Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 6190–6199
Marseille, 20-25 June 2022

Enhancing Relation Extraction via Adversarial Multi-task Learning

Han Qin[♠*], **Yuanhe Tian**[♥*], **Yan Song**[♠†]
[♠]The Chinese University of Hong Kong (Shenzhen) [♥]University of Washington
[♠]hanqin@link.cuhk.edu.cn [♥]yhtian@uw.edu [♠]songyan@cuhk.edu.cn

Abstract

Relation extraction (RE) is a sub-field of information extraction, which aims to extract the relation between two given named entities (NEs) in a sentence and thus requires a good understanding of contextual information, especially the entities and their surrounding texts. However, limited attention is paid by most existing studies to re-modeling the given NEs and thus lead to inferior RE results when NEs are sometimes ambiguous. In this paper, we propose a RE model with two training stages, where adversarial multi-task learning is applied to the first training stage to explicitly recover the given NEs so as to enhance the main relation extractor, which is trained alone in the second stage. In doing so, the RE model is optimized by named entity recognition (NER) and thus obtains a detailed understanding of entity-aware context. We further propose the adversarial mechanism to enhance the process, which controls the effect of NER on the main relation extractor and allows the extractor to benefit from NER while keep focusing on RE rather than the entire multi-task learning. Experimental results on two English benchmark datasets for RE demonstrate the effectiveness of our approach, where state-of-the-art performance is observed on both datasets.[‡]

Keywords: relation extraction, multi-task learning, adversarial learning

1. Introduction

Relation extraction (RE) is an important task in natural language processing (NLP), which has been widely used in many downstream NLP applications such as summarization (Wang and Cardie, 2012), question answering systems (Xu et al., 2016a) and text mining (Distiawan et al., 2019). The object of RE is to detect the relation between two given named entities (NEs) in the input sentence, where a good understanding of the contextual information is important to achieve a satisfying performance. In doing so, most previous studies (Xu et al., 2015; Miwa and Bansal, 2016; Xu et al., 2016b; Zhang et al., 2018; Guo et al., 2019; Sun et al., 2020; Yu et al., 2020; Mandya et al., 2020; Chen et al., 2021; Tian et al., 2022) leveraged the dependency tree of the input sentence and model the contextual information along the shortest dependency path between the two entities and showed promising performance on RE. However, although NEs are usually given for RE, a good modeling of them is also highly important especially when they are ambiguous in some scenarios. For example, the example sentence in Figure 1 has three different NEs (i.e., "*president*", "*cabinet*", and "*room*"), where the NE "*room*" is paired with the other two NEs with separate relations: "*room*" is the location of "*president*" and owned by "*cabinet*". In this case, with the same sentential context, "*room*" has ambiguities when it is paired with different entities.

Although identifying NEs (even though they are given) could provide more information to support RE, previous studies paid little attention to doing so. Currently, the most common approach to model NEs is to add special tokens before and after each NE in the input sentence (Baldini Soares et al., 2019; Wu and He, 2019) so as to explicitly mark them for next step. This simple and straightforward approach has been demonstrated to be effective, yet it only marks the boundary of the NEs and cannot help the model to understand further information about those NEs (e.g., the meaning of the NEs). Thus, an appropriate approach to identifying the given NEs in the input sentence has the potential to enhance RE models with better entity-aware context understanding.

Sent.: *The **president** offers a statement in the **cabinet** **room** .*

Rel.: Located (***president***, ***room***) , Owner (***cabinet***, ***room***)

Figure 1: An illustration of the ambiguities carried by multiple NEs (i.e., "*president*", "*cabinet*", "*room*") in an example sentence, where "*room*" is paired with other NEs via separate relations (e.g., "Located" means the "*president*" is located at the "*room*"; "Owner" indicates the "*cabinet*" is the owner of the "*room*").

In this paper, we propose an approach to enhance RE through adversarial multi-task learning. Specifically, our approach has **two training stages**. In the first training stage, an NE tagger is added to the main relation extractor as another learning task and its object is to recover the given NEs so as to enhance the extractor. As a result, the extractor is optimized by the named entity recognition (NER) task, from which RE is incorporated with detailed understanding to entity-aware context. To further enhance the RE and NER learning, a discriminator is added to the relation extractor to control the impact of the NER task on the main relation extractor, which allows the extractor to benefit from the NER task in a discriminative manner. In doing so, RE learning is not dominated by the multi-task process and thus the entire model focuses more on RE. Later, in the

[*]Equal contribution.
[†]Corresponding author.
[‡]Our code is available at https://github.com/synlp/RE-AMT.

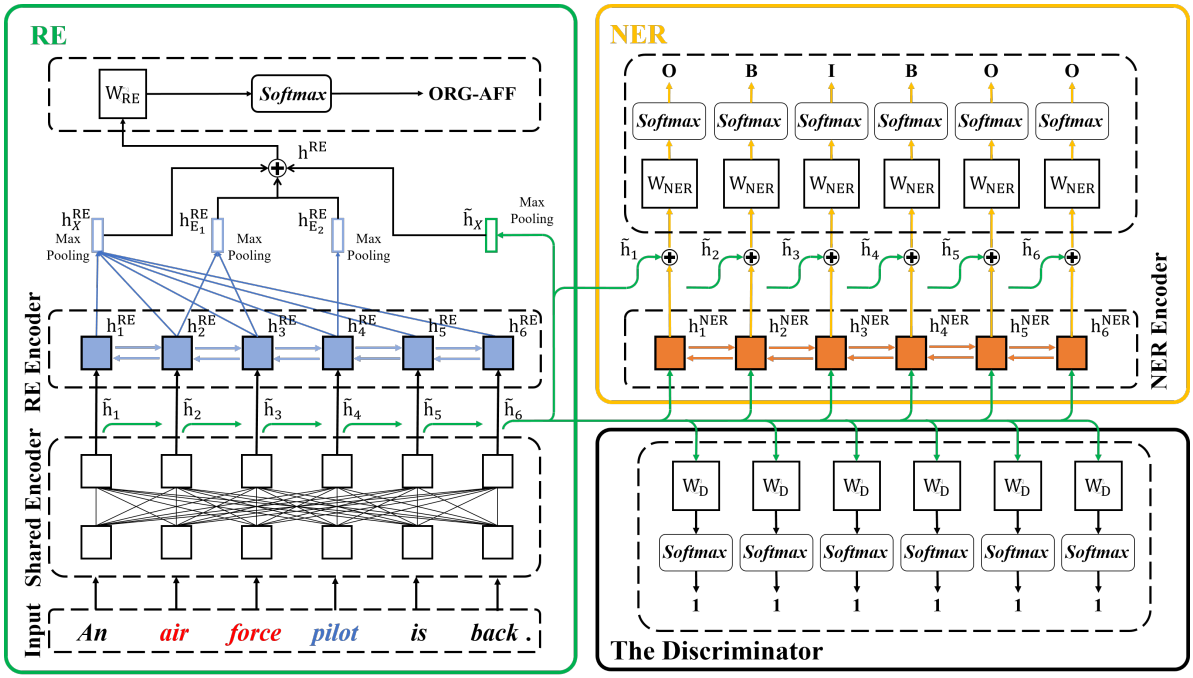

Figure 2: The overall architecture of the proposed adversarial multi-task learning for RE with an example input sentence and its two entities (i.e., "*air force*" and "*pilot*" highlighted in red and blue, respectively). The main relation extractor and NE tagger for multi-task learning of RE and NER are illustrated at left and top right, respectively, where the encoder shared by both tasks (i.e., shared encoder) is presented at bottom left. The discriminator to facilitate adversarial multi-task learning to control the effect of NER task is shown at bottom right.

second training stage, the main relation extractor is further trained alone on RE without NER and the discriminator, following the standard procedure of supervised RE training. This two-stage-training design allows our model to not only benefit from the NER task, but also be used in the same way as general RE models during reference. Particularly, compared with previous studies (Gupta et al., 2016; Luan et al., 2018) with multi-task learning for NER and RE, our approach differs from theirs in the task setting and learning objectives. In general multi-task learning for the two tasks, the NEs are not available and thus the object of their approaches is to optimize NER and RE equally at the same time, where only one training stage is required. However, in our approach, RE is the final target and NEs are given with our approach designed to focus on RE, where the discriminator (which does not exist in general multi-task learning framework) controls the effect of NER in only a part of training process without affecting the entire training and inference stages. Experimental results on two English benchmark datasets, i.e., ACE2005EN and SemEval 2010 Task 8, demonstrate the effectiveness of our approach to RE, where state-of-the-art performance is observed on both datasets.

2. The Proposed Approach

The architecture of our approach with adversarial multi-task learning is illustrated in Figure 2, where the main relation extractor for RE and the NE tagger for NER are illustrated on the left and top right side, respectively, with the shared encoder ($\mathcal{SE}$) for both tasks shown at the bottom left part. A discriminator that takes the output of the shared encoder and determines whether the NE tagger can correctly recover each given NE in the input, is illustrated on the bottom right side. Overall, our approach has two training stages. In the **first training stage**, we train the model on both RE and NER with adversarial multi-task learning, formalized by

$$\widehat{y}^{\text{RE}}, \widehat{\mathbf{y}}^{\text{NER}} = \text{Adv-MT}(\mathcal{X}, E_1, E_2) \quad (1)$$

where $\mathcal{X} = x_1 \cdots x_n$ is the input sequence with n words; E_1 and E_2 denote the two given entities in $\mathcal{X}$, which are usually sub-strings of $\mathcal{X}$ and is only visible to the main relation extractor; $\widehat{y}^{\text{RE}}$ is the predicted relation between E_1 and E_2; $\widehat{\mathbf{y}}^{\text{NER}}$ is the sequence of recovered NE tags. In the **second training stage** (presented in the green box in Figure 2), we further train the main relation extractor alone without the NE tagger and the discriminator (i.e., in the same way as the standard supervised RE training), which is formalized by

$$\widehat{y}^{\text{RE}} = f(\mathcal{X}, E_1, E_2) \quad (2)$$

In doing so, the main relation extractor is further enhanced on the target RE task itself and thus is able to achieve higher RE performance than models trained on both RE and NER.

Since the second training stage follows the standard procedure of supervised RE training, in the following texts, we focus on the proposed adversarial multi-task learning used in the first training stage. Specifically,

we start with describing the multi-task learning of RE and NER and then elaborate the details of the adversarial mechanism applied to multi-task learning in our approach.

2.1. Multi-task Learning for RE

In general, a good understanding of NEs and their surrounding texts is highly important for RE. One common approach to benefit from heterogeneous tasks is to perform multi-task learning, so that the model can learn from different resources and thus achieve promising results (Gupta et al., 2016; Chen et al., 2017; Luan et al., 2018; Xia et al., 2019; Barnes et al., 2019; Qin et al., 2021a; Qin et al., 2022). Following this paradigm, we propose to perform multi-task learning for RE and NER to learn entity-aware contextual information and thus enhance RE performance accordingly. Specifically, our approach follows the encoding-decoding setting, where the task-free encoder in the main relation extractor is shared by both RE and NER. Such shared encoder (SE) encodes task-free contextual information in the input sentence $\mathcal{X}$ by

$$[\widetilde{\mathbf{h}}_1, \cdots, \widetilde{\mathbf{h}}_n] = \text{SE}(\mathcal{X}) \tag{3}$$

where $\widetilde{\mathbf{h}}_i$ ($i \in [1, n]$) denotes the task-free hidden vector for x_i. Then, $\widetilde{\mathbf{h}}_1 \cdots \widetilde{\mathbf{h}}_n$ are fed to task-specific encoders (e.g., BiLSTM) and decoders (e.g., softmax decoder) to predict the task labels (i.e., relations and NE tags). Afterwards, the predictions are compared with the gold standards to obtain the task-specific loss. Finally, the losses from RE and NER (denoted by $\mathcal{L}_{\text{RE}}$ and $\mathcal{L}_{\text{NER}}$, respectively) are summed together to obtain the final loss $\mathcal{L}$ of the entire model by

$$\mathcal{L} = \mathcal{L}_{\text{RE}} + \lambda \cdot \mathcal{L}_{\text{NER}} \tag{4}$$

where λ is a positive hyper-parameter to control how much NER contribute to the overall learning process. The detailed procedure to obtain the losses for RE (i.e., $\mathcal{L}_{\text{RE}}$) and NER (i.e., $\mathcal{L}_{\text{NER}}$) is described as follows.

Relation Extraction RE is generally formalized as a classification task with a given input sentence $\mathcal{X}$ and two NEs, i.e., E_1 and E_2. We firstly feed $\widetilde{\mathbf{h}}_1 \cdots \widetilde{\mathbf{h}}_n$ obtained from the shared encoder to the RE encoder (which can be any type of popular encoder such as BiLSTM or Transformer (Vaswani et al., 2017)) and obtain the task-specific hidden vector, i.e., $\mathbf{h}_i^{\text{RE}}$, for each x_i. Then, we apply the max pooling mechanism onto two text spans. The first is for all $\widetilde{\mathbf{h}}_i$ and $\mathbf{h}_i^{\text{RE}}$ to obtain the task-free and task-specific global sentence representations, i.e., $\widetilde{\mathbf{h}}_{\mathcal{X}}$ and $\mathbf{h}_{\mathcal{X}}^{\text{RE}}$, respectively, through

$$\widetilde{\mathbf{h}}_{\mathcal{X}} = \text{MaxPooling}(\{\widetilde{\mathbf{h}}_1, \cdots, \widetilde{\mathbf{h}}_n\}) \tag{5}$$

and

$$\mathbf{h}_{\mathcal{X}}^{\text{RE}} = \text{MaxPooling}(\{\mathbf{h}_1^{\text{RE}}, \cdots, \mathbf{h}_n^{\text{RE}}\}) \tag{6}$$

The second is for the $\mathbf{h}_i^{\text{RE}}$ of those words that belong to a particular NE (i.e., $E_k, k = 1, 2$) to compute the vector representation of the entity $\mathbf{h}_{E_k}^{\text{RE}}$ by

$$\mathbf{h}_{E_k}^{\text{RE}} = \text{MaxPooling}(\{\mathbf{h}_i^{\text{RE}} | x_i \in E_k\}) \tag{7}$$

Then, we concatenate (i.e., $\oplus$) the task-free and task-specific representations of the sentence (i.e., $\widetilde{\mathbf{h}}_{\mathcal{X}}$ and $\mathbf{h}_{\mathcal{X}}^{\text{RE}}$), as well as the two entity representations (i.e., $\mathbf{h}_{E_1}^{\text{RE}}$ and $\mathbf{h}_{E_2}^{\text{RE}}$), to obtain

$$\mathbf{h}^{\text{RE}} = \widetilde{\mathbf{h}}_{\mathcal{X}} \oplus \mathbf{h}_{\mathcal{X}}^{\text{RE}} \oplus \mathbf{h}_{E_1}^{\text{RE}} \oplus \mathbf{h}_{E_2}^{\text{RE}} \tag{8}$$

for final prediction and feed $\mathbf{h}^{\text{RE}}$ to a softmax classifier and obtain $\mathbf{o}^{\text{RE}}$ in the output space by

$$\mathbf{o}^{\text{RE}} = \text{softmax}\left(\mathbf{W}_{\text{RE}} \cdot \mathbf{h}^{\text{RE}} + \mathbf{b}_{\text{RE}}\right) \tag{9}$$

where $\mathbf{W}_{\text{RE}}$ and $\mathbf{b}_{\text{RE}}$ represent the trainable matrix and bias vector, respectively, and each dimension of $\mathbf{o}^{\text{RE}}$ represents the predicted probability of a particular relation type given $\mathcal{X}$ and two NEs (i.e., E_1 and E_2). Finally, we apply the negative log likelihood loss function to the predictions, where the loss for RE (i.e., $\mathcal{L}_{\text{RE}}$) is computed by

$$\mathcal{L}_{\text{RE}} = -\log p(y^{\text{RE}^*} | \mathcal{X}, E_1, E_2) \tag{10}$$

where $p(y^{\text{RE}^*} | \mathcal{X}, E_1, E_2)$ denotes the predicted probability of the ground truth relation y^{RE^*} between the given entity pair (i.e., E_1 and E_2) in the sentence $\mathcal{X}$.

Named Entity Recognition NER is conventionally performed as a sequence labeling task, where each word is tagged by an NE label following the "BIO" schema. Similar to RE, an NER encoder firstly takes the output of the shared encoder (i.e., $\widetilde{\mathbf{h}}_1 \cdots \widetilde{\mathbf{h}}_n$) and outputs the task-specific hidden vector $\mathbf{h}_i^{\text{NER}}$ for x_i. Next, for each x_i, we concatenate $\widetilde{\mathbf{h}}_i$ and $\mathbf{h}_i^{\text{NER}}$ and feed the computed vector to a softmax classifier for NER, which is formalized by

$$\mathbf{o}_i^{\text{NER}} = \text{softmax}\left(\mathbf{W}_{\text{NER}} \cdot \left(\widetilde{\mathbf{h}}_i \oplus \mathbf{h}_i^{\text{NER}}\right) + \mathbf{b}_{\text{NER}}\right) \tag{11}$$

where $\mathbf{W}_{\text{NER}}$ and $\mathbf{b}_{\text{NER}}$ are the trainable matrix and bias vector, respectively, and each dimension of $\mathbf{o}_i^{\text{NER}}$ represents the predicted probability of a particular NE tag for x_i. Finally, we apply the negative log likelihood loss function to NER predictions for all words and obtain the loss (i.e., $\mathcal{L}_{\text{NER}}$) by

$$\mathcal{L}_{\text{NER}} = -\sum_{i=1}^{n} \log p(y_i^{\text{NER}^*} | \mathcal{X}) \tag{12}$$

where $p(y_i^{\text{NER}^*} | \mathcal{X})$ is the predicted probability of the ground truth NE label $y_i^{\text{NER}^*}$ for the word x_i.

2.2. The Adversarial Learning of Multi-tasks

Although the aforementioned multi-task learning approach uses a predefined hyper-parameter (i.e., λ) to

balance the effect of NER on the main relation extractor, this approach cannot automatically adjust the appropriate contribution of NER during the training process, which may lead the main relation extractor to either learn limited entity information from the NER task (in cases where the λ is small) or be dominated by NER (in cases where the λ is big). To further enhance RE and NER learning, we add a discriminator to the shared encoder (i.e., SE) to control the impact of NER on the main relation extractor. The discriminator is designed to take the output of the shared encoder (i.e., $\widetilde{\mathbf{h}_1} \cdots \widetilde{\mathbf{h}_n}$) as its input and predict whether the NE tagger can successfully recover the given NEs. Therefore, for each word x_i, the discriminator aims to make a binary classification (we define the prediction for x_i as $\widehat{y}_i^{\mathrm{D}}$ in the $\{0, 1\}$ label set), where the ground truth (denoted by $y_i^{\mathrm{D}^*}$) of x_i can be defined by

$$y_i^{\mathrm{D}^*} = \begin{cases} 0 & \widehat{\mathbf{y}}_i^{\mathrm{NER}} \neq \mathbf{y}_i^{\mathrm{NER}^*} \\ 1 & \widehat{\mathbf{y}}_i^{\mathrm{NER}} = \mathbf{y}_i^{\mathrm{NER}^*} \end{cases} \tag{13}$$

where $\widehat{\mathbf{y}}_i^{\mathrm{NER}}$ and $\mathbf{y}_i^{\mathrm{NER}^*}$ refer to the recovered and the ground truth NE label for the word x_i, respectively.
Specifically, for each word x_i in $\mathcal{X}$, the discriminator maps $\widetilde{\mathbf{h}}_i$ to a two dimensional vector $\mathbf{o}_i^D$ by a fully connect layer with softmax activation function, which is denoted by

$$\mathbf{o}_i^{\mathrm{D}} = \mathrm{softmax}\left(\mathbf{W}_{\mathrm{D}} \cdot \widetilde{\mathbf{h}}_i + \mathbf{b}_{\mathrm{D}}\right) \tag{14}$$

where $\mathbf{W}_{\mathrm{D}}$ and $\mathbf{b}_{\mathrm{D}}$ represent the trainable matrix and bias vector, respectively. Herein, the values at the first and second dimension of $\mathbf{o}_i^{\mathrm{D}}$ represent the probabilities of classifying x_i as class 0 and class 1 (defined by Eq. (13)), respectively. Afterwards, we apply the negative log likelihood loss function to the discriminator and compute the loss $\mathcal{L}_{\mathrm{D}}$ by

$$\mathcal{L}_{\mathrm{D}} = -\sum_{i=1}^{n} \log p(y_i^{\mathrm{D}^*} | \mathcal{X}) \tag{15}$$

Finally, we use the loss from the discriminator to control the effect of NER by multiplying $\mathcal{L}_{\mathrm{D}}$ and $\mathcal{L}_{\mathrm{NER}}$. Therefore, the objective of our approach with adversarial multi-task learning is to minimize the total loss $\mathcal{L}$ defined by

$$\mathcal{L} = \mathcal{L}_{\mathrm{RE}} + \mathcal{L}_{\mathrm{D}} \times (\lambda \cdot \mathcal{L}_{\mathrm{NER}}) \tag{16}$$

with $\mathcal{L}_{\mathrm{RE}}$, $\mathcal{L}_{\mathrm{NER}}$, and λ following Eq. (4).
Through this process, the effect of NER is dynamically controlled by the discriminator, which can be further explained as follows. On the one hand, when the NE tagger successfully recovers the given NEs (i.e., $y_i^{\mathrm{D}^*} = 1$) and the discriminator predicts that the NE tagger can recover the NEs (i.e., $\widehat{y}_i^{\mathrm{D}} = 1$) (which means the shared encoder in the main relation extractor has already have a good modeling to the NEs and their context), both $\mathcal{L}_{\mathrm{D}}$ and $\mathcal{L}_{\mathrm{NER}}$ are relatively small. Therefore, the loss from the NE tagger is further small and thus reduces the influence of NER to the main relation extractor. On the other hand, when the NE tagger makes an incorrect prediction (i.e., $y_i^{\mathrm{D}^*} = 0$) and the discriminator predicts that the NE tagger cannot recover the NEs (i.e., $\widehat{y}_i^{\mathrm{D}} = 0$) (which means the NE tagger cannot identify the NEs correctly), $\mathcal{L}_{\mathrm{D}}$ is relatively small while $\mathcal{L}_{\mathrm{NER}}$ is rather large. As a result, a mild loss from NER is obtained, which allows the entire model learns to identify NEs in a gentle manner. In the rest cases where $\widehat{y}_i^{\mathrm{D}} \neq y_i^{\mathrm{D}^*}$ (which means the main relation extractor cannot understand the NEs correctly), $\mathcal{L}_{\mathrm{D}}$ is relatively large, the main relation extractor is then forced to learn more NE information from NER.

RE Label Set	**NER Label Set**	
	E_1	E_2
Cause-Effect	Cause	Effect
Instrument-Agency	Instrument	Agency
Product-Producer	Product	Producer
Content-Container	Content	Container
Entity-Origin	Entity	Origin
Entity-Destination	Entity	Destination
Component-Whole	Component	Whole
Member-Collection	Member	Collection
Message-Topic	Message	Topic
other	other	other

Table 1: Rules applied to tag NEs from different relation types.

Hyper-parameters	**Values**
Learning Rate	$5e-6, \mathbf{1e-5}, 3e-5, 5e-5$
Warmup Rate	$\mathbf{0.06}, 0.1$
Dropout Rate	$\mathbf{0.1}$
Batch Size	$4, \mathbf{8}$

Table 2: The hyper-parameters are tested in tuning our models and the best one used in our final experiments are highlighted in boldface.

3. Experimental Settings

3.1. Datasets

Following previous studies (Wu and He, 2019; Tian et al., 2021; Chen et al., 2021), we use two English benchmark datasets in the experiments for RE, namely, ACE2005EN (ACE05) and SemEval 2010 Task 8 (SemEval) (Hendrickx et al., 2010). For ACE05, we use its English section and follow previous studies (Miwa and Bansal, 2016; Christopoulou et al., 2018; Ye et al., 2019) to pre-process it, where two small subsets cts and un are removed. Then, we split the dataset into training, development, and test sets. For SemEval, we use its official train/test split.

3.2. NE Label Extraction

To perform our approach, the ground truth NE labels with necessary detailed type information (e.g., person, organization, or location) would be helpful to enhance the model's understanding to the given NEs. Therefore, to facilitate the adversarial multi-task learning of RE and NER, we try three different methods to assign words with NE labels.

The first method uses simple labels (SL), which do not distinguish different NE types (i.e., each word is either tagged by "NE" or "non-NE"). The second method uses relation labels (RL), which are obtained from the relation type between the two NEs, to present NEs. Normally, a relation label is in a "type1-type2" form with two parts, where each of them illustrates the role of a NE in that relation. Therefore, in this method, different parts of the label (i.e., "type1", "type2") are assigned to the two corresponding NEs. For example, for the two entities E_1 and E_2 in relation "Content-Container (e1, e2)", the NE type of E_1 is "Content" while the NE type of E_2 is "Container". For the special relation "Other" that only contains one part, we regard "Other" as the NE type for both NEs. Table 1 elaborates all the rules to extract NE types from the relation type. The third method uses the NER results of an off-the-shelf toolkit (TL), namely, Stanford CoreNLP Toolkit Manning et al. (2014). We use the following rules to resolve the conflicts between toolkit outputs and the given NEs in an input instance: (1) we ignore the NEs recognized by the toolkit for the words other than the given NEs; (2) we use "UNK" as the NE type for the given NEs if they are not recognized by the toolkit.

Note that, the extracted NE labels are only used in the first training stage and they are either extracted from the ground truth that are available in supervised training (for SL and RL) or obtained through off-the-shelf-toolkit, without requiring any manual work. In the second training stage and in inference, we follow the standard procedure to train and test supervised RE models, where the NE labels are not used.

3.3. Implementation

For the shared encoder, because pre-trained word embeddings and language models have shown their effectiveness for many NLP tasks (Pennington et al., 2014; Song et al., 2018; Peters et al., 2018; Song and Shi, 2018; Zhang et al., 2019; Yang et al., 2019; Diao et al., 2020; Joshi et al., 2020; Song et al., 2021; Diao et al., 2021), we try one of the most representative ones, namely, BERT-large-uncased[1] (Devlin et al., 2019), following the default settings (i.e., we use 24 layers of multi-head attentions with 1024-dimensional hidden vectors). In addition, we try two different types of task-specific encoder, namely, BiLSTM and Transformer, for RE and NER. For other hyper-parameters used in training our model, we illustrate them in Table 2. We test all combinations of them for each model and use the one achieving the highest accuracy score in our final experiments.

For evaluation, we follow previous studies to use the standard micro-F1 scores for ACE05 and macro-averaged F1 scores for SemEval, where the relation type "*Other*" is ignored. In our experiments, we try different combinations of hyper-parameters, and tune them on the dev set. Then we conduct evaluation on the test set of the model that achieves the highest F1 score on the dev set.

[1]We download the BERT models from `https://github.com/huggingface/transformers`.

	ACE05	SemEval	# Para.
BiLSTM	75.84	88.93	351,964K
+ MT (SL)	76.27	89.53	368,783K
+ Adv-MT (SL)	77.39	89.69	368,785K
+ MT (RL)	78.35	89.53	368,805K
+ Adv-MT (RL)	**79.11**	**89.89**	368,807K
+ MT (TL)	77.27	89.35	368,818K
+ Adv-MT (TL)	77.48	89.58	368,820K
Transformer	77.04	89.01	343,556K
+ MT (SL)	77.13	89.47	351,972K
+ Adv-MT (SL)	77.70	89.64	351,974K
+ MT (RL)	77.19	89.53	351,983K
+ Adv-MT (RL)	**79.25**	**90.02**	351,985K
+ MT (TL)	77.25	89.35	351,989K
+ Adv-MT (TL)	77.69	89.51	351,991K

Table 3: F1 scores of different models with BiLSTM and Transformer task-specific encoder on the test sets of SemEval and ACE05, where the number of parameters (i.e., Para.) is also reported. "MT" is the baseline with multi-task learning; "Adv-MT" refers to our approach with adversarial multi-task learning. "SL", "RL", and "TL" are three different types of NE labels used for NER, respectively.

4. Results and Analyses

4.1. Overall Results

In the experiments, we run two baselines and our model with BERT-large as the shared encoder and different types of task-specific encoders (i.e., BiLSTM or Transformer). The first baseline (i.e., "BiLSTM" or "Transformer" in Table 3) follows the standard process to train a RE model without using multi-task learning; the second baseline (i.e., "MT " in Table 3) performs multi-task learning on RE and NER without the adversarial learning. For all models with multi-task learning, we try three different methods (i.e., SL, RL, and TL) to extract NE labels for the NE tagger. The F1 scores[2] of different models (as well as their sizes in terms of

[2]We report the performance of these models on the development set and the mean and the standard deviation of the test set results in Appendix E and Appendix F, respectively.

Models	ACE05	SemEval
†Zhang et al. (2018)	-	84.8
Wu and He (2019)	-	89.2
Christopoulou et al. (2018)	64.2	-
Ye et al. (2019)	68.9	-
Baldini Soares et al. (2019)	-	89.5
†Mandya et al. (2020)	-	85.9
†Sun et al. (2020)	-	86.0
†Yu et al. (2020)	-	86.4
Wang et al. (2020)	66.7	-
Wang and Lu (2020)	67.6	-
Wang et al. (2021)	66.0	-
†Tian et al. (2021)	79.05	89.85
BiLSTM + Adv-MT (RL)	79.11	89.89
Transformer + Adv-MT (RL)	**79.25**	**90.02**

Table 4: The comparison of F1 scores between previous studies and our best model with BERT-large on the test sets of ACE05 and SemEval. Previous studies that leverage syntactic information (e.g., the dependency tree of the input sentence) are marked by "†".

	ACE05		SemEval	
	RE	NER	RE	NER
TF + MT (SL)	76.81	52.31	88.72	52.49
TF + Adv-MT (SL)	77.02	50.28	89.13	49.37
TF + MT (RL)	77.10	50.85	88.96	52.34
TF + Adv-MT (RL)	**77.70**	49.92	**89.57**	47.21
TF + MT (TL)	76.58	50.39	88.67	51.97
TF + Adv-MT (TL)	77.03	48.32	89.11	50.02

Table 5: The RE and NER results (F1 scores) of multi-task baselines and our approach after the first training stage. "TF" denotes Transformer task-specific encoder.

the number of trainable parameters) on the test sets of ACE05 and SemEval are reported in Table 3.

There are several observations. First, although the baselines without multi-task learning (MT) have achieved outstanding performance, all models with MT further improve model performance. This observation confirms the effectiveness of learning from NER task to improve RE. Second, compared with the baseline with the standard multi-task learning (i.e., MT), our approach with adversarial learning (i.e., Adv-MT) further improves the model performance with only 2K more parameters. It indicates the effectiveness of the discriminator to control the impact of the NER task on the main relation extractor, which allows the extractor to benefit from the NER task in a discriminative manner. Third, among models with different NE types (i.e., SL, RL, and TL), the ones configured with RL achieve the highest F1 scores in most cases. A possible explanation could be that, compared with SL that ignores NE type, labels extracted from the relation labels (i.e., the RL NE type) provides more detailed information of the given

	ACE05
Transformer	73.27
+ MT (SL)	73.35
+ Adv-MT (SL)	73.94
+ MT (RL)	73.39
+ Adv-MT (RL)	**74.51**
+ MT (TL)	73.33
+ Adv-MT (TL)	73.96

Table 6: F1 scores of baselines and our approach with Transformer task-specific encoder and different types of NE label (i.e., SL, RL, and TL) on a subset of the ACE05 test set, where each test sentence contains at least two entity pairs.

NEs, which leads our approach to have a better NE understanding. In addition, compared with TL, RL tends to provide more information of the particular roles that the NEs are playing in a relation, which is more closer to the object of the RE. As a result, our model benefits more from the RL and obtains the highest performance among different NE type settings.

4.2. Comparison with Previous Studies

To further demonstrate the effectiveness of our approach, we compare our approach with BERT-large shared encoder under the best setting (i.e., Adv-MT (RL) with Transformer task-specific encoder) with previous studies and report the results in Table 4, where state-of-the-art performance is observed. Specifically, our approach outperforms Wu and He (2019) and Baldini Soares et al. (2019) that use BERT-large encoder, which demonstrates the effectiveness of our approach to learn NEs and their contexts from adversarial multi-task learning. In addition, compared with previous studies (Zhang et al., 2018; Mandya et al., 2020; Guo et al., 2019; Sun et al., 2020; Yu et al., 2020) that leverage dependency information (marked by † in Table 4), our approach provides an alternative to improve RE by explicitly modeling the NEs and its contexts through adversarial multi-task learning, without relying on the existence of a dependency parser.

4.3. Effect of the Discriminator

Compared with the standard multi-task learning baseline, our approach applies a discriminator to the main relation extractor to control the effect of NER task so as to allow the extractor discriminatively learn from but not being dominated by NER. To explore whether the discriminator functionalizes as excepted, we extract the intermediate models (checkpoints) obtained after the first training stage from our best performing models with and without the discriminator (Transformer+MT) and evaluate such intermediate models on RE and NER separately. The results are reported in Table 5. It is clearly showed that our approach (i.e., Transformer+Adv-MT) outperforms the baseline

Sentence:	*Well, security is tight as **exxon mobil** **shareholders** meet today in **dallas**.*
Transformer (SL):	Other (***shareholders***, ***exxon mobil***), PART-WHOLE (***shareholders***, ***dallas***)
Transformer+MT (SL):	PHYS (***shareholders***, ***exxon mobil***), PHYS (***shareholders***, ***dallas***)
Transformer+Adv-MT (SL):	ORG-AFF (***shareholders***, ***exxon mobil***), PHYS (***shareholders***, ***dallas***)

Figure 3: A case study of different models on an example sentence with three NEs (i.e., "*exxon mobil*", "*shareholders*", and "*dallas*" highlighted in blue, red, and green colors, respectively), where the relations of each NE pair are different. The model predictions on different NE pairs are also presented, where the prediction from our model (i.e., Transformer + Adv-MT (SL)) matches the gold standard while the other two baseline models fail to do so.

without the discriminator (i.e., Transformer+MT) on RE and both models show comparable performance on NER. Given that our approach outperforms the baseline after the second training stage (see Table 3), the observation from intermediate models confirms that the discriminator successfully controls the influence of NER on the main relation extractor and avoid performance hurt from the plain multi-task learning.

4.4. Effect of NER for RE

To explore whether our approach successfully leverages the information from NE from the first training stage, particularly by handling the ambiguities introduced by different NEs, we extract sentences with at least two entity pairs from the test set of ACE05 dataset and evaluate both baselines and our approach with the best setting on these sentences. Table 6 reports the F1 scores of different models on the subset, where our model outperforms the two baselines and demonstrates its validity and effectiveness to learn and address the ambiguities of NEs to improve RE. Specifically, among the three different types of NE labels, our models with RL achieves the best performance, which is reasonable based on the following explanations. Compared with SL, RL carries more detailed information of the given NE (i.e. the corresponding relation types), leading to better understandings of NEs along with their surrounding texts. In addition, compared with TL which migrates the label types from other benchmark datasets whose domain might be different from the domain of ACE05, RL provides more localized knowledge instead of cross domain knowledge for RE.

4.5. Case Study

Further more, we conduct a case study on an example sentence to further analyze the effect of NER for RE. Figure 3 illustrates the example sentences with two entity pairs, namely ("*exxon mobil*", "*shareholders*") and ("*shareholders*", "*dallas*"), as well as the predictions from the baseline (i.e., Transformer), the multi-task learning baseline (i.e., Transformer+MT (SL)), and our approach (i.e., Transformer+Adv-MT (SL)), where our approach correctly extracts the relation between both entity pairs whereas the two baselines do not. In this case, the NE "*shareholders*" is ambiguous because it plays different roles when it is paired with "*exxon mobil*" and "*dallas*". Therefore, the baseline model without multi-task learning misinterprets the entity pair "*shareholders*", "*dallas*" and hence predicts incorrect relation "PART-WHOLE" between them. In addition, we note that, for both Transformer+MT (SL) and Transformer+Adv-MT (SL) with multi-task learning, their corresponding intermediate models obtained from the first training stage can successfully recognize the NEs. In this case, although the Transformer+MT (SL) baseline identifies the NEs, it still makes incorrect prediction, where one possible explanation is that the main relation tagger in the baseline is potentially over-influenced by the NER task and thus leads to inferior performance for RE. As a comparison, our approach with the discriminator is able to prevent the main relation extractor from being dominated by NER and hence extracts the correct relations between different entity pairs.

5. Related Work

Recently, neural methods (Zeng et al., 2014; Zhang et al., 2015; Zhou et al., 2016; Wang et al., 2016; Shen and Huang, 2016; Zhang et al., 2017; Christopoulou et al., 2018; Wang and Lu, 2020; Wang et al., 2020; Sainz et al., 2021; Chen et al., 2021; Lyu and Chen, 2021; Qin et al., 2021b) with advanced encoders (e.g., Transformer) have achieved satisfying performance in RE because of their power in capturing contextual information. To further improve model performance, most studies (Xu et al., 2016b; Zhang et al., 2018; Guo et al., 2019; Sun et al., 2020; Yu et al., 2020; Mandya et al., 2020) leverage extra resources, such as dependency trees of the input sentence, to capture more contextual information along the shorted dependency path between two given NEs, where necessary pruning strategies are required (Xu et al., 2015; Xu et al., 2016a; Miwa and Bansal, 2016; Zhang et al., 2018; Yu et al., 2020) to filter noise in the dependency trees. However, although NEs are given in the task setting, a good modeling to them and their contexts is also highly important, especially in cases where NEs are ambiguous. Currently, the most common approach to identifying NEs is to add special tokens before and after each entity in the input sentence (Baldini Soares et al., 2019; Wu and He, 2019). However, it only marks boundaries of NEs and cannot assist the model to understand further contextual information around NEs.

Compared with the aforementioned methods, our approach learns to identify NEs from NER through adversarial multi-task learning, from which RE is enhanced

with detailed understanding to entity-aware context. In terms of multi-tasking learning, previous studies (Gupta et al., 2016; Luan et al., 2018) for both RE and NER are set in the situation that NEs are not available and their object is to optimize both tasks equally, our approach treats RE as the final target, where the discriminator controls the effect of NER in only a part of training process without affecting the entire training and inference stages. Moreover, considering that in most RE tasks that NEs are given, our proposed approach has the advantage of improving RE performance without requiring complicated resources (e.g., syntax) which may not be available in real applications.

6. Conclusion

In this paper, we propose to enhance relation extraction (RE) through adversarial multi-task learning. In detail, our approach has two training stages, where an NE tagger is added to the main relation extractor as an extra learning task. The object of the NE tagger is to recover the given NEs in order to improve the main extractor with detailed understanding to entity-aware context through named entity recognition (NER). Based on multi-task learning of RE and NER, adversarial learning is applied to it to further control the effect of NER on the main relation extractor and thus allows RE to benefit from NER other than equally treating the two tasks. Experimental results on two English benchmark datasets illustrate the validity and effectiveness of our model, with state-of-the-art performance.

7. Acknowledgements

This work is supported by Shenzhen Science and Technology Program under the project "Fundamental Algorithms of Natural Language Understanding for Chinese Medical Text Processing" (JCYJ20210324130208022) and the Natural Science Foundation of Guangdong Province, China, under the project "Deep Learning based Chinese Combination Category Grammar Parsing and its Application in Relation Extraction".

8. Bibliographical References

Baldini Soares, L., FitzGerald, N., Ling, J., and Kwiatkowski, T. (2019). Matching the Blanks: Distributional Similarity for Relation Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905.

Barnes, J., Touileb, S., Øvrelid, L., and Velldal, E. (2019). Lexicon Information in Neural Sentiment Analysis: A Multi-task Learning Approach. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 175–186.

Chen, X., Shi, Z., Qiu, X., and Huang, X. (2017). Adversarial Multi-Criteria Learning for Chinese Word Segmentation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1193–1203, Vancouver, Canada, July.

Chen, G., Tian, Y., Song, Y., and Wan, X. (2021). Relation Extraction with Type-aware Map Memories of Word Dependencies. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2501–2512, Online, August.

Christopoulou, F., Miwa, M., and Ananiadou, S. (2018). A Walk-based Model on Entity Graphs for Relation Extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 81–88.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Diao, S., Bai, J., Song, Y., Zhang, T., and Wang, Y. (2020). ZEN: Pre-training Chinese Text Encoder Enhanced by N-gram Representations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4729–4740, November.

Diao, S., Xu, R., Su, H., Jiang, Y., Song, Y., and Zhang, T. (2021). Taming Pre-trained Language Models with N-gram Representations for Low-Resource Domain Adaptation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3336–3349, Online, August.

Distiawan, B., Weikum, G., Qi, J., and Zhang, R. (2019). Neural Relation Extraction for Knowledge Base Enrichment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 229–240.

Guo, Z., Zhang, Y., and Lu, W. (2019). Attention Guided Graph Convolutional Networks for Relation Extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 241–251, July.

Gupta, P., Schütze, H., and Andrassy, B. (2016). Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2537–2547.

Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., Pennacchiotti, M., Romano, L., and Szpakowicz, S. (2010). SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, July.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., and Levy, O. (2020). Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Luan, Y., He, L., Ostendorf, M., and Hajishirzi, H. (2018). Multi-task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232.

Lyu, S. and Chen, H. (2021). Relation Classification with Entity Type Restriction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 390–395, Online, August.

Mandya, A., Bollegala, D., and Coenen, F. (2020). Graph Convolution over Multiple Dependency Subgraphs for Relation Extraction. In *COLING*, pages 6424–6435. International Committee on Computational Linguistics.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., and McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Miwa, M. and Bansal, M. (2016). End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June.

Qin, H., Chen, G., Tian, Y., and Song, Y. (2021a). Improving Arabic Diacritization with Regularized Decoding and Adversarial Training. In *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*.

Qin, H., Tian, Y., and Song, Y. (2021b). Relation Extraction with Word Graphs from N-grams. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2860–2868, Online and Punta Cana, Dominican Republic, November.

Qin, H., Tian, Y., Xia, F., and Song, Y. (2022). Complementary Learning of Aspect Terms for Aspect-based Sentiment Analysis. In *Proceedings of the 13th Language Resources and Evaluation Conference*.

Sainz, O., Lopez de Lacalle, O., Labaka, G., Barrena, A., and Agirre, E. (2021). Label Verbalization and Entailment for Effective Zero and Few-Shot Relation Extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1199–1212, Online and Punta Cana, Dominican Republic, November.

Shen, Y. and Huang, X.-J. (2016). Attention-based convolutional neural network for semantic relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2526–2536.

Song, Y. and Shi, S. (2018). Complementary Learning of Word Embeddings. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4368–4374, 7.

Song, Y., Shi, S., and Li, J. (2018). Joint Learning Embeddings for Chinese Words and Their Components via Ladder Structured Networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4375–4381.

Song, Y., Zhang, T., Wang, Y., and Lee, K.-F. (2021). ZEN 2.0: Continue Training and Adaption for N-gram Enhanced Text Encoders. *arXiv preprint arXiv:2105.01279*.

Sun, K., Zhang, R., Mao, Y., Mensah, S., and Liu, X. (2020). Relation Extraction with Convolutional Network over Learnable Syntax-Transport Graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8928–8935.

Tian, Y., Chen, G., Song, Y., and Wan, X. (2021). Dependency-driven Relation Extraction with Attentive Graph Convolutional Networks. In *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*.

Tian, Y., Song, Y., and Xia, F. (2022). Improving Relation Extraction through Syntax-induced Pre-training with Dependency Masking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, May.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is All you Need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Wang, L. and Cardie, C. (2012). Focused Meeting Summarization via Unsupervised Relation Extraction. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 304–313.

Wang, J. and Lu, W. (2020). Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1706–1721, Online, November.

Wang, L., Cao, Z., De Melo, G., and Liu, Z. (2016).

Relation classification via multi-level attention cnns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1298–1307.

Wang, Y., Sun, C., Wu, Y., Yan, J., Gao, P., and Xie, G. (2020). Pre-training entity relation encoder with intra-span and inter-span information. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1692–1705, Online, November.

Wang, Y., Sun, C., Wu, Y., Zhou, H., Li, L., and Yan, J. (2021). UNIRE: A Unified Label Space for Entity Relation Extraction. *arXiv preprint arXiv:2107.04292*.

Wu, S. and He, Y. (2019). Enriching pre-trained language model with entity information for relation classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2361–2364.

Xia, Q., Li, Z., and Zhang, M. (2019). A Syntax-aware Multi-task Learning Framework for Chinese Semantic Role Labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5382–5392.

Xu, Y., Mou, L., Li, G., Chen, Y., Peng, H., and Jin, Z. (2015). Classifying Relations via Long Short Term Memory Networks Along Shortest Dependency Paths. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1785–1794.

Xu, K., Reddy, S., Feng, Y., Huang, S., and Zhao, D. (2016a). Question Answering on Freebase via Relation Extraction and Textual Evidence. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2326–2336.

Xu, Y., Jia, R., Mou, L., Li, G., Chen, Y., Lu, Y., and Jin, Z. (2016b). 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.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems 32*, pages 5753–5763.

Ye, W., Li, B., Xie, R., Sheng, Z., Chen, L., and Zhang, S. (2019). Exploiting entity BIO tag Embeddings and Multi-task Learning for Relation Extraction with Imbalanced Data. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1351–1360.

Yu, B., Xue, M., Zhang, Z., Liu, T., Wang, Y., and Wang, B. (2020). Learning to Prune Dependency Trees with Rethinking for Neural Relation Extraction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3842–3852.

Zeng, D., Liu, K., Lai, S., Zhou, G., and Zhao, J. (2014). Relation Classification via Convolutional Deep Neural Network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, August.

Zhang, S., Zheng, D., Hu, X., and Yang, M. (2015). Bidirectional Long Short-Term Memory Networks for Relation Classification. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, pages 73–78.

Zhang, Y., Zhong, V., Chen, D., Angeli, G., and Manning, C. D. (2017). Position-aware Attention and Supervised Data Improve Slot Filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Zhang, Y., Qi, P., and Manning, C. D. (2018). Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215.

Zhang, H., Bai, J., Song, Y., Xu, K., Yu, C., Song, Y., Ng, W., and Yu, D. (2019). Multiplex Word Embeddings for Selectional Preference Acquisition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5247–5256, Hong Kong, China, November.

Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., and Xu, B. (2016). Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*, pages 207–212.