Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Joint extraction of entities and overlapping relations using source-target entity labeling

Tingting Hang[a,b], Jun Feng[a,*], Yirui Wu[a], Le Yan[a], Yunfeng Wang[a]

[a] College of Computer and Information, Hohai University, Nanjing 211100, China
[b] College of Electricity and Information, Wanjiang University of Technology, Maanshan 243031, China

ARTICLE INFO

Keywords:
BERT
Entity recognition
Overlapping relation extraction
Joint extraction model

2021 MSC:
00-01
99-00

ABSTRACT

Joint extraction of entities and overlapping relations has attracted considerable attention in recent research. Existing relation extraction methods rely on a training set that is labeled by the distant supervision method for supervised relation extraction. However, the drawbacks of these methods are that large-scale unlabeled data cannot be used and the quality of labeled data cannot be guaranteed. Moreover, owing to the relatively complex overlapping relations, it is difficult to perform joint entity-relation extraction accurately. In this study, we propose an end-to-end neural network model (BERT-JEORE) for the joint extraction of entities and overlapping relations. First, we use the BERT-based parameter-sharing layer to capture the joint features of entities and overlapping relations. Then, we implement the source-target BERT model to assign entity labels to each token in a sentence, thereby expanding the amount of labeled data and improving their quality. Finally, we design a three-step overlapping relations extraction model and use it to predict the relations between all entity pairs. Experiments conducted on two public datasets show that BERT-JEORE achieves the best current performance and outperforms the baseline models by a significant margin. Further analysis shows that our model can effectively capture different types of overlapping relational triplets in a sentence.

1. Introduction

Entity and relation extraction is crucial for a range of downstream tasks in natural language processing (NLP), such as constructing knowledge graph and answering questions. Most neural network models (Miwa & Bansal, 2016; Zheng et al., 2017; Ren et al., 2017) for entity and relation extraction assume that a sentence contains only one relational fact. Nevertheless, relational facts in sentences are often complicated, and different relational triplets may overlap in a sentence (Zeng, Zeng, He, Liu, & Zhao, 2018). Therefore, the joint extraction of entities and overlapping relations has attracted considerable attention. This task mainly aims to detect all possible relational facts from a sentence while considering the complex overlap between triplets.

Existing methods for extracting overlapping relations can be divided into two categories: sequence-to-sequence (Seq2Seq) methods (Hoffmann, Zhang, Ling, Zettlemoyer, & Weld, 2011; Zeng et al., 2018; Zeng et al., 2019; Tan, Zhao, Wang, & Xiao, 2019; Zeng, Zhang, & Liu, 2020) and graph-based methods (Fu, Li, & Ma, 2019; Fei, Ren, & Ji, 2020a). Seq2Seq methods take unstructured text as an input and directly decode relational triplets as a sequential output. For example, Zeng et al. (2018) proposed a Seq2Seq model with a copy mechanism to extract overlapping relational triplets. Graph-based methods construct a graph neural network for the joint extraction of entities and overlapping relations. For example, Fu et al. (2019) employed a graph convolutional network (GCN) for modeling a word graph.

Despite their success, traditional methods for extracting overlapping triplets have several shortcomings. First, they rely on distant supervision methods (Mintz, Bills, Snow, & Jurafsky, 2009) to label training sets for supervised relation extraction. Although such methods reduce the dependence on manually labeled data to a certain extent, they have the following limitations. (1) The data labeling process completely relies on the knowledge base, and large-scale unlabeled data cannot be used. (2) If two entities have a certain relation in the knowledge base, all unstructured sentences containing the two entities can express this relation; this assumption is not always true. Furthermore, a large amount of noise data is inevitably introduced, and the quality of labeled data cannot be guaranteed. Thus, an effective extraction method should meet quantity and quality requirements of labeled data.

[*] Corresponding author.
E-mail addresses: httsf@hhu.edu.cn (T. Hang), fengjun@hhu.edu.cn (J. Feng), wuyirui@hhu.edu.cn (Y. Wu), yanle@hhu.edu.cn (L. Yan), naive@hhu.edu.cn (Y. Wang).

https://doi.org/10.1016/j.eswa.2021.114853
Received 10 August 2020; Received in revised form 9 January 2021; Accepted 3 March 2021
Available online 13 March 2021
0957-4174/© 2021 Elsevier Ltd. All rights reserved.

Sentence:

But that spasm of irritation by a master intimidator was minor compared with what **Bobby Fischer** [Entity], the erratic former world chess champion, dished out in March at a news conference in **Reykjavik** [Entity], **Iceland** [Entity].

Triples:

EPO	{Iceland, **/location/country/capital**, Reykjavik}
	{Iceland, **/location/location/contains**, Reykjavik}

SEO	{Bobby Fischer, **/people/person/nationality**, Iceland}
	{Bobby Fischer, **/people/deceased_person/place_of_death**, Reykjavik}

Fig. 1. Example sentence containing overlapping relations. The first one includes triplets with the overlapped entity pair (*Iceland*, *Reykjavik*) belonging to the EPO class. The second one includes triplets with the shared entity "Bobby Fischer" belonging to the SEO class.

Second, traditional extraction methods fail to extract relations accurately in the case of complex overlaps. As shown in Fig. 1, the entity pair "Iceland" and "Reykjavik" has two relations, namely "*/location/country/capital*" and "*/location/location/contains*"; these belong to EntityPairOverlap (EPO). Furthermore, "Bobby Fischer" appears in {*Bobby Fischer*,/people/person/nationality, Iceland} and {*Bobby Fischer*,/people/deceased_person/place_of_death, Reykjavik}; these belong to SingleEntityOverlap (SEO). Such overlapping relations are very common in sentences. Statistically, 34.39% (19,327/56,195) and 65.81% (3,303/5,019) of the sentences in the NYT (Riedel, Yao, & McCallum, 2010) and WebNLG (Gardent, Shimorina, Narayan, & Perez-Beltrachini, 2017) training sets contain multiple relations, respectively. The above-mentioned problems should be addressed, to avoid performance degradation of the entity and relation extraction task. Thus, an effective extraction method must capture complex overlapping relations between entities.

To meet the two above-mentioned requirements, we propose an end-to-end neural network model for the joint extraction of entities and overlapping relations, namely **BERT-JEORE**, which is used to extract relational triplets from normal, EPO, and SEO sentences. Our model aims to improve the extraction performance of overlapping relations. It comprises three main parts: the parameter-sharing layer, named entity recognition (NER) downstream task layer, and relation classification (RC) downstream task layer. The novelty of BERT-JEORE is that a fine-tuned entity tagging model is introduced to generate accurate entity labels in the NER phase, and a new overlapping relation extraction model (OREM) is employed to generate an unlimited number of relational triplets in the RC phase. Experiments on two public datasets show that BERT-JEORE can significantly outperform state-of-the-art methods, thereby confirming its effectiveness.

The main contributions of this work are summarized as follows:

- We propose a fine-tuned entity tagging model based on the interdependence of source BERT and target BERT.
- We propose a three-step OREM based on multiple binary relation classifiers and multi-head attention (MHA).
- Experiments confirm that BERT-JEORE achieves state-of-the-art performance with significant improvements over various baseline models. Further analyses indicate that BERT-JEORE exhibits consistent improvement in all overlap scenarios.

The remainder of the paper is organized as follows. Section 2 reviews related studies. Section 3 presents the BERT-JEORE model. Section 4 describes the experimental settings. Section 5 presents and analyzes the experimental results. Finally, Section 6 concludes the paper and discussed directions for future work.

2. Related work

In this section, we briefly review the relevant studies that inspired us to design BERT-JEORE, including those on entity and relation extraction, pre-trained language models, and the prediction of overlapping relations.

2.1. Entity and relation extraction

In recent years, entity and relation extraction has attract considerable attention. Existing methods are mainly divided into two categories: pipeline methods and joint extraction methods. Pipeline methods extract relational triplets in two steps. First, NER (Nadeau & Sekine, 2007; Fei, Ren, & Ji, 2020b) identifies all the entities in a sentence. Second, RC (Rink & Harabagiu, 2010; Wang, Qin, Lu, Luo, & Liu, 2020) is performed on the entity pair. Pipeline methods ignore the relevance between the two subtasks and suffer from error propagation (Li & Ji, 2014; Gupta, Schütze, & Andrassy, 2016). By contrast, joint extraction methods aim to integrate the information of entities and relations, reducing the error propagation and achieving better results. Traditional joint extraction methods are feature-based methods (Singh, Riedel, Martin, Zheng, & McCallum, 2013; Miwa & Sasaki, 2014; Li & Ji, 2014; Ren et al., 2017) that require a complicated process of feature engineering and heavily depend on NLP tools. Joint extraction methods based on neural networks have been developed to reduce the manual effort involved in feature extraction (Miwa & Bansal, 2016; Zheng et al., 2016; Zheng et al., 2017; Katiyar & Cardie, 2017; Bekoulis, Deleu, Demeester, & Develder, 2018; Lei, Huang, Feng, Gao, & Su, 2019; Dai et al., 2019).

These methods can be divided into two categories: those with shared parameters and those with joint decoding. Shared parameter methods (Bekoulis et al., 2018; Wei, Su, Wang, Tian, & Chang, 2020; Eberts & Ulges, 2019) realize the dependency between the NER and RC tasks by sharing parameters. However, in the extraction process of entities and relations, these methods still separate the two subtasks, which generates redundant information of entity pairs that have no relation. Joint decoding methods (Zheng et al., 2017; Li et al., 2019) have been proposed to address the issue of redundant information. These methods are global optimization problems that can jointly decode entities and relations during inference. For example, Zheng et al. (2017) proposed a novel tagging scheme that converts the task of relational triplet extraction into an end-to-end sequence tagging problem. Despite their success, joint extraction methods rely on a large amount of labeled data for supervised training and they cannot obtain pre-trained knowledge through many unsupervised corpora, resulting in low generalization performance. Furthermore, they ignore the relational triplet overlapping problem.

2.2. Pre-trained language models

The pre-trained language model (Devlin, Chang, Lee, & Toutanova, 2019) was first proposed by Google in 2018. It can better utilize a large-scale unlabeled corpus for unsupervised pertaining, and then apply it to various NLP downstream tasks through fine-tuning, with considerable success. Major companies and universities have released pre-trained models (Peters et al., 2018; Dai et al., 2019; Radford et al., 2019; Yang et al., 2019; Liu et al., 2019) that have been successfully applied to various NLP tasks (Erhan et al., 2010; Tsai et al., 2019; Zhao & He, 2019; Cui et al., 2019; Yang, Feng, Qiao, Kan, & Li, 2019).

In recent years, pre-trained language models have also been employed for entity and relation extraction, thereby reducing the dependence on supervised learning and achieving superior results (Alt, Hübner, & Hennig, 2019; Xue et al., 2019; Wei et al., 2020). For example, Alt et al. (2019) proposed a transformer-based relation extraction method that learns implicit language features from a plain-text corpus through unsupervised pre-training and then fine-tunes the

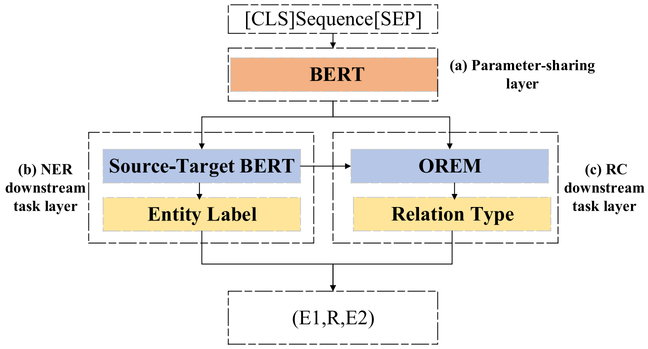

[CLS]Sequence[SEP]
BERT
(a) Parameter-sharing layer
(b) NER downstream task layer
Source-Target BERT
Entity Label
OREM
Relation Type
(c) RC downstream task layer
(E1,R,E2)

Fig. 2. Overall structure of BERT-JEORE. BERT, source-target BERT and OREM are used as the parameter-sharing layer, NRE downstream task layer, and RC downstream task layer, respectively.

language representation in the relation extraction task. Further, Xue et al. (2019) proposed a focused attention model for the joint extraction of entities and relations, which integrates the BERT language model into joint learning through a dynamic range attention mechanism, thereby improving the feature representation ability of the shared parameter layer. In addition, Wei et al. (2020) proposed a novel cascade binary tagging framework that models relations as functions that map subjects to objects, thereby solving the problem of overlapping relations intuitively.

2.3. Prediction of overlapping relations

Many recent studies aim to further explore the extraction of overlapping relations. To this end, several Seq2Seq methods (Hoffmann et al., 2011; Zeng et al., 2018; Zeng et al., 2019; Tan et al., 2019; Zeng et al., 2020) have been proposed. Zeng et al. (2018) was the first researchers to consider the overlapping triplet problem in relational triplet extraction, they proposed a Seq2Seq model with a copy mechanism to extract triplets. To improve the extraction performance of overlapping relations, they further studied the influence of the extraction order (Zeng et al., 2019) and achieved considerable improvement through reinforcement learning (RL). However, these models only copy the last word of an entity; therefore, they cannot extract the entire entity consisting of multiple words. To solve this problem, Zeng et al. (2020) proposed multi-task learning based on the Seq2Seq model to extract multi-token entities. However, the above-mentioned Seq2Seq methods only partially handle the interaction between triplets.

To predict relational triplets while considering the interactions between them, Fu et al. (2019) and Fei et al. (2020a) proposed graph-based methods to improve the extraction performance of overlapping relations. Further, Fu et al. (2019) studied the overlapping relation problem by modeling the text as relational graphs using a GCN-based model. Subsequently, Fei et al. (2020a) proposed an end-to-end neural network model for overlapping relation extraction by treating the task as a quintuple prediction problem. However, despite their success, these methods face difficulties in data migration when applied to multiple-triplet extraction in other fields. Moreover, they fail to achieve satisfactory performance when the overlapping relations are relatively complex.

To address the above-mentioned issues, we propose BERT-JEORE, an end-to-end neural network model for the joint extraction of entities and overlapping relations. Our work is inspired by Cui et al. (2019) and Yang et al. (2019).

3. Proposed model

We first define the problem of the joint extraction of entities and overlapping relations. Given a sentence with n words, $s = (w_1, w_2, \ldots, w_n)$, and a set of predefined relations $R = (r_1, r_2, \ldots, r_n)$, it aims to extract one or more triplets from normal, EPO, and SEO sentences.

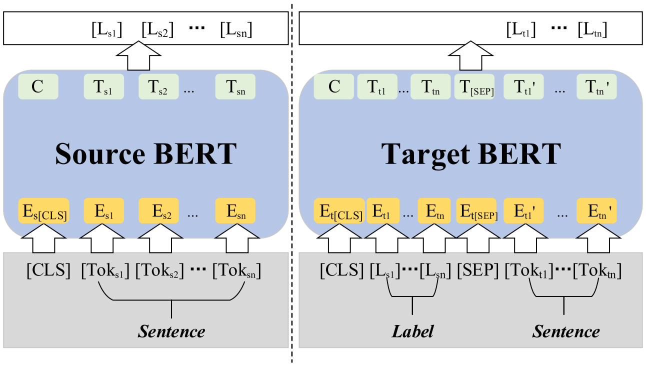


Fig. 3. Architecture of source-target BERT. [CLS] is a special symbol added in front of every input example. Here, the source sentence and target sentence refer to the same sentence.

$$Y = \{(e_1, r, e_2) | e \in E, r \in R\}, \tag{1}$$

where e_1 and e_2 denote the head and tail entities, respectively, r is the relation type that connects e_1 to e_2, and $E = \{(e_i, \ldots, e_j) | 1 \leqslant i \leqslant j \leqslant n\}$ is a set of candidate entities. Note that $(e_1, r, e_2) \neq (e_2, r, e_1)$ in terms of the relation between two entities.

Now, we will introduce the general framework of BERT-JEORE. It comprises three main parts: (a) parameter sharing layer, (b) NER downstream task layer, and (c) RC downstream task layer. Fig. 2 shows the workflow of BERT-JEORE, which comprises three steps:

Step 1: The parameter-sharing layer extracts the context information of the tokens in the sentence sequence and passes it to the corresponding downstream task layer.

Step 2: The NER downstream task layer converts the representation vector of each token in the output of the parameter-sharing layer into the probability distribution of the corresponding entity label.

Step 3: The RC downstream task layer converts the sentence representation vector in the output of the parameter-sharing layer into the probability distribution of the corresponding relation type.

We will describe the details of each part in the following subsections.

3.1. Parameter-sharing layer

We employ a pre-trained BERT model to encode the context information. It is composed of a stack of N identical transformer blocks. We denote a transformer block as $Trans(x)$, where x represents the input vector. The detailed operations are as follows:

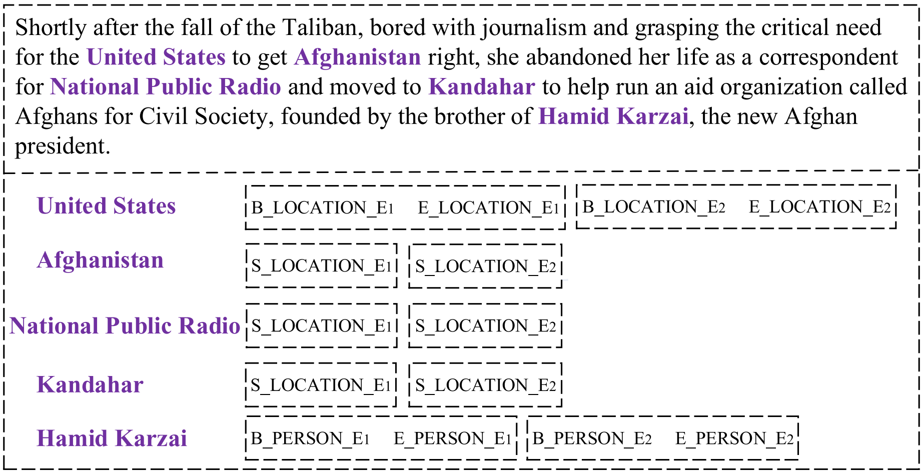
$$h_i = Trans(h_{i-1}), i \in [1, N] \tag{2}$$

where h_i is the hidden state vector, i.e., the context representation of the input sentence in the i-th layer, and N is the number of transformer blocks.

3.2. NER downstream task layer

The NER downstream task layer aims to combine the prior knowledge in the pre-trained language model with the fine-tuning of downstream tasks, thereby expanding the quantity and improving the quality of labeled data. It exploits the interdependence between source and target BERT to construct an entity labeling model, namely source-target BERT, and automatically adjusts the value λ to reduce the labeling loss of the model. Fig. 3 shows the details of the entity tagging task, which comprise two steps:

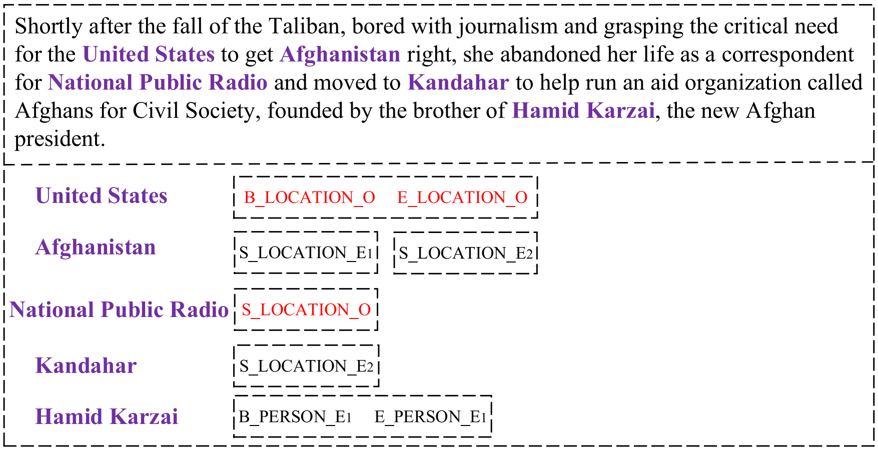
Step 1: Perform pre-training on a large amount of unlabeled text data to complete the corpus labeling task of the source sentence.



Shortly after the fall of the Taliban, bored with journalism and grasping the critical need for the United States to get Afghanistan right, she abandoned her life as a correspondent for National Public Radio and moved to Kandahar to help run an aid organization called Afghans for Civil Society, founded by the brother of Hamid Karzai, the new Afghan president.

United States	B_LOCATION_E1 E_LOCATION_E1	B_LOCATION_E2 E_LOCATION_E2
Afghanistan	S_LOCATION_E1	S_LOCATION_E2
National Public Radio	S_LOCATION_E1	S_LOCATION_E2
Kandahar	S_LOCATION_E1	S_LOCATION_E2
Hamid Karzai	B_PERSON_E1 E_PERSON_E1	B_PERSON_E2 E_PERSON_E2

Fig. 4. Example of source BERT, where we generate a label for the different tokens.



Shortly after the fall of the Taliban, bored with journalism and grasping the critical need for the United States to get Afghanistan right, she abandoned her life as a correspondent for National Public Radio and moved to Kandahar to help run an aid organization called Afghans for Civil Society, founded by the brother of Hamid Karzai, the new Afghan president.

United States	B_LOCATION_O E_LOCATION_O	
Afghanistan	S_LOCATION_E1	S_LOCATION_E2
National Public Radio	S_LOCATION_O	
Kandahar	S_LOCATION_E2	
Hamid Karzai	B_PERSON_E1 E_PERSON_E1	

Fig. 5. Example of target BERT, where the labels of certain words are changed after fine-tuning.

Step 2: Use the pre-trained parameters of step 1 as the initial parameters and combine the corresponding supervision targets to fine-tuned the initial parameters to make them suitable for the corpus tagging task of the target sentence.

Corresponding to the two steps, source-target BERT comprises two modules: (a) source BERT and (b) target BERT.

source BERT. It uses the BERT-base-case model as the text representation model to treat a structured text input as a single continuous sequence of tokens. For a given unsupervised source sentence sequence $s_S = (w_{s1}, w_{s2}, \ldots, w_{sn})$, the source sentence tag sequence $L_S = (L_{s1}, L_{s2}, \ldots, L_{sn})$ is output, after source BERT encoding, where sn is the length of Tok_S. Fig. 4 shows an example of source BERT. There are five entities in the sentence: "United States," "Afghanistan," "National Public Radio," "Kandahar," and "Hamid Karzai." We perform sequence labeling on these entities.

target BERT. If the target sentence has a certain amount of training data, target BERT can further improve the impact of token labeling. The source sentence tag sequence $L_S = (L_{s1}, L_{s2}, \ldots, L_{sn})$ and target sentence sequence $s_T = (w_{t1}, w_{t2}, \ldots, w_{tn})$ are input into target BERT, and the real tag sequence of the target sentence is the target for training. Thus, the token sequence tagger can be obtained. In this case, tn is the length of Tok_T. Fig. 5 shows an example of target-BERT. We fine-tuned the labels of entities in the sentences based on downstream tasks. As can be seen, the labels "Kandahar" and "Hamid Karzai" were changed, and the unrelated entities "United States" and "National Public Radio" were excluded in the fine-tuning phase. We can perform the RC task based on the entity labels after fine-tuning.

Loss function. The overall label loss function $\mathscr{L}_E$ of source-target BERT includes two parts: the label loss of the source sentence $\mathscr{L}_S$ and that of the target sentence $\mathscr{L}_T$, which are the auxiliary and main losses, respectively. Further, $\lambda \in [0, 1]$ is a scaling factor for balancing source BERT and target BERT. The detailed operations of the source-target BERT loss function are as follows:

$$\mathscr{L}_E = \mathscr{L}_T + \lambda \mathscr{L}_S \tag{3}$$

$$\mathscr{L}_{\mathrm{T}} = -\frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} \left(\mathrm{y}_{\mathrm{ti}}^{\mathrm{start}} \log\left(\mathrm{P}_{\mathrm{ti}}^{\mathrm{start}}\right) + \mathrm{y}_{\mathrm{ti}}^{\mathrm{end}} \log\left(\mathrm{P}_{\mathrm{ti}}^{\mathrm{end}}\right) \right) \tag{4}$$

$$\mathscr{L}_S = -\frac{1}{n} \sum_{i=1}^{n} \left(y_{si}^{start} \log\left(P_{si}^{start}\right) + y_{si}^{end} \log\left(P_{si}^{end}\right) \right) \tag{5}$$

$$P_i^{start} = \sigma(\omega_{start} h_i + b_{start}) \tag{6}$$

$$P_i^{end} = \sigma(\omega_{end} h_i + b_{end}) \tag{7}$$

where n is the length of a sentence sequence; y_{ti}^{start} (y_{ti}^{end}) and y_{si}^{start} (y_{si}^{end}) represent the ground truth values that the i-th token is at the start (end) position of the correct candidate entity in the target and source sentences, respectively; P_{ti}^{start} (P_{ti}^{end}) and P_{si}^{start} (P_{si}^{end}) represent the probabilities of identifying the i-th token in the input sequence at the start (end) position of the predicted entity in the target and source sentences,

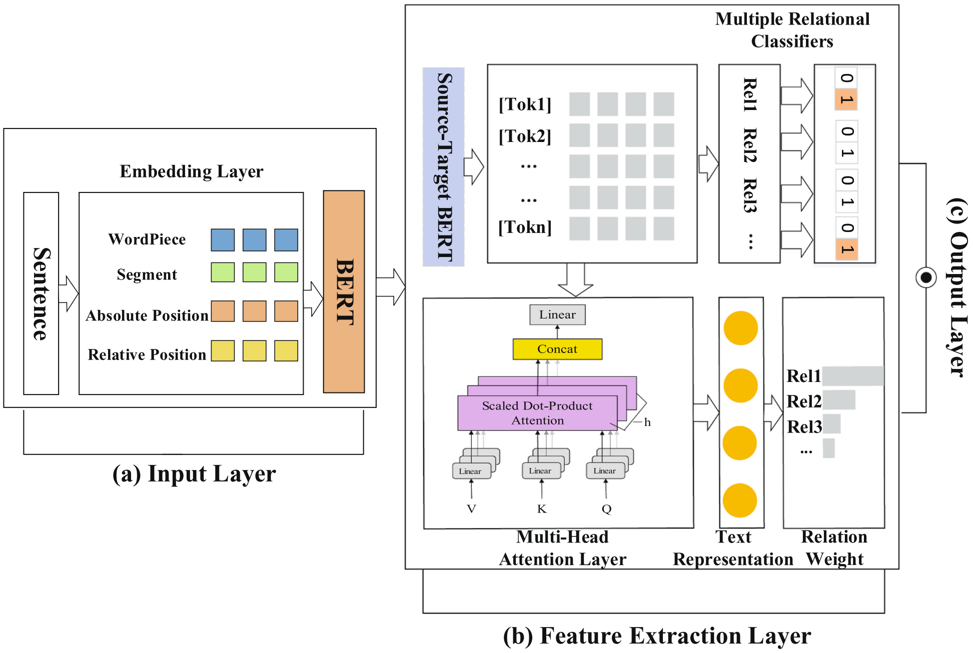


Fig. 6. Schematic of the OREM architecture. The input of the OREM model is the encoding result of the parameter-sharing layer. source-target BERT extracts the entity label representation for each token. Multi-relation classifiers are used to predict whether there is a certain relation between two entities. MHA extracts the multiple relation representations of the entity pair. Thereafter, the product of the binary classification result and relation weight is used as the output result of the model.

respectively; h_i is the encoded representation of the i-th token in the input sequence, produced by the parameter-sharing layer; ω represents the trainable weight; b is the bias; and σ is the sigmoid activation function.

Scaling factor. To ensure the quality of the existing and newly added labels, the value of $\lambda$is automatically adjusted to reduce the tagging loss of source-target BERT. When the entity label in the source sentence is similar to that in the target sentence, the λ value is increased; otherwise, only the labeling loss $\mathscr{L}_T$ of the target sentence is used, as λ may decrease to zero. This is mainly because, if the tagging quality of the source sentence is high, $\mathscr{L}_S$ is reliable; conversely, if the tagging quality of the source sentence is low, it is assumed that $\mathscr{L}_S$ does not affect the main loss $\mathscr{L}_T$.

$$L_S = Contact(B_S^P, B_S^C, B_S^E) \tag{8}$$

$$L_T = Contact(B_T^P, B_T^C, B_T^E) \tag{9}$$

$$\lambda = \max\{0, \cos\langle L_S, L_T\rangle\ \} \tag{10}$$

where L_S and L_T are the label representations of the source and target sentences, respectively, which are spliced by the position information, entity type, and entity information. Moreover, B_S^P, B_S^C, B_S^E, and B_T^P, B_T^C, B_T^E denote the position information, entity type, and entity information of each label in source BERT and target BERT, respectively. The position information is used to indicate the position of the token in the entity. Specifically, P_B, P_I, and P_E indicate that it is located at the start, intermediate, and end positions of the entity, respectively, whereas P_S indicates a single-word entity. The entity type denotes the type of entity to which the token belongs. The entity information is used to indicate whether the current token belongs to the head or tail entity, where E_1 represents the head entity, E_2 represents the tail entity, and E_O indicates a non-entity. For example, a token in a sentence may be labeled as “$B - PER - E_1$,′ ′ where B, PER, and E_1 represent the start position of the entity, a natural person, and the head entity, respectively.

3.3. RC downstream task layer

The NER downstream task layer considers only the entity label, it does not consider the overlapping relations between entity pairs. Therefore, the RC downstream task layer aims to predict the possible relation between each token pair through the OREM model. Fig. 6 shows the details of the overlapping relation extraction task, which comprises three steps:

Step 1: Use the pre-trained language model BERT to encode the sentence.

Step 2: Use multiple relational classifiers to predict whether the entity pair has a certain relation. Next, use MHA to find the relation weight of a certain relation in the sentence.

Step 3: Combine the classification result of the multiple relational classifiers with the relation weights and set the relation threshold to output the true relation in the sentence.

Corresponding to the three steps, the OREM model comprises three layers: (a) input layer, (b) feature extraction layer, and (c) output layer.

Input layer. For any given sentence, the input embedding for tokens is fed to BERT to learn the contextual representation of each token. The input embedding of each token is the sum of four types of embeddings: word embedding, segmentation embedding, absolute position embedding, and relative position embedding.

- *Word embedding.* The word segmentation granularity of WordPiece (Wu et al., 2016) embedding lies between word-level and character-level sequences. For example, walking can be categorized as the tag “walk” and ##ing. This enables the model to obtain certain inferences based on the word structure: verbs that begin with “walk” have similar semantic functions, and verbs that end with *-ing* have similar grammatical functions. This method is mainly used to avoid situations in which words are not sufficiently frequent and have not been added to the dictionary, and, finally, can only be replaced by [UNK].
- *Segmentation embedding.* The sentence embedding is added to each token to indicate whether it belongs to sentence A or sentence B. If

the input contains only one sentence, all its segment IDs are set to zero. In addition, tokens [CLS] and [SEP] are placed at the start and end of the sentence, respectively.

- *Absolute position embedding.* A vector can be added for each input token, which helps determine the actual position of each token in the sentence.
- *Relative position embedding.* The difference between the absolute positions of the two tokens is defined as the relative position. According to the direction of the relative position, it needs to be multiplied by 1 (if two words are in a positive order in the sentence), −1 (if two words are in reverse order in the sentence), or 0 (for the same word).

Feature extraction layer. On the basis of the input layer, the extraction of deeper semantic features is realized. In terms of the feature extraction, we mainly focus on the entity label feature and the relation feature between two entities in the sentence. Entity label features can be extracted using the source-target BERT model. For a more comprehensive description of the source-target BERT model, readers may refer to Section 3.2. Relation features can be extracted through multiple relational classifiers and MHA.

- *Multiple Relational Classifiers.* As a sentence may contain multiple relations, we design multiple relational classifiers, which are used to convert the RC task of entity pairs in a sentence into multiple binary classification problems. We perform a bilinear transformation on the entity pair and then use the softmax activation function to predict the probability distribution of whether the entity pair contains a certain relation type.
- *MHA.* Based on multiple binary classification results, the MHA mechanism is used to obtain the new sentence representation, and then the sigmoid activation function is used to obtain the matching degree between the sentence representation and the target relation type.

Output layer. After obtaining the feature representation, once the matching degree is calculated, a score for a certain relation can be obtained. Next, we set a global relation threshold θ that maximizes the evaluation metrics. If the relation score is greater than θ, relation r_k is returned as an associated relation between the entity pair in the sentence.

We employ the cross-entropy loss function to define the loss of OREM, which is denoted as $\mathscr{L}_R$.

$$\zeta_R = \sum_{k=1}^{m} \sum_{\substack{i,j=1\\ i\neq j}}^{n} -\log P(r_k|(e_i e_j),\theta) \tag{11}$$

where m is the number of relations for $(e_i,\ e_j)$, n is the number of entities in the sentence, r_k represents the ground truth relation type for $(e_i,\ e_j)$, and θ is the relation threshold. The details of the relation threshold are presented in Section 5.3.

3.4. Training

As can be seen from the overall architecture of BERT-JEORE, the parameters are shared in the model except for the downstream task layers, which enables BERT-JEORE to learn the joint features of entities and overlapping relations. Finally, the source-target BERT loss $\mathscr{L}_E$ and OREM loss $\mathscr{L}_R$ are combined to obtain the BERT-JEORE loss $\mathscr{L}_{all}$.

$$\mathscr{L}_{all} = \mathscr{L}_E + \mathscr{L}_R \tag{12}$$

where $\mathscr{L}_E$ and $\mathscr{L}_R$ are defined in Eqs. (3) and (11), respectively. We minimize $\mathscr{L}_{all}$ and train the entire BERT-JEORE in an end-to-end manner.

Table 1
Statistics of the datasets. Note that a sentence can belong to both EPO and SEO.

Category	NYT		WebNLG	
	Train	Test	Train	Test
normal	36,868	3,244	1,716	266
EPO	9,782	978	227	26
SEO	14,306	1,262	3,261	435
Sentence	56,195	5,000	5,019	703
Relation	24		246	

4. Experimental setup

4.1. Datasets

To evaluate the effectiveness of the proposed model, we conducted experiments on two datasets: NYT (Riedel et al., 2010) and WebNLG (Gardent et al., 2017).

NYT: It is a news corpus produced by a distant supervision method (Mintz et al., 2009). There are 24 predefined relation types. We used the pre-processed dataset[1] released by Zeng et al. (2018), in which sentences with more than 100 words and those without positive triplets were filtered out. After filtering, the training, test, and validation sets contained 56,195, 5,000, and 5,000 sentences, respectively.

WebNLG: It was originally created for the task of natural language generation (NLG) and subsequently used for triplet extraction (Zeng et al., 2018). This dataset comprises 246 predefined relation types, and each instance contains multiple triplets and several standard sentences (written by humans). We used the pre-processed dataset[2] released by Zeng et al. (2018), in which the first standard sentence was selected as the training corpus. If all the entities of the triplets were not found in this standard sentence, the instances would be removed. In the pre-processed dataset of Zeng et al. (2018), the original training set was divided into training and validation sets, and the original development set was used as the test set. After filtering and splitting, the training, test, and validation sets contained 5,019, 500, and 703 sentences, respectively.

The statistics of the two datasets are summarized in Table 1. We divided the sentences into three categories: normal, EPO, and SEO. It is worth noting that a sentence can belong to both EPO and SEO. Further details about the datasets are presented in Appendix A.

4.2. Implementation details

Implementation. We constructed our model using Keras (Gulli & Pal, 2017) on a Linux machine with the NVIDIA TITAN V (12 GB) GPU.

BERT. Because pre-training is computationally expensive and we aim to prove the effectiveness of the model by fine-tuning the NER and RC tasks, we reused the BERT-base-case model released by Google in the parameter-sharing layer as the basis for the experiment. It consists of a 12-layer encoding network, the size of the hidden state is 768, with 12 self-attention heads, containing 110 M parameters. The parameters of these pre-trained models were used as initialization parameters for different downstream tasks.

Hyperparameters. All the hyperparameters were tuned on the validation set. The relation thresholds of the two datasets were set to 0.6 and 0.4, respectively. Further, the learning rate were set to 1e-5 and 2e-5, respectively. We chose the batch size in $[16, 32, 64]$. We also adopted 40 training epochs with the early stopping mechanism to prevent overfitting of the model. Specifically, we stopped the training process and saved the optimal model when the performance on the validation set did

[1] https://drive.google.com/open?id=10f24s9gM7NdyO3z5OqQxJgYud4NnCJg3.

[2] https://drive.google.com/open?id=1zISxYa-8ROe2Zv8iRc82jY9QsQrfY1Vj.

Table 2
Parameters of the baseline models.

Model	Parameters				
	Cell unit number	Word embedding dimension	Batch size	Learning rate	Epoch
NovelTagging	300	300	50	0.001	100
CopyRE	1,000	100	100	0.001	100
GraphRel	256	300	100	0.0008	100
CopyMTL	1,000	100	100	0.001	40
CopyR$_{RL}$	1,000	100	100	0.001/ 0.0005	50

Table 3
Results of different models on the NYT and WebNLG datasets, where the numbers in boldface indicate the best results. We reported the mean and standard deviation by conducting 10 runs. The significance levels are denoted by asterisks, where * indicates the significance at p $<$ 0.01 compared to *CopyR$_{RL}$*. As GraphRel is an not open-source model, we have not reproduced its results.

Model	NYT			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1
NovelTagging (Zheng et al., 2017)	0.624	0.317	0.420	0.525	0.193	0.283
OneDecoder (Zeng et al., 2018)	0.594	0.531	0.560	0.322	0.289	0.305
MultiDecoder (Zeng et al., 2018)	0.610	0.566	0.587	0.377	0.364	0.371
GraphRel1p (Fu et al., 2019)	0.629	0.573	0.600	0.423	0.392	0.407
GraphRel2p (Fu et al., 2019)	0.639	0.600	0.619	0.447	0.411	0.429
CopyMTL–One (Zeng et al., 2020)	0.727	0.692	0.709	0.578	0.601	0.589
CopyMTL-Mul (Zeng et al., 2020)	0.757	0.687	0.720	0.580	0.549	0.564
CopyR$_{RL}$ (Zeng et al., 2019)	0.779	0.672	0.721	0.633	0.599	0.616
NovelTagging (ours)	0.575±0.069	0.327±0.013	0.415±0.007	0.501±0.035	0.190±0.005	0.275±0.011
CopyRE (ours)	0.614±0.006	0.567±0.004	0.590±0.004	0.368±0.008	0.356±0.007	0.361±0.006
CopyMTL-Mul (ours)	0.696±0.011	0.637±0.006	0.656±0.008	0.560±0.019	0.548±0.027	0.554±0.022
CopyR$_{RL}$ (ours)	0.727±0.005	0.687±0.005	0.706±0.003	0.581±0.007	0.586±0.004	0.583±0.003
BERT-JEORE	**0.885±0.003***	**0.846±0.003***	**0.865±0.002***	**0.791±0.005***	**0.914±0.004***	**0.848±0.003***

not show any improvement for at least five consecutive epochs. We used the exponential moving average with a decay rate of 0.999 to ensure a stable improvement in the training results. The head number in MHA was set to 4. For training, we used Adam (Kingma & Ba, 2015) to optimize the parameters. For a fair comparison, the maximum length of the input sentence in our model was set to 100 words, as in previous studies (Fu et al., 2019).

4.3. Baseline models and evaluation metrics

To verify the effectiveness of the proposed model, we compared it with the following baseline models, the parameters of which are listed in Table 2.

NovelTagging (Zheng et al., 2017): This model applies a novel tagging scheme to the joint extraction of entities and relations. Simultaneously, the bias objective function is used to enhance the correlation between entities, and the influence of invalid labels is reduced. However, this model cannot extract triplets with overlapping entities.

CopyRE (Zeng et al., 2018): This model is a Seq2Seq model that uses a copy mechanism to generate a triplet by jointly copying a relation from the relation set and an entity pair from the source text in a sequential manner. However, this model can only copy the last word of the entity.

GraphRel (Fu et al., 2019): This model is an end-to-end relation extraction model that uses GCNs to learn named entities and relations jointly. It considers the interaction between entities and relations via two-phase GCN.

CopyMTL (Zeng et al., 2020): This model uses a multi-task learning framework that is equipped with a copy mechanism for predicting multi-token entities. It overcomes the inability of CopyRE (Zeng et al., 2018) to (1) distinguish between head and tail entities and (2) predict multi-token entities.

CopyR$_{\mathrm{RL}}$ (Zeng et al., 2019): This model applies RL to the Seq2Seq model to learn the extraction order of multiple relational facts in a sentence. This allows the model to generate triplets freely to obtain higher rewards.

We used the precision, recall, and F1 score (Liu, Zhou, Wen, & Tang, 2014) to assess the extracted triplets. When the relation type and the head and tail entities were correct, the extracted triplet was considered correct. Further, if the head and tail entities were offset correctly, the extracted entities were correct. We performed the same experiment 10 times. The average and standard deviation values are listed in Table 3.

5. Results and discussion

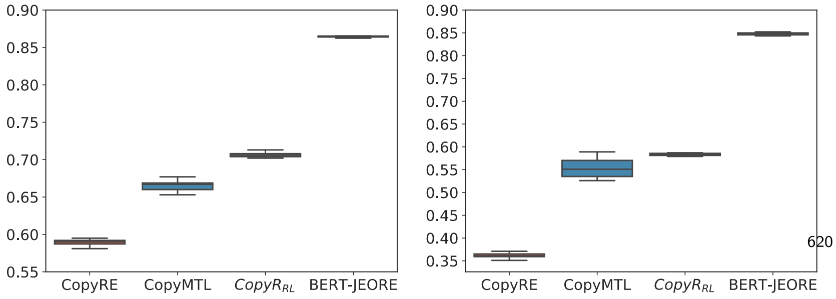
The following subsections present the main results, discuss the experimental results, provide the detailed analysis and error analysis, and describe ablation, and case studies, respectively. Our experiments achieved the following six research targets:

(1) Evaluating the overall performance and stability of BERT-JEORE.
(2) Discussing the performance and mutual influence of model subtasks as well as the advantages of pre-training language models.
(3) Analyzing the performance of the model under different relation threshold, overlapping types and degrees, as well as showing the ability of the model to handle overlap relations.
(4) Analyzing the performance of different elements in the triplets and determining the key factors that affect the model extraction of triplets under different datasets.
(5) Analyzing the impact of the main modules on the model performance.
(6) Showing the actual effect of the model under different datasets.

5.1. Main results

Table 3 summarizes the experimental results[3] of different models for relational triplet extraction on two datasets. BERT-JEORE exhibited the best performance in this task, achieving 86.5% (95 %CI: 86.3% to 86.7%) and 84.8% (95% CI: 84.5% to 85.1%) in terms of the F1 score, respectively. Moreover, compared with *CopyR$_{RL}$*, our model achieves a significant increase in F1. Specifically, BERT-JEORE achieves

[3] As NovelTagging is significantly superior to the previous methods, we did not include further comparisons.



(a) NYT (b) WebNLG

Fig. 7. Boxplots of F1 scores for four models (CopyRE, CopyMTL, *CopyR*$_{RL}$, and BERT-JEORE) on the NYT and WebNLG datasets. As GraphRel is not an open-source model, it is difficult to use the same indicator to compare our model with the existing models. Thus, we list only the F1 scores of the three open-source baseline models and our model. It can be clearly observed that the overall level of BERT-JEORE is higher and that its distribution is more concentrated.

Table 4
F1 scores of two subtasks. The value in boldface indicate the best results for each dataset. The significance levels are denoted by asterisks, where * indicates the significance at $p < 0.01$ compared to *CopyR*$_{RL}$.

Dataset	Model	Entity	Relation
NYT	CopyRE (Zeng et al., 2018)	0.647	0.846
	CopyMTL (Zeng et al., 2020)	0.756	0.869
	CopyR$_{RL}$ (Zeng et al., 2019)	0.873	0.890
	BERT-JEORE	**0.930***	**0.925***
WebNLG	CopyRE (Zeng et al., 2018)	0.595	0.767
	CopyMTL (Zeng et al., 2020)	0.782	0.797
	CopyR$_{RL}$ (Zeng et al., 2019)	0.882	0.804
	BERT-JEORE	**0.956***	**0.926***

improvements of 15.9% and 26.5% in the F1 score, respectively. This verifies the effectiveness of the proposed model.

We can also observe from the table that there is a significant difference between the performances of all the models on the NYT and WebNLG datasets. This is due to the difference in the data distributions. More precisely, as indicated in Table 1, the NYT dataset mainly consists normal sentences, whereas most sentences in the WebNLG dataset belong to the overlapping relation classes. This inconsistent data distribution of the two datasets leads to comparatively better performance on NYT and worse performance on WebNLG for all the models, which reflects the difficulty in the task of extracting overlapping relational triplets.

Fig. 7 shows boxplots of the F1 scores for four models. It can be seen that the F1 value of BERT-JEORE is the largest and its distribution is the most concentrated; thus, BERT-JEORE shows the best performance and stability. We attribute the gains of BERT-JEORE to the following factors. (1) It can identify long-distance relations through pre-trained language models. (2) It uses the cross-entropy loss function to predict the start and end positions of the entity, thereby determining the length of the entity and predicting the entire entity. (3) An unlimited number of triplets can be extracted through multiple binary classifiers and MHA. In addition, there are some differences between the models, leading to differences in their extraction performance. Further details are presented in Appendix B.

5.2. Discussion on the experimental results

In this subsection, we focus on the following three questions about our experiments:

RQ1: Why are NER and RC improved?

The previous results (Table 3) confirmed that BERT-JEORE outperforms the other baseline models. To understand why BERT-JEORE is superior the other baseline models, we analyzed their NER and RC ability. The experiment results are summarized in Table 4.

Table 5
Chi-square test results to determine whether NER affects RC on the WebNLG dataset.

	RC-correct	RC-error
NER-correct	1,437	65
NER-error	230	76
Total	1,667	141

For the NER subtask, the F1 score of BERT-JEORE increased by 5.7% on the NYT dataset and 7.4% on the WebNLG dataset. This is mainly because our model uses the cross-entropy loss function to predict the start and end positions of the entity, thereby effectively determining the entity boundaries. Furthermore, the evaluation metrics used in our model are stricter than those used in the other models. Only when an entity with a relation is identified and the entity type is correct, the entity is considered correct.

For the RC subtask, the F1 score of BERT-JEORE increased by 3.5% on the NYT dataset and by 12.2% on the WebNLG dataset. This is attributed to the fact that BERT-JEORE uses multiple binary classifiers and MHA to generate triplets with different relations, thereby ensuring the diversity and accuracy of the relation classification results.

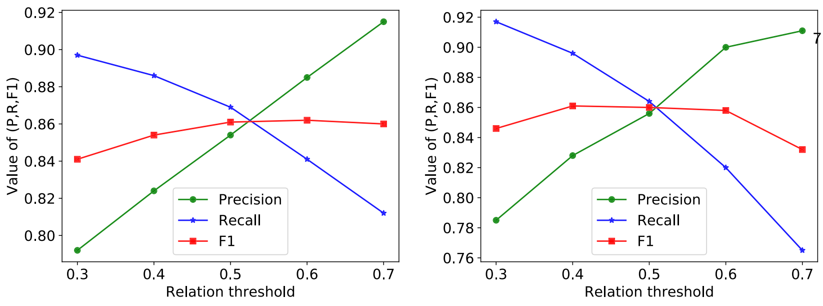
RQ2: Does NER affect RC?

The chi-square test can be used to analyze the relation between two variables and determine whether the value of one variable affects that of the other. We used this method to determine whether NER affects RC in BERT-JEORE. We divided NER into two states: NER-correct and NER-error. Similarly, we divided RC into two states: RC-correct and RC-error. We determined whether NER affects RC under $\alpha = 0.01$. We used the SPSS v24 software for data analysis. The results are summarized in Table 5.

We assumed that NER is not related to RC. Following the data analysis, $\chi^2 > \chi^2_{0.01}(1)$. Therefore, the original hypothesis was rejected, and we could concluded that NER and RC are related. Furthermore, we can observe from Table 5 that when NER was correct, the RC correctness rate was 95.67%, and when NER was incorrect, the RC correctness rate was 75.16%. This indicates that accurate entity recognition helps classify the relation in the next step. Table 5 also presents special situations of NER miss and RC hit, which arise mainly because there are incorrect head or tail entities in the predicted triplets. For example, in the sentence “The members of Apollo 8 were Buzz Aldrin who was backup pilot, commander Frank Borman and William Anders who retired on September 1st, 1969 ⋯,” there exists a ground truth relation (*William Anders*, *was a crew memberof*, *Apollo 8*). However, our model predicted not only this triplet but also another triplet

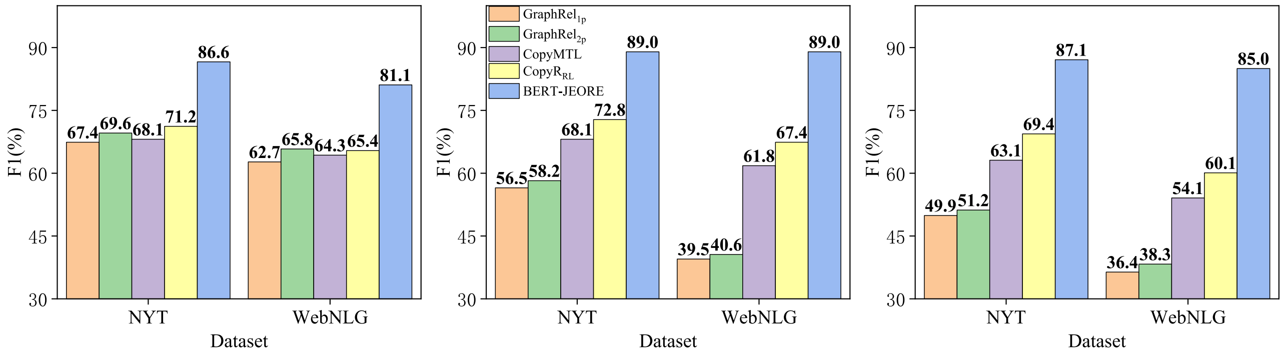
Table 6
Results of the test sets for overlapping relation extraction for varying amounts of training data available to our model and other models.

% of training set	10%			20%			50%			100%		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
						NYT						
CopyRE	0.455	0.390	0.420	0.510	0.436	0.470	0.562	0.505	0.532	0.614	0.567	0.590
CopyMTL	0.399	0.347	0.372	0.539	0.455	0.493	0.630	0.568	0.598	0.696	0.637	0.656
CopyR$_{RL}$	0.545	0.492	0.517	0.625	0.569	0.596	0.699	0.647	0.672	0.727	0.687	0.706
BERT-JEORE	**0.713**	**0.498**	**0.586**	**0.737**	**0.746**	**0.741**	**0.781**	**0.846**	**0.812**	**0.885**	**0.846**	**0.865**
						WebNLG						
CopyRE	0.179	0.172	0.176	0.237	0.230	0.234	0.304	0.290	0.297	0.368	0.356	0.361
CopyMTL	0.150	0.118	0.132	0.328	0.304	0.315	0.434	0.414	0.424	0.560	0.548	0.554
CopyR$_{RL}$	0.280	0.267	0.274	0.343	0.350	0.347	0.481	0.484	0.483	0.581	0.586	0.583
BERT-JEORE	**0.614**	**0.431**	**0.506**	**0.804**	**0.710**	**0.754**	**0.818**	**0.805**	**0.812**	**0.791**	**0.914**	**0.848**



(a) NYT (b) WebNLG

Fig. 8. Performance with different relation thresholds.



(a) F1 of normal Class (b) F1 of EntityPairOverlap Class (c) F1 of SingleEntityOverlap Class

Fig. 9. F1 score on sentences with different overlapping types.

(*Buzz Aldrin*, *was a crew member of*, *Apollo 8*), resulting in a situation where the relation was correct but the entity recognition was incorrect. This may be due to the interference word "members" in the sentence, which caused the model to mistakenly assume that "Buzz Aldrin" was a crew member.

RQ3: Does the pre-trained language model reduce the need for labeled data?

Table 6 summarizes the performance of each model when the amount of training data is reduced. As can be seen, our model is more effective when the data resources are scarce. This further supports our argument that training through the pre-trained language model can significantly reduce the amount of manual labeling data required for relation extraction tasks.

5.3. Detailed analysis

Relation threshold. We changed the relation threshold θ from 0.3 to 0.7 to analyze the impact of different relation thresholds on the performance of BERT-JEORE. The predicted results are shown in Fig. 8. When θ was too large, the recall deteriorated, and when $\theta$was too small, the precision of the prediction was affected. BERT-JEORE obtained a reliable balance between the precision and the recall, and achieved the best F1 scores on the NYT and WebNLG datasets when $\theta = 0.6$ and $\theta = 0.4$, respectively.

Overlapping types and degrees. To further investigate the ability of BERT-JEORE to extract overlapping relational triplets, we conducted two extended experiments on different overlapping types and degrees.

The detailed results for three different overlapping types are shown in Fig. 9. As can be seen, all the models could achieve competitive

Table 7
F1 score on sentences with different overlapping degrees.

Method	NYT					WebNLG				
	N = 1	N = 2	N = 3	N = 4	N ⩾ 5	N = 1	N = 2	N = 3	N = 4	N ⩾ 5
GraphRel$_{1p}$	69.1	59.5	54.4	53.9	37.5	63.8	46.3	34.7	30.8	29.4
GraphRel$_{2p}$	71.0	61.5	57.4	55.1	41.1	66.0	48.3	37.0	32.1	32.1
CopyMTL	67.9	69.4	65.7	69.6	44.5	63.5	57.1	58.8	50.5	45.4
CopyR$_{RL}$	71.7	72.6	72.5	77.9	45.9	63.4	62.2	64.4	57.2	55.7
BERT-JEORE	**87.4**	**85.7**	**88.2**	**92.1**	**75.2**	**81.7**	**84.0**	**87.1**	**85.7**	**82.2**

Table 8
Ablation study of BERT-JEORE on the NYT and WebNLG datasets. We reported the mean and standard deviation by conducting 10 runs. Here, ** denotes the significance at p < 0.05 compared to the ablation model.

Setting	NYT			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1
BERT-JEORE	**0.885 ± 0.003****	**0.846 ± 0.003****	**0.865 ± 0.002****	**0.791 ± 0.005****	**0.914 ± 0.004****	**0.848 ± 0.003****
w/o source-target BERT	0.835 ± 0.002	0.842 ± 0.004	0.838 ± 0.002	0.741 ± 0.003	0.900 ± 0.003	0.813 ± 0.002
w/o MHA	0.826 ± 0.003	0.792 ± 0.005	0.809 ± 0.004	0.732 ± 0.004	0.856 ± 0.003	0.789 ± 0.004

performances in the regular normal triplet detection. When handling EPO triplets, certain baseline models performed worse, and their performance was poorer in the case of SEO. These observations indicate that the SEO triplet is the most challenging case. Our model always showed the best performance for all types (especially SEO) and performed significantly better than the previous methods. For the normal class, BERT-JEORE outperformed *CopyR$_{RL}$* by 15.4% and 15.7% in terms of the F1 scores, respectively. For EPO class, BERT-JEORE outperformed *CopyR$_{RL}$* by 16.2% and 21.6% in terms of the F1 scores, respectively. For the SEO class, BERT-JEORE outperformed *CopyR$_{RL}$* by 17.7% and 24.9% in terms of the F1 scores, respectively. This is mainly because all the entities can be associated with other entities when BERT-JEORE predicts the relation between entity pairs. Therefore, the extraction of overlapping relations is not a problem.

We also validated the ability of BERT-JEORE to extract relational triplets from sentences with different overlapping degrees. The results are summarized in Table 7. As can be seen, the performance of most of the baseline models declined with an increase in the number of relational triplets in a sentence. By contrast, the performance of BERT-JEORE improved considerably when extracting multiple triplets. In particular, when a sentence contained more than five overlapping relations, BERT-JEORE significantly outperformed the existing baseline systems. Specifically, BERT-JEORE outperformed *CopyR$_{RL}$* by 29.3% and 26.5% on the NYT and WebNLG datasets, respectively. Thus, our model is more suitable for handling complex overlapping relations than the baseline models.

5.4. Ablation study

To further investigate the effects of our model in terms of source-target BERT and MHA, we conducted ablation experiments on the NYT and WebNLG datasets. Table 8 summarizes the ablation results.

As can be seen, both parts can assist BERT-JEORE in jointly extracting entities and overlapping relations, where the MHA mechanism seems to play a more significant role. When we removed source-target BERT, the performance decreased by 2.7% and 3.5% in terms of the F1 score, respectively. The main reason for this is that source-target BERT can fine-tune the labeled data to improve the accuracy of data tagging, reduce the impact of noise during the distant supervision process, and provide better support for relation extraction. Furthermore, upon removing MHA, the performance decreased by 5.6% and 5.9% in terms of the F1 score, respectively. This illustrates that MHA can naturally capture the overlapping relations in sentences and obtain the weights of the overlapping relations.

Table 9
Results for relational triplet elements.

Element	NYT			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1
E_1	0.947	0.890	0.918	0.949	0.957	0.953
E_2	0.940	0.894	0.916	0.931	0.942	0.936
R	0.953	0.898	0.925	0.921	0.932	0.926
(E_1, R)	0.922	0.868	0.894	0.826	0.920	0.870
(R, E_2)	0.921	0.868	0.894	0.890	0.923	0.906
(E_1,E_2)	0.894	0.858	0.875	0.820	0.928	0.871
(E_1, R, E_2)	0.885	0.846	0.865	0.791	0.914	0.848

5.5. Error analysis

To explore the factors that affect the extraction of relational triplets by BERT-JEORE, we analyzed the performance of different elements in the triplet (E_1, R, E_2). Table 9 summarizes the results for the different relational triplet elements.

The performance gap between E_1 and E_2 was consistent with that between (E_1,R) and (R,E_2) for NYT, demonstrating the effectiveness of our model in identifying both subject and object entity mentions. Moreover, there was only a trivial difference between the F1 scores on (E_1,E_2) and (E_1,R,E_2), but a clear difference between those on (E_1,R,E_2) and $(E_1,R)/(R,E_2)$, suggesting that most of the relations for the entity pairs in the extracted triplets were identified correctly, whereas certain extracted entities failed to form a valid relational triplet. This implies that identifying relations is easier than identifying entities.

For WebNLG, the performance gap between (E_1,E_2) and (E_1,R,E_2) was 2.3%, which was greater than that between (E_1,E_2) and (E_1,R,E_2) for NYT (1.0%). This suggests that it is more challenging to identify relations in WebNLG. We attribute this difference to the different number of predefined relations contained in the two datasets (i.e., 24 in NYT and 246 in WebNLG).

5.6. Case study

For elucidation, we list a few examples from the NYT and WebNLG datasets in Tables 10 and 11. As can be seen, BERT-JEORE could identify one or more triplets in each sentence. This demonstrates the effectiveness of the proposed model in solving the problems of extracting entities

Table 10
Case study of BERT-JEORE on the NYT dataset. Here, (*Iraq*, *Baghdad*) and *Brooklyn* are an overlapping entity pair and an overlapping entity that appear frequently in the NYT dataset, respectively.

Sentence S1	**Yoshi Tsuji** runs several cooking schools in *Japan* and in **Europe**; **Kazuki Kondo**, dean of the *Osaka* school, joined us for dinner.
BERT-JEORE	(*Japan*, /location/location/contains, *Osaka*)
Ground truth	(Japan, /location/location/contains, Osaka)
Sentence S2	If the initial phase is successful, he said, it will be expanded so that 15,000 new troops will guard pipelines across northern *Iraq*, all the way south to *Baghdad* and north to the **Turkish** border.
BERT-JEORE	(*Iraq*, /location/country/capital, *Baghdad*) (*Iraq*, /location/location/contains, *Baghdad*)
Ground truth	(Iraq, /location/country/capital,Baghdad) (Iraq, /location/location/contains,Baghdad)
Sentence S3	"I'm not funny, " *Jonathan Safran Foer* announced when I walked into his office in the *Park Slope* neighborhood of *Brooklyn.*
BERT-JEORE	(*Brooklyn*,/location/location/contains, *Park Slope*) (*Park Slope*, /location/neighborhood/neighborhood_of, *Brooklyn*)
Ground truth	(Jonathan Safran Foer, /people/person/place_lived, Brooklyn)

and overlapping relations.

As indicated in Table 10, the first and second examples belong to the normal and EPO classes, respectively, and the third example belong to the SEO class. We can observe the following. (1) In the first and second examples, the irrelevant entities "Yoshi Tsuji," "Europe," "Kazuki Kondo," and "Turkish" are excluded, which help to reduce the impact of irrelevant entities and improve the relation extraction performance of BERT-JEORE. (2) In the third example, the annotated triplets are not always the ground truth, which could affect the evaluation of our model. For example, the relational triplets (*Brooklyn*, */location/location/contains*, *Park Slope*) and (*Park Slope*, */location/neighborhood/neighborhood_of*, *Brooklyn*) should have been annotated in the sentence but were omitted. BERT-JEORE identified the entity "Park Slope" that did not appear in the ground truth and identified these triplets through overlapping relation extraction. This observation demonstrates that our model could extract more relational triplets. In addition, our model lost the triplet (*Jonathan Safran Foer*, */people/person/place_lived*, *Brooklyn*) because the residence relation was not mentioned in the context of the sentence; hence, our model excluded this residence relational triplet based on the semantic information of the sentence, alleviating the noise problem of distant supervision.

As indicated in Table 11, the first example belong to the normal class, the second example belong to the EPO and SEO classes, and the third example belongs to the SEO class. In the second example, although *where* is not a named entity, it contains the implicit semantic information *Sri Lanka*. BERT-JEORE could use this implicit information to predict the currency of *Sri Lanka*, which is the *Ski Lankan rupee*.

6. Conclusion

We proposed BERT-JEORE, an end-to-end neural network model for extracting all the entities and their overlapping relations in a sentence. We used source-target BERT to generate an entity label for each token in the sentence. Furthermore, we used OREM to generate an unlimited number of relational triplets. Experiments on the NYT and WebNLG datasets indicated that BERT-JEORE outperforms $CopyR_{RL}$ by 15.9% and 26.5%, respectively, and achieves state-of-the-art performance in terms of overlapping relation extraction. We also observed that the pre-trained language model can significantly reduce the amount of manual annotation data required for relation extraction tasks. Our model can be applied to large-scale text corpus relation extraction tasks in other fields as well as to NLP tasks such as event extraction.

BERT-JEORE has considerable potential for further applications. For example, the current model can extract only predefined overlapping relations within sentences. We plan to extend it to the extraction of predefined overlapping relations between multiple sentences as well as open-field overlapping relations.

CRediT authorship contribution statement

Tingting Hang: Methodology, Software, Validation, Formal analysis, Investigation, Writing - original draft. **Jun Feng:** Resources, Writing - review & editing, Supervision, Project administration, Funding acquisition. **Yirui Wu:** Writing - review & editing. **Le Yan:** Writing - review & editing. **Yunfeng Wang:** Software, Validation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Table 11
Case study of BERT-JEORE on the WebNLG dataset. Here, (*Ampara Hospital*, *Sri Lanka*) and *United States* are an overlapping entity pair and an overlapping entity that appear frequently in the WebNLG dataset, respectively.

Sentence S1	The leader of *Aarhus* is *Jacob Bundsgaard*.
BERT-JEORE	(*Aarhus*, /location/location/contains, *Jacob Bundsgaard*)
Ground truth	(Aarhus, /location/location/contains, *Jacob Bundsgaard*)
Sentence S2	*Ampara Hospital* is located in the Eastern Province of *Sri Lanka*, where the currency is the *Ski Lankan rupee*. One of the leaders of *Sri Lanka* is *Ranil Wickremesinghe*.
BERT-JEORE	(*Sri Lanka*, currency, *Ski Lankan rupee*) (*Sri Lanka*, leaderName, *Ranil Wickremesinghe*) (*Ampara Hospital*, country, *Sri Lanka*) (*Ampara Hospital*, state, *Sri Lanka*)
Ground truth	(*Sri Lanka*, currency, *Ski Lankan rupee*) (*Sri Lanka*, leaderName, *Ranil Wickremesinghe*) (*Ampara Hospital*, country, *Sri Lanka*) (*Ampara Hospital*, state, *Sri Lanka*)
Sentence S3	The book *Alcatraz Versus the Evil Librarians* was written in English and comes from the *United States* where the capital city is Washington DC and the *African Americans* are the ethnic group.
BERT-JEORE	(*United States*, ethnicGroup, *African Americans*) (*Alcatraz Versus the Evil Librarians*, country, *United States*)
Ground truth	(*United States*, ethnicGroup, *African Americans*) (*Alcatraz Versus the Evil Librarians*, country, *United States*)

Acknowledgements

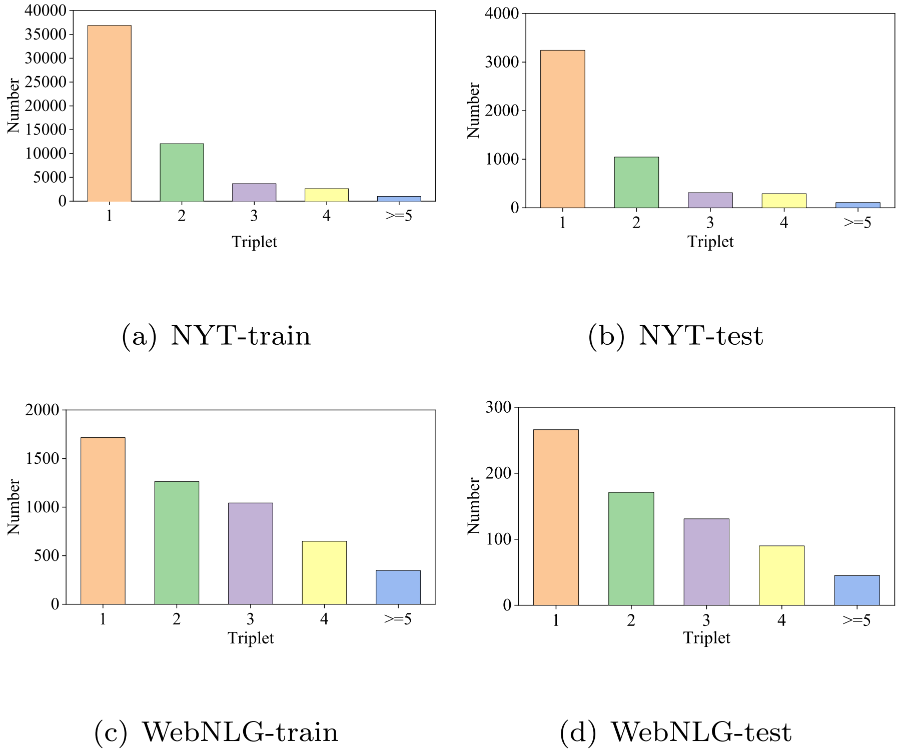
This work was supported by the National Key R&D Program of China (2018YFC0407901), Graduate Research Innovation Support Program of Jiangsu Province, China (2019B64214), and University Natural Science Research Key Projects of Anhui Province, China (KJ2019A1277).

Appendix A. Dataset analysis

In this section, we report the details about the datasets. Specifically, we (i) describe the distribution of the number of triplets and (ii) report the entity pairs and entities that appear most frequently.

Fig. 10 shows the distribution of the number of triplets of the two datasets. According to Fig. 10(a) and (b), more than 90% of the sentences had up to three triplets. Fig. 10(c) and (d) indicate that more than 90% of the sentences had up to four triplets. The number of sentences with five or more triplets is relatively small; hence, it is more difficult to extract such sentences.

Figs. 11 and 12 show the top five overlapping entity pairs with the highest frequency under EPO and the top 10 overlapping entities with the highest frequency under SEO, respectively. In the NYT dataset, (*Iraq*, *Baghdad*) is the most frequently appearing entity pair in the EPO triplets and “Brooklyn” is the most frequently appearing entity in the SEO triplets. In the WebNLG dataset, (*Ampara Hospital*, *Sri Lanka*) is the most frequently appearing entity pair in the EPO triplets, and “United States” is the most frequently appearing entity in the SEO triplets. Statistically, 7.93% (1,090/13,739) of the entity pairs and 16.04% (1,515/9,445) of the entities overlap in the NYT training set, while 3.26% (37/1,136) of the entity pairs and 28.65% (251/876) of the entities overlap in the WebNLG training set.



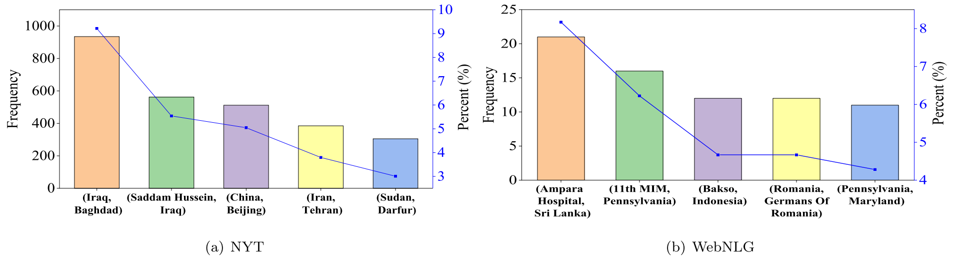
(a) NYT-train

(b) NYT-test

(c) WebNLG-train

(d) WebNLG-test

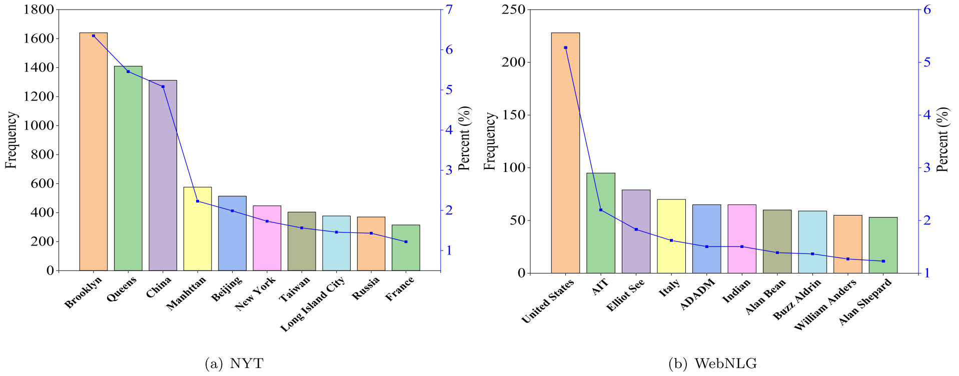
Fig. 10. Distribution of the number of triplets in the two datasets.



(a) NYT

(b) WebNLG

Fig. 11. Top five overlapping entity pairs. The abbreviations 11*th MIM* stand for 11*th Mississippi Infantry Monument*.



(a) NYT

(b) WebNLG

Fig. 12. Top 10 overlapping entities. The abbreviations *AIT* and *ADADM* stand for *Acharya Institute of Technology* and *Accademia di Architettura di Mendrisio*, respectively.

Appendix B. B Model difference analysis

Table 12 summarizes the differences between the models. The first column indicates the model. In the second column, we list the evaluation criteria for each model. Extract match is more in line with real-world usage. As indicated in the third column, only our model uses the entity type.

Table 12
Results of our model and the baseline models. We included different evaluation types (extract match and partial match) to facilitate the comparison of our results with those of previous studies. Extract match refers to the situation in which the relation and the heads and tails of both the subject and the object are correct, and the extracted triplets are considered correct. Partial match refers to the situation in which the relation and the heads of both the subject and the object are correct, and the extracted triplets are considered correct. The symbols √ and × indicate whether or not the models meet this condition.

Model	Evaluation	Entity type	Triplet number	Multi-token entities
NovelTagging	Extract match	×	1	Complete
CopyRE	Partial match	×	≤ 5	Incomplete
GraphRel	Extract match	×	≥ 5	Incomplete
CopyMTL	Extract match	×	≤ 5	Complete
$CopyR_{RL}$	Partial match	×	≤ 5	Incomplete
BERT-JEORE	Extract match	√	≥ 5	Complete

Explicit encoding of the entity type information is critical for relation models, as has been mentioned and confirmed in the work of Peng et al. (2020). As can be seen in the fourth column, many models could extract only a limited number of triplets. An analysis of Fig. 10 demonstrates that there were more than five triplets in the sentences in the two datasets. If the extraction of more than five triplets is ignored, the extraction performance of overlapping relations will be affected. As indicated in the last column, many models could not extract entities with multiple tokens. As multi-token entities are common in real-world scenarios, this could considerably degrade the model performance. The analysis presented above shows that our model bridges the gap due to the other models; hence, its performance advantage in the extraction of overlapping relations is more obvious.

References

Alt, C., Hübner, M., & Hennig, L. (2019). Improving relation extraction by pre-trained language representations. In *Proceedings of the 1st conference on automated knowledge base construction, Massachusetts, USA*.

Bekoulis, G., Deleu, J., Demeester, T., & Develder, C. (2018). Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications, 114*, 34–45.

Cui, Y., Che, W., Liu, T., Qin, B., Wang, S., & Hu, G. (2019). Cross-lingual machine reading comprehension. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing, Hong Kong, China* (pp. 1586–1595).

Dai, D., Xiao, X., Lyu, Y., Dou, S., She, Q., & Wang, H. (2019). Joint extraction of entities and overlapping relations using position-attentive sequence labeling. In *Proceedings of the 33rd AAAI conference on artificial intelligence, Hawaii, USA* (pp. 6300–6308).

Dai, Z., Yang, Z., Yang, Y., Cohen, W. W., Carbonell, J., Le, Q. V., & Salakhutdinov, R. (2019). Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual meeting of the association for computational linguistics (volume 1: Long Papers) Florence, Italy* (pp. 2978–2988).

Devlin, J., Chang, M.-W., Lee, K., & 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), Minnesota, USA* (pp. 4171–4186).

Eberts, M., & Ulges, A. (2019). Span-based joint entity and relation extraction with transformer pre-training. In *Proceedings of the 24th European conference on artificial intelligence, Santiago de Compostela, Spain* (pp. 2006–2013).

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., & Bengio, S. (2010). Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research, 11*, 625–660.

Fei, H., Ren, Y., & Ji, D. (2020a). Boundaries and edges rethinking: An end-to-end neural model for overlapping entity relation extraction. *Information Processing & Management, 57*, Article 102311.

Fei, H., Ren, Y., & Ji, D. (2020b). Dispatched attention with multi-task learning for nested mention recognition. *Information Sciences, 513*, 241–251.

Fu, T.-J., Li, P.-H., & Ma, W.-Y. (2019). Graphrel: Modeling text as relational graphs for joint entity and relation extraction. In *Proceedings of the 57th annual meeting of the association for computational linguistics (volume 1: Long Papers) Florence, Italy* (pp. 1409–1418).

Gardent, C., Shimorina, A., Narayan, S., & Perez-Beltrachini, L. (2017). Creating training corpora for nlg micro-planning. In *Proceedings of the 55th annual meeting of the association for computational linguistics, Vancouver, Canada* (pp. 179–188).

Gulli, A., & Pal, S. (2017). *Deep learning with keras*. Birmingham: Packt Publishing Ltd.

Gupta, P., Schütze, H., & Andrassy, B. (2016). Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of the 26th international conference on computational linguistics, Osaka, Japan* (pp. 2537–2547).

Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., & Weld, D. S. (2011). Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies, Oregon, USA* (pp. 541–550).

Katiyar, A., & Cardie, C. (2017). Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long Papers) Vancouver, Canada* (pp. 917–928).

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the 3rd international conference on learning representations*. California, USA.

Lei, M., Huang, H., Feng, C., Gao, Y., & Su, C. (2019). An input information enhanced model for relation extraction. *Neural Computing and Applications, 31*, 9113–9126.

Li, Q., & Ji, H. (2014). Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (Volume 1: Long Papers) Maryland, USA* (pp. 402–412).

Li, X., Yin, F., Sun, Z., Li, X., Yuan, A., Chai, D., Zhou, M., & Li, J. (2019). Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th annual meeting of the association for computational linguistics, Florence, Italy* (pp. 1340–1350).

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). *Roberta: A robustly optimized bert pretraining approach*. arXiv preprint arXiv:1907.11692.

Liu, Y., Zhou, Y., Wen, S., & Tang, C. (2014). A strategy on selecting performance metrics for classifier evaluation. *International Journal of Mobile Computing and Multimedia Communications, 6*, 20–35.

Mintz, M., Bills, S., Snow, R., & Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of the joint conference of the 47th annual meeting of the ACL and the 4th international joint conference on natural language processing of the AFNL, Singapore* (pp. 1003–1011).

Miwa, M., & 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), Berlin, Germany* (pp. 1105–1116).

Miwa, M., & Sasaki, Y. (2014). Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing, Doha, Qatar* (pp. 1858–1869).

Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes, 30*, 3–26.

Peng, H., Gao, T., Han, X., Lin, Y., Li, P., Liu, Z., Sun, M., & Zhou, J. (2020). Learning from context or names? An empirical study on neural relation extraction. In *Proceedings of the 2020 conference on empirical methods in natural language processing online* (pp. 3661–3672).

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & 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), Louisiana, USA* (pp. 2227–2237).

Radford, A., Wu, J., Amodei, D., Amodei, D., Clark, J., Brundage, M., & Sutskever, I. (2019). *Better language models and their implications*. URL: https://openai.com/blog/better-language-models,.

Ren, X., Wu, Z., He, W., Qu, M., Voss, C. R., Ji, H., Abdelzaher, T. F., & Han, J. (2017). Cotype: Joint extraction of typed entities and relations with knowledge bases. In *Proceedings of the 26th international conference on world wide web, Perth, Australia* (pp. 1015–1024).

Riedel, S., Yao, L., & McCallum, A. (2010). Modeling relations and their mentions without labeled text. In *Proceedings of the joint European conference on machine learning and knowledge discovery in databases, Catalonia, Spain* (pp. 148–163).

Rink, B., & Harabagiu, S. (2010). Utd: Classifying semantic relations by combining lexical and semantic resources. In *Proceedings of the 5th international workshop on semantic evaluation, Uppsala, Sweden* (pp. 256–259).

Singh, S., Riedel, S., Martin, B., Zheng, J., & McCallum, A. (2013). Joint inference of entities, relations, and coreference. In *Proceedings of the 2013 workshop on automated knowledge base construction, California, USA* (pp. 1–6).

Tan, Z., Zhao, X., Wang, W., & Xiao, W. (2019). Jointly extracting multiple triplets with multilayer translation constraints. In *Proceedings of the 33rd AAAI conference on artificial intelligence, Hawaii, USA* (pp. 7080–7087).

Tsai, H., Riesa, J., Johnson, M., Arivazhagan, N., Li, X., & Archer, A. (2019). Small and practical bert models for sequence labeling. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing, Hong Kong, China* (pp. 3630–3634).

Wang, H., Qin, K., Lu, G., Luo, G., & Liu, G. (2020). Direction-sensitive relation extraction using bi-sdp attention model. *Knowledge-Based Systems* (p. 105928).

Wei, Z., Su, J., Wang, Y., Tian, Y., & Chang, Y. (2020). A novel cascade binary tagging framework for relational triple extraction. In *Proceedings of the 58th annual meeting of the association for computational linguistics online* (pp. 1476–1488).

Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K. et al. (2016). *Google's neural machine translation system: Bridging the gap between human and machine translation*. arXiv preprint arXiv: 1609.08144,.

Xue, K., Zhou, Y., Ma, Z., Ruan, T., Zhang, H., & He, P. (2019). Fine-tuning bert for joint entity and relation extraction in chinese medical text. In *Proceedings of the 2019 IEEE international conference on bioinformatics and biomedicine, California, USA* (pp. 892–897).

Yang, S., Feng, D., Qiao, L., Kan, Z., & Li, D. (2019). Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (volume 1: Long Papers) Florence, Italy* (pp. 5284–5294).

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In *Proceedings of the advances in neural information processing systems 32: Annual conference on neural information processing systems, Vancouver, Canada* (pp. 5754–5764).

Zeng, D., Zhang, H., & Liu, Q. (2020). Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning. In *Proceedings of the 34rd AAAI conference on artificial intelligence, New York, USA* (pp. 9507–9514).

Zeng, X., He, S., Zeng, D., Liu, K., Liu, S., & Zhao, J. (2019). Learning the extraction order of multiple relational facts in a sentence with reinforcement learning. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing, Hong Kong, China* (pp. 367–377).

Zeng, X., Zeng, D., He, S., Liu, K., Zhao, J., et al. (2018). Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long Papers), Melbourne, Australia* (pp. 506–514).

Zhao, C., & He, Y. (2019). Auto-em: End-to-end fuzzy entity-matching using pre-trained deep models and transfer learning. In *Proceedings of the 2019 world wide web conference, California, USA* (pp. 2413–2424).

Zheng, S., Hao, Y., Lu, D., Bao, H., Xu, J., Hao, H., & Xu, B. (2017). Joint entity and relation extraction based on a hybrid neural network. *Neurocomputing, 257*, 59–66.

Zheng, S., Wang, F., Bao, H., Hao, Y., Zhou, P., & Xu, B. (2017). Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long Papers), Vancouver, Canada* (pp. 1227–1236).

Zheng, S., Xu, J., Bao, H., Qi, Z., Zhang, J., Hao, H., & Xu, B. (2016). Joint learning of entity semantics and relation pattern for relation extraction. In *Proceedings of the joint European conference on machine learning and knowledge discovery in databases, Riva del Garda, Italy* (Vol. 9851, pp. 443–458).