.	47.4%	52.6%		
	Acc.	89.2%	100.0%		
Both	Acc.	74.0%			

Table 2: Manual evaluation of the ReP-AUTO. **Para.**: correct paraphrase. Acc.: accuracy. **Entity**: performance of entity alignment. **Prop.**: proportion. **Diff.**: whether entities have been changed.

Figure 3: An example of aligning entities and relations.

human-annotated sentence.

2.2.2 Entity Alignment

In this paper, entity alignment is defined as aligning entities between the source human-annotated sentences and target paraphrase sentences. Intuitively, pattern matching is the simplest postprocessing way which searches in the paraphrase sentences for the entity words. But it fails to tackle with the situation that entities are replaced by synonyms after back-translation. To solve the above problem, we propose a contextual similarity based method to align the entities.

Suppose that we have a human-annotated sentence (source) s and its corresponding paraphrase sentence (target) t. We first use a pretrained BERT as an encoder to output the representations of s and t, \mathbf{h}^s and \mathbf{h}^t , respectively. Then, we map words between s and t by calculating cosine based similarity. Formally, for i_{th} word in t, we get the mapped word s^{t_i} in s with the highest cosine score,

$$s^{t_i} = \underset{s_j \in s}{\operatorname{argmax}} \{ \cos(\mathbf{h}_i^t, \mathbf{h}_j^s) \}.$$
⁽¹⁾

After obtaining the words that mapped with entities in human-annotated sentence s, we greedly keep sequential mapped words in t as aligned entities.

An example of entity alignment is shown in Figure 3. Taking head entity "all basotho convention" in annotated sentence for explanation, the mapped words in the paraphrase sentence are "wholly", "basotho", and "who", respectively. Based on the mapping information, we recognize the words "wholly basotho" as head entity in paraphrase sentence. The word "who" in paraphrase sentence which maps with the word "convention" is deprecated because it is not conjoined with its previous mapped word ("basotho").

2.3 Statistics of the ReP Data

Using three different NMT systems, we obtain three possible paraphrase sentences for each sentence in the training set of TACRED. Please note that we do not generate paraphrases for the sentences in the development set and test set. The statistics are shown in Table 1. In total, the ReP data contains 68,124 human-annotated sentences as ReP-GOLD (which is the original training set of TACRED) and 204,372 paraphrase sentences as ReP-AUTO.

To evaluate the quality of these auto-generated sentences in ReP-AUTO, we randomly select 100 sentences to take a manual evaluation. The evaluation results are shown in Table 2. First, we check whether the candidate paraphrase sentence is a correct paraphrase expression of the original sentence. The results show that 78% sentences can be regarded as correct paraphrases and others are errors. Second, we check the performance of entity alignment on these paraphrase sentences (that is, 78% of all). Results show that nearly half (47.4%) of the paraphrase sentences have changed their wording of entities which can explain why pattern matching does not work well. The accuracy of entity alignment is 89.2% for the changed examples while the accuracy is 100% for unchanged examples. In total, 74.0% of the sentences in ReP-AUTO are paraphrase expressions with proper annotations of entities and relations. How to reduce the effect of noises becomes a challenge when we build our relation extraction system.



Figure 4: Training Framework

Figure 6: Multi-Instance Learning

3 Our Approach

In this section, we describe our relation extraction system in detail. To train on both ReP-GOLD and ReP-AUTO, we propose a joint learning framework. During training, each human-annotated sentence in ReP-GOLD is provided with three corresponding paraphrase sentences from ReP-AUTO and we put the four sentences into one input unit. As shown in Figure 4, there are three key components in our system: (1) A sentence encoder, which encodes sentences into distributed representations; (2) A multi-instance learning module, which models three paraphrase sentences from one input unit into a mixed distributed representation; (3) A relation extractor, where the input representations are classified into different relations.

3.1 BERT-based Sentence Encoder

When using the pretrained model BERT, the fine-tuning based approach gives impressive performance in many tasks (Devlin et al., 2018). Following Soares et al. (2019), we augment each sentence with four reserved tokens ([E1], [/E1], [E2] and [/E2]) to mark the beginning and ending of subject entity and object entity. Then, we concatenate the outputs of [E1] and [E2] as relational representation. Figure 5 illustrates the input and output of the sentence encoder used in this paper.

Formally, we first build the input sentence as x:

$$x = [\text{CLS}] \ x_0 \dots [\text{E1}] \ x_i \dots x_j \ [/\text{E1}] \dots [\text{E2}] \ x_k \dots x_l \ [/\text{E2}] \dots x_n \ [\text{SEP}], \tag{2}$$

where [CLS] and [SEP] are two special tokens used in BERT, and $x_i \dots x_j$, $x_k \dots x_l$ are words of two entities, respectively. Let $f(\theta)$ be the BERT-based sentence encoder used in our model, where θ represents parameters. We output a relational representation of the sentence x by concatenating $\mathbf{h}_{[E1]} \in \mathbb{R}^d$ and $\mathbf{h}_{[E2]} \in \mathbb{R}^d$. That is, the relational representation **s** of x is:

$$\mathbf{s}_{x} = \mathbf{h}_{[E1]} \oplus \mathbf{h}_{[E2]}, \ \mathbf{h}_{[E1]} = f(x|\theta, [E1]), \ \mathbf{h}_{[E2]} = f(x|\theta, [E2]),$$
(3)

where $\mathbf{s}_x \in \mathbb{R}^{2d}$ and d is the size of token representation.

3.2 Multi-Instance Learning

To make full use of the paraphrase sentences in ReP-AUTO to relieve the noise problem, we adopt the idea of multi-instance learning for modeling paraphrase sentences (Riedel et al., 2010). In this way,

we first put three paraphrase sentences from one input unit into one bag and then output the bag-level representations. Formally, each bag B contains three sentences:

$$B = \{x^{google}, x^{baidu}, x^{xiaoniu}\},\tag{4}$$

where x^* means a paraphrase sentence from one of the three NMT systems and it is constructed in the same way described in Equation (2). We can get the sentence-level representation s_x for each sentence x in B by Equation (3). Then, we apply the following strategies to obtain bag-level representations.

Pre-Select. In this method, we only use paraphrase sentences generated by one selected NMT system. In this way, the representation of a bag is the representation of one sentence. Thus, we have three choices: Google, Baidu, and Xiaoniu.

Bag-Max. In this method, we generate the bag-level representations by performing maximum pooling on outputs of sentences in bag *B*:

$$\mathbf{s}_B = \underset{x \in B}{maxpool}(\mathbf{s}_x). \tag{5}$$

Bag-One. Different from outputting a maximum value on each dimension in Bag-Max method, Bag-One outputs the best representation from one of three sentences in *B* by calculating the probability on its gold relation type after a softmax layer.

$$\mathbf{s}_{B} = \mathbf{s}_{x'}, \tag{6}$$
$$x' = \operatorname*{argmax}_{P} p(r_{x} | x, \theta),$$

where p() outputs the probability of relation type r_x for the input sentence x under current model parameters θ .

Bag-Avg. Similar to Bag-Max, Bag-Avg method adds an averaged pooling layer after encoding sentences in *B*:

$$\mathbf{s}_B = \frac{1}{|B|} \sum_{x \in B} \mathbf{s}_x. \tag{7}$$

Bag-Att. Inspired by the attention mechanism used in Lin et al. (2016), we add an attention layer to output bag-level representations for sentences in B. First we generate attention weights α for sentences in B by calculating how well it matches with their gold relation type. Then, we output a weighted sum of representations:

$$\mathbf{s}_{B} = \sum_{x \in B} \alpha_{x} \mathbf{s}_{x},$$

$$\alpha_{x} = \frac{exp(e_{x})}{\sum_{x' \in B} exp(e_{x'})},$$

$$e_{x} = \mathbf{s}_{x} \mathbf{Ar},$$
(8)

where e_x measures how well \mathbf{s}_x matches with the query vector $\mathbf{r} \in \mathbb{R}^{2d}$ which is the representation of the gold relation of x, and $\mathbf{A} \in \mathbb{R}^{2d \times 2d}$ represents a diagonal matrix.

3.3 Relation Extractor

After obtaining the relational representations, we build the relation extractor for final classification. In this paper, we choose the fine-tuning strategy of BERT, where we add a fully connected linear layer with a softmax function layer on BERT.

In the baseline system, we use the original training set of TACRED (named as ReP-GOLD in this paper) to train the model. Hence, the probability distribution for the input sentence x is:

$$p(x,\theta) = Softmax(\mathbf{W}^T \mathbf{s}_x + \mathbf{b}), \tag{9}$$

where the matrix $\mathbf{W} \in \mathbb{R}^{2d \times d^r}$ and bias vector $\mathbf{b} \in \mathbb{R}^{d^r}$ are model parameters in which d^r is the number of pre-defined relation types. Then, we use the standard cross-entropy function to calculate the loss on the ReP-GOLD:

$$loss_{gold} = -\sum_{x \in \text{ReP-GOLD}} log(p(r_x | x, \theta)),$$
(10)

where r_x is the gold relation for the input sentence x.

In our proposed approach, we take the ReP data as input, which includes ReP-GOLD and ReP-AUTO. Under the joint learning framework, the two data sets are processed in two different routes, respectively. However, the sentence encoder and relation extractor are shared by the two routes. For the ReP-GOLD, we use the same procedure as baseline system. For ReP-AUTO, we take the multi-instance learning methods (described in Section 3.2) to output the bag-level representation s_B for sentences in *B*. Firstly, the probability distribution for bag *B* is:

$$p(B,\theta) = Softmax(\mathbf{W}^T \mathbf{s}_B + \mathbf{b}).$$
(11)

Then, the loss on ReP-AUTO is:

$$loss_{auto} = -\sum_{B \in \text{ReP-AUTO}} log(p(r_B|B, \theta)),$$
(12)

where r_B is the gold relation for the sentences in bag B.

To jointly train on both ReP-GOLD and ReP-AUTO, we take a weighted sum of two losses as the final loss function:

$$loss = loss_{gold} + \lambda \ loss_{auto},\tag{13}$$

where λ is a hyper-parameter.

3.4 Training and Testing

To solve the optimization problem, we adopt Adam to minimize the objective function. During training, we train the relation extractor on the ReP data including ReP-GOLD and ReP-AUTO. In the testing phase, to simulate the scenario of real applications, we directly perform relation extraction by Equation (9) on input sentences, which means no extra paraphrase sentences are required.

4 **Experiments**

4.1 Experimental Settings

Datasets and evaluation. In our experiments, we use TACRED and the newly built ReP data. For the baseline system, we use ReP-GOLD (training set of TACRED) as training data. For our approaches, we use the ReP data as training data. We use the development set and test set in TACRED to do evaluation. Statistics of TACRED and the ReP data are described in Table 1. Following previous studies (Zhang et al., 2017; Zhang et al., 2018; Soares et al., 2019), we report micro average F1 scores. We run 3 times with random seeds to initialize the model and report the average results.

Hyperparapmeters After tuning the hyperparameters on the development set, we choose the following settings: batch size is 16, learning rate is 3e-5 with Adam, and training epoch is [1-5]. We use PyTorch (Paszke et al., 2019) as our machine learning library and the architecture of BERT from Wolf et al. (2019). Two versions of pretrained BERT models (Devlin et al., 2018), $BERT_{base}$ and $BERT_{large}$ are used in this paper.

Tuning of λ . The hyperparapmeter λ described in Equation (13) is used to combine two losses trained on ReP-GOLD and ReP-AUTO. For simplicity, we tune λ on Pre-Select system with Google Translation (ReP-GOLD + Google) in the range of [0.1, 0.2, 0.4, 0.6, 0.8, 1.0] on the development set. As shown in Table 3, the system with $\lambda = 0.2$ achieves the best performance. Hence, we set λ as 0.2 in the following experiments without an explicit statement.

λ	0.0	0.1	0.2	0.4	0.6	0.8	1.0
F1	68.94	69.22	69.31	69.25	69.02	68.76	68.53

Table 3: Results of tuning with λ on the development set. $\lambda = 0.0$ equals to the baseline system.

		Systems
Systems		CNN-PE [†] (Zeng et al., 2014)
Deseling (DeD COLD)	1 ⁻¹	PCNN ^{\dagger} (Zeng et al., 2015)
Baseline (ReP-GOLD)	08.07	SDP-LSTM \ddagger (Xu et al., 2015)
Rep-AUTO	66.75	Tree-LSTM [‡] (Tai et al., 2015)
ReP-GOLD U ReP-AUTO	68.53	PA-LSTM (Zhang et al., 2017)
ReP-GOLD + Google	69.37	SA-LSTM+D (Yu et al. 2019)
ReP-GOLD + Baidu	69.12	C-GCN + PA-I STM (Zhang et al. 2018)
ReP-GOLD + Xiaoniu	69.24	$MTB \text{ on } BERT, \qquad (Soares et al. 2010)$
ReP-GOLD + Bag-Max	69.45	Resoling on <i>BEPT</i> .
ReP-GOLD + Bag-One	69.46	Daseline on DEnt base
ReP-GOLD + Bag-Avg	69.60	Rer-GOLD + Dag-Avg on BERI base
ReP-GOLD + Bag-Att	69.38	Baseline on <i>BERT</i> _{large}
itter colle : bug itte		ReP-GOLD + Bag-Avg on $BERT_{large}$

Table 4: Comparison with Baseline on test set.

Table 5: Comparion with previous results. † marks results reported in Yu et al. (2019); ‡ marks results reported in Zhang et al. (2017).

F1

4.2 Experimental Results

Main results. As shown in Table 4, we compare our systems with the baseline system, where all the systems use $BERT_{base}$. The baseline system is trained on ReP-GOLD (the original training set of TACRED). For simplicity, we classify our approaches into three groups: (1) Merging; (2) Joint learning with single paraphrase; (3) Joint learning with multiple paraphrases. Firstly, we find that directly using paraphrase sentences (ReP-AUTO) performs worse than the Baseline. The reason might be that the noise in the ReP-AUTO harms the performance. Further, ReP-GOLD U ReP-AUTO (directly merging two data sets) also performs a little worse than the Baseline. Secondly, we find that the performance can be improved after adding the ReP-AUTO under the joint learning framework even using a single NMT system to generate paraphrase sentences. Thirdly, applying multi-instance learning methods (ReP-GOLD + Bag-Max/Bag-One/Bag-Avg/Bag-Att) on paraphrase sentences from more than one NMT systems can further improve the performance. In total, ReP-GOLD + Bag-Avg yields the best performance among all the systems. This indicates that our proposed approach can improve the performance of relation extraction. We use the system with Bag-Avg as our final system in the following experiments.

Comparision with previous approaches. We further compare with several RE systems proposed in the previous studies, as shown in Table 5. From the table, We find that "Baseline on $BERT_{base}$ " achieves an impressive performance which outperforms most of the previous studies. Fine-tuning on $BERT_{large}$ is expected to further improve the performance of both baseline system and our system. We set $\lambda = 0.4$ for "ReP-GOLD + Bag-Avg on $BERT_{large}$ " as it achieves the best performance on the development set. We find that our system achieves a better performance than baseline system no matter under $BERT_{base}$ or $BERT_{large}$. The results indicate that our approach using paraphrase sentences to learn from diverse expressions can yield comparable performance with MTB on $BERT_{large}$, which achieves the best reported score on TACRED.

5 **Analysis and Discussion**

Here, we study the effectiveness of our system in different situations. Thus, we compare the outputs of ours (ReP-GOLD+Bag-Avg) with Baseline on $BERT_{base}$ on the test set.

The results are shown in Table 6, where we exclude the sentences labeled with no_relation in test

	Number of Sentences		Sentence	e Length	Entity Distance	
Test Set	Small Set	Large Set	Short Set	Long Set	Short Set	Long Set
Amount	1,746	1,579	1,742	1,583	1,686	1,639
Baseline	71.60	83.21	82.71	71.04	87.36	65.35
Ours	72.97	83.68	83.45	72.12	87.75	66.90
Improvement	+1.37	+0.47	+0.74	+1.08	+0.39	+1.55

Table 6: Analysis results on the test set (excluding sentences with no_relation).

set. (1) Performance by Number of Sentences. We first sort the relations by the number of sentences they have in the training set. Then, we split the test set into two approximately equal sets according to the sorted relations, a Small set and a Large set. From the table, we find that the improvement gap is larger on the Small set than that on the Large set (+1.37 VS +0.47). The reason might be that the relations in the Small set have less training sentences that results in lack of diverse expressions and our relational paraphrase sentences can provide more help for them. (2) Performance by Sentence Length. We sort the relations by the average length of sentences they have in the training set. Then, we also split the test set into two approximately equal sets according to the sorted relations, a Short set and a Long set. We find that our system is not sensitive to the sentence length (+0.74 VS +1.08). (3) Performance by Entity Distance. Entity Distance means the number of words between two entities in a sentence. We sort the relations by the average entity distance of sentences they have in the training set. Then, we also split the test set into two approximately equal sets according to the sorted relations, a Short set and a Long set. We find that our system achieves the more significant improvement (+1.55) on the Long set than the Short set (+0.39). The reason might be that the NMT systems have more chances to generate different expressions of relations for the sentences with longer entity distance.

6 Related Work

Recent years, with the development of neural networks, the neural models for relation extraction attract many researchers spending time on improving the performance. Zeng et al. (2014) proposes a CNN model with position features. Nguyen and Grishman (2015) improves the CNN model with multiple window sizes. Margin based ranking loss is used instead of cross-entropy in dos Santos et al. (2015). There are also many researchers focusing on other neural networks like RNN (Zhang and Wang, 2015), LSTM (Xu et al., 2015; Tai et al., 2015; Miwa and Bansal, 2016) and GCN (Zhang et al., 2018). Recently, transfer learning from pre-trained model like BERT to downstream supervised tasks is popular. For relation extraction, the main challenge of applying BERT is how to model the input sentences in an entity-aware way. Wang et al. (2019) adds relative position features in a self-attention layer. Soares et al. (2019) directly inserts four reserved tags in sentences to represent borders of entities. We also build our system based on BERT, which is a very strong baseline.

In addition to the development of models for sentence encoding, studies on relieving the dependence on human-annotated data are also popular. Distant supervision is proposed by Mintz et al. (2009) to automatically build labeled data for RE. Although a lot of approaches have been proposed to relieve the wrong labeling problem in distant supervised data (Takamatsu et al., 2012; Lin et al., 2016; He et al., 2020), there is a gap between models that trained on supervised data and distant supervision data. Using some carefully selected human-annotated examples as partial supervision, Angeli et al. (2014) combines the reliability from human-annotated data and the large coverage from distant supervision data. Based on the directionality of relations, Xu et al. (2016) proposes a data augmentation method to alleviate the sparse problem. Vashishth et al. (2018) generates aliases for relation names via phrase-level paraphrases. Beltagy et al. (2019) proposes to combine the distant supervision data with an existing human-annotated RE data. None of the above studies has used the paraphrase sentences. In this paper, we propose to enlarge the coverage of relation expressions by building a relational paraphrase data for an existing RE data.

7 Conclusion

In this paper, we show that using the newly built task-specific paraphrase data can have a substantial effect on the performance of relation extraction. In particular, we demonstrate that our proposed system consistently outperforms the strong baseline system using BERT. The gains we find come not only from the joint learning framework, but also from the multi-instance learning strategies which model the paraphrase sentences at bag level. Our code and data resources are available at https://github.com/jjyunlp/ReP-RE.

Acknowledgements

The research work is supported by the National Natural Science Foundation of China (Grant Nos. 61876115 and 61936010). This work is partially supported by the joint research project of Alibaba and Soochow University. Corresponding author is Wenliang Chen. We also thank Yanxia Qin for her helpful suggestions, and the anonymous reviewers for their constructive comments.

References

- Gabor Angeli, Julie Tibshirani, Jean Wu, and Christopher D Manning. 2014. Combining distant and partial supervision for relation extraction. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1556–1567.
- Iz Beltagy, Kyle Lo, and Waleed Ammar. 2019. Combining distant and direct supervision for neural relation extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1858–1867.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. In *Proceedings of the 53rd ACL*, volume 1, pages 626–634.
- Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500.
- Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (*Volume 2: Short Papers*), pages 567–573.
- Zhengqiu He, Wenliang Chen, Yuyi Wang, Wei Zhang, Guanchun Wang, and Min Zhang. 2020. Improving Neural Relation Extraction with Positive and Unlabeled Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7927–7934, April.
- David Kauchak. 2013. Improving text simplification language modeling using unsimplified text data. In *Proceedings of the 51st ACL*), pages 1537–1546, Sofia, Bulgaria, August. ACL.
- Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. 2017. A continuously growing dataset of sentential paraphrases. In *Proceedings of the 2017 EMNLP*, pages 1224–1234, Copenhagen, Denmark, September. ACL.
- Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th ACL*, pages 2124–2133.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 EMNLP*, pages 1412–1421, Lisbon, Portugal, September. ACL.
- Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the 47th AFNLP*, pages 1003–1011. ACL.
- Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of the 54th ACL*, volume 1, pages 1105–1116.
- Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, pages 39–48.

- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *NIPS*, pages 8024–8035.
- Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.
- Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. *arXiv preprint arXiv:1906.03158*.
- Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd ACL*.
- Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. 2012. Reducing Wrong Labels in Distant Supervision for Relation Extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 721–729, Jeju Island, Korea, July. Association for Computational Linguistics.
- Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha Talukdar. 2018. RE-SIDE: Improving Distantly-Supervised Neural Relation Extraction using Side Information. In *Proceedings* of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1257–1266, Brussels, Belgium. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NPIS*, pages 5998–6008.
- Haoyu Wang, Ming Tan, Mo Yu, Shiyu Chang, Dakuo Wang, Kun Xu, Xiaoxiao Guo, and Saloni Potdar. 2019. Extracting multiple-relations in one-pass with pre-trained transformers. *arXiv preprint arXiv:1902.01030*.
- John Wieting and Kevin Gimpel. 2018. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th ACL*, pages 451–462, Melbourne, Australia, July. ACL.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-theart natural language processing. *ArXiv*, abs/1910.03771.
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.
- Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *proceedings of the 2015 EMNLP*, pages 1785–1794.
- Yan Xu, Ran Jia, Lili Mou, Ge Li, Yunchuan Chen, Yangyang Lu, and Zhi Jin. 2016. Improved relation classification by deep recurrent neural networks with data augmentation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1461–1470.
- Bowen Yu, Zhenyu Zhang, Tingwen Liu, Bin Wang, Sujian Li, and Quangang Li. 2019. Beyond word attention: using segment attention in neural relation extraction. In *Proceedings of the 28th IJCAI*, pages 5401–5407. AAAI Press.
- Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of machine learning research*, 3(Feb):1083–1106.
- Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014*, pages 2335–2344.
- Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 EMNLP*, pages 1753–1762.
- Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network. *arXiv preprint* arXiv:1508.01006.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 EMNLP*, pages 35–45.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 EMNLP*, pages 2205–2215, October-November.

GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd ACL*, pages 427–434. ACL.