Expert Systems With Applications 224 (2023) 119905

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Bidirectional relation-guided attention network with semantics and knowledge for relational triple extraction

Yi Yang, Shangbo Zhou *, Yuxuan Liu

College of Computer Science, Chongqing University, Chongqing, China

ARTICLE INFO

Keywords:
Relational triple extraction
Bidirectionality
Semantic and knowledge relations
Multitask learning
Attention mechanism

ABSTRACT

Relational triple extraction is aimed at detecting entity pairs with relations from sentences, which is a key technology for large-scale knowledge graph construction. Recent studies focus on the overlapping triple problem, where multiple relational triples may have overlaps in a sentence. However, these methods disregard the bidirectionality of triple extraction, which may lead to extracting invalid triples. In addition, many relational triples are labeled in datasets of the triple extraction task, implying domain knowledge information of these datasets, but current methods rarely consider it. In this paper, we present a bidirectional relation-guided attention network with semantics and knowledge (BRASK) for relational triple extraction. BRASK is a bidirectional extraction framework that is based on multitask learning and contains forward and backward triple extraction tasks. Forward triple extraction and backward triple extraction are parallel and complementary, which can obtain predicted results with high confidence. We utilize semantic relations and knowledge relations as guidance in forward triple extraction and backward triple extraction, respectively, thus integrating general and domain knowledge into our model. In addition, we adopt an attention mechanism to learn fine-grained sentence representations for different relations. BRASK can solve the triple overlap problem and capture bidirectional dependencies between subjects and objects. Experimental results show that BRASK achieves new state-of-the-art results in two public datasets, which demonstrates its effectiveness.

1. Introduction

Relational triples are crucial components of knowledge graphs, which are commonly observed in natural language phrases. Relational triple extraction is aimed at learning relational facts (subject, relation, and object) from unstructured texts. Previously, extracting relation triples adopted pipelined models, which identify entities and then predict their relations. These traditional approaches (Chan & Roth, 2011; Nadeau & Sekine, 2007; Zelenko, Aone, & Richardella, 2003; Zhou, Su, Zhang, & Zhang, 2005) disregard the correlation between the two processes, resulting in error propagation. Thus, the joint extraction of entities and relations is subsequently applied. Prior joint learning frameworks (Kate & Mooney, 2010; Li & Ji, 2014; Miwa & Sasaki, 2014; Yu & Lam, 2010) heavily rely on complicated feature engineering and natural language processing tools. The latter methods (Katiyar & Cardie, 2017; Miwa & Bansal, 2016) are based on neural networks, which utilize a parameter sharing strategy for jointly learning entities and relations. Although these methods achieve better performance, entity pairs and their relations are still separately identified. Zheng et al. (2017) presented a tagging framework that combines the triple extraction problem as a sequence labeling task. This scheme cannot address the overlapping problem because each word of the sentence is only assigned one label. Zeng, Zeng, He, Liu, and Zhao (2018) considers the overlapping triple problem where the same entities may appear in multiple relational facts. As illustrated in Fig. 1, there are four overlapping scenarios: Normal, EPO, SEO and RO classes. Relation Overlap (RO) indicates that multiple triples may contain the same relations. To solve this problem, Zeng et al. (2018) adopts a sequence-to-sequence learning method based on a copy mechanism to identify overlapping triples. Yu, Zhang, and Su (2019) presents an end-to-end framework based on a decomposition approach, which identifies all possible subjects and then distinguishes relevant objects and relations. Instead of considering the relations as classification labels, Wei, Su, Wang, Tian, and Chang (2020) provides a fresh perspective to detect relational triples. Their model can naturally address the triple overlapping problem but disregards the relation guidance.

Based on relation guidance, we divide current research into nonrelation-guided approaches and relation-guided extraction approaches. Wei et al. (2020), Yu et al. (2019) are nonrelation-guided approaches for learning overlapping triples. Yang, Li, and Li (2021), Yuan et al. (2020), Zeng et al. (2018) are relation-guided learning

* Corresponding author.
E-mail addresses: yang.yi@cqu.edu.cn (Y. Yang), shbzhou@cqu.edu.cn (S. Zhou), liuyuxuan@cqu.edu.cn (Y. Liu).

https://doi.org/10.1016/j.eswa.2023.119905
Received 24 March 2022; Received in revised form 12 March 2023; Accepted 16 March 2023
Available online 20 March 2023
0957-4174/

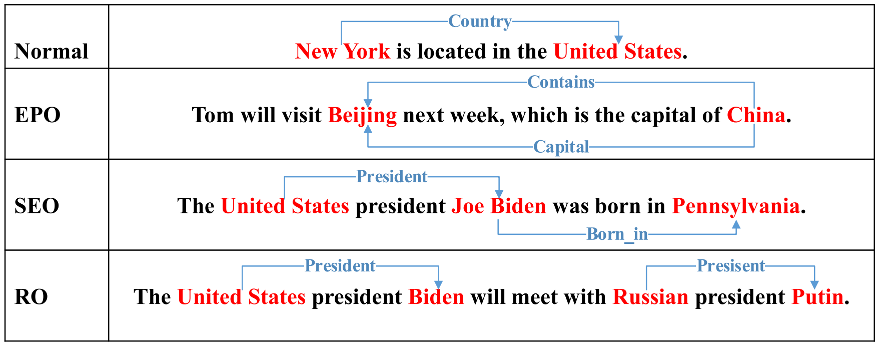

Fig. 1. Examples of the Normal, Entity Pair Overlap (EPO), Single Entity Overlap (SEO) and Relation Overlap (RO) cases.

models to identify relational facts. Zeng et al. (2018) presents an end-to-end framework with a copy mechanism, which conducts relation classification and then uses the extracted relations to guide entity extraction. However, the authors disregard the fine-grained correlation between the relation and tokens in a sentence. Yuan et al. (2020) considers relation guidance, which learns specific relation representations with an attention mechanism to guide the extraction of entity pairs. Their model cannot learn dependencies between entity pairs. Yang et al. (2021) introduces implicit relational triples whose relations are not obvious in a sentence. The authors consider each relation as prior knowledge instead of discrete tags and capture fine-grained semantic expressions of relations to guide all subjects to identify related objects, which is suitable for detecting implicit relational triples.

Although these extraction methods achieve great success, they disregard the bidirectionality of relational triple extraction. Bidirectionality indicates that using the bidirectional extraction method identifies relational triples, which contain forward and backward triple extraction. We define forward triple extraction as using extracted subjects to identify objects and backward triple extraction as using objects to identify subjects. Both (Wei et al., 2020; Yang et al., 2021) are forward triple extraction methods to extract relational triples, which belong to the unidirectional models. However, the predicted results of these unidirectional extraction methods may not be credible enough. For example, as shown in Fig. 2, we try extracting target relational triples from the sentence '*Biden and Blinken will visit China on behalf of the United States*'. However, forward triple extraction has extracted an invalid relational triple *(Blinken, president_of, United States)*. Due to different extraction directions, this invalid triple has not been extracted by backward triple extraction. From this example, we observe that target triples will be correctly extracted if they can simultaneously be obtained by forward triple extraction and backward triple extraction. As a result, bidirectional extraction may filter some invalid triples and obtain more accurate predicted results, revealing the necessity of bidirectionality.

In addition, previous triple extraction methods usually adopted semantic learning methods but rarely considered evaluating knowledge. Here, the semantics indicate the general language information, while the knowledge is private knowledge information of the specific field. Normally, knowledge graphs are composed of relational triples, which represent domain knowledge. Because of the peculiarity of the triple extraction task, there are many relational triples in each dataset, which contain specific knowledge of these datasets. To learn the knowledge, we utilize the knowledge embedding algorithm TransE (Bordes, Usunier, Garcia-Duran, Weston, & Yakhnenko, 2013). Therefore, we simultaneously integrate semantic information and knowledge information into bidirectional extraction, which represent common knowledge and domain knowledge, respectively.

In this work, we present a relational triple extraction model based on the bidirectional relation-guided attention network with semantics and knowledge. We adopt a bidirectional extraction framework to distinguish relational triples from sentences, which is based on multitask learning. Our model is constrained by two tasks: forward triple extraction and backward triple extraction. For forward triple extraction, we identify subjects that may be relevant to target triples and then identify the corresponding objects. For backward triple extraction, we extract all possible objects and then identify the subjects under each relation. The relational triples are extracted if there are such objects and subjects in forward triple extraction and backward triple extraction, respectively. To integrate the semantics and knowledge into our model, we utilize semantic and knowledge relations as guidance in bidirectional extraction. Semantic relations are acquired with the pretrained language model BERT (Devlin, Chang, Lee, & Toutanova, 2018). The knowledge relations are acquired by the knowledge embedding algorithm TransE, which learns the related information about entities. The semantic and knowledge relations correspond to the general and domain knowledge, which are applied in forward triple extraction and backward triple extraction, respectively. Moreover, considering that each token may play a different role in the identification of relevant objects or subjects, we utilize specific attention to learn the fine-grained relation features.

The major contributions of our paper are presented as follows:

(1) We introduce the bidirectionality of relational triple extraction. To learn the bidirectionality, we present a bidirectional relation-guided attention network with semantics and knowledge (BRASK) to extract relational triples. Bidirectional extraction is based on multitask learning, which can learn bidirectional dependencies between subjects and objects.

(2) Semantic and knowledge information is simultaneously utilized in bidirectional extraction, which represents general knowledge and domain knowledge, respectively, to guide the identification of corresponding entities. Our model adopts an attention mechanism to acquire different sentence representations of relations.

(3) Our BRASK can solve the overlapping triple problem and filter invalid predicted triples to some extent. We test our method on two public datasets, NYT and WebNLG. Experimental results show that BRASK outperforms previous works, which verifies its effectiveness.

2. Related work

Extracting relational facts from unstructured sentences is a key technology for constructing knowledge graphs. Traditional pipelined methods (Chan & Roth, 2011; Gormley, Yu, & Dredze, 2015; Mintz, Bills, Snow, & Jurafsky, 2009; Nadeau & Sekine, 2007; Zelenko et al., 2003; Zhou et al., 2005) recognize all entities from a sentence and then conduct relation classification. They disregard internal connections between the two steps and may suffer from error propagation. To address these limitations, many researchers apply joint models to detect entities and relations. Early joint methods (Kate & Mooney, 2010; Li & Ji, 2014; Miwa & Sasaki, 2014; Yu & Lam, 2010) were based on complicated feature engineering and natural language processing tools.

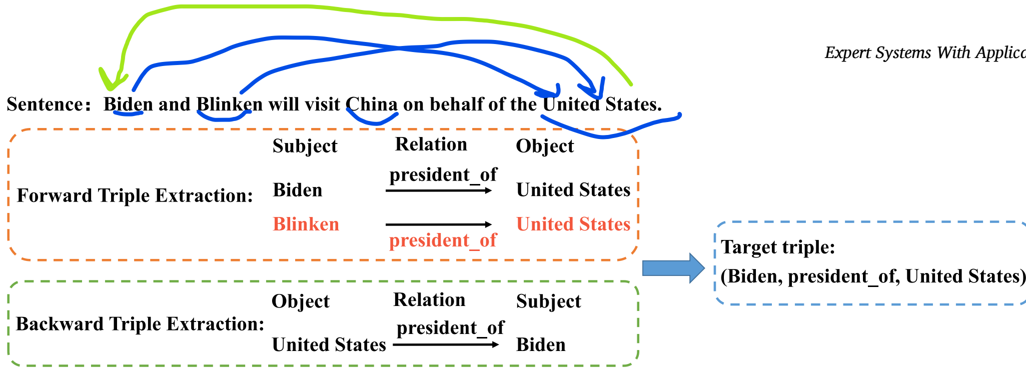

Fig. 2. An example of bidirectional extraction for identifying relational triples.

To reduce manual work, later studies have investigated neural network-based approaches (Katiyar & Cardie, 2017; Miwa & Bansal, 2016), which learn entities and relations by shared parameters. However, these approaches still individually identify entity pairs and relations.

Zheng et al. (2017) considers the joint extraction problem as a labeling task. The authors examine different end-to-end models to extract relational triples but cannot address the triple overlap problem. Zeng et al. (2018) proposes the overlapping problem in triple extraction and then introduces different overlapping types. To address this problem, the authors adopted an end-to-end method based on a copy mechanism to extract entities and relations from sentences with different overlapping patterns. Nayak and Ng (2020) proposed two methods using encoder–decoder architecture to extract relational facts. The authors present a representation framework for entities and relations that can produce one word at a time. Next, they adopt pointer networks in the decoder, which can identify a relational triple at every time step. Yu et al. (2019) presents an end-to-end model based on a decomposition strategy, which extracts possible subjects and then distinguishes the relevant relations and objects. Previous studies consider relations as discrete labels, but Wei et al. (2020) utilizes relations as functions, which identifies all possible subjects and then distinguishes the related objects. However, the authors disregard the relation guidance for subjects to identify objects. Yuan et al. (2020) learns the specific expressions with relation-based attention for guiding the extraction of entities. Their model cannot learn the connections between entity pairs. Yang et al. (2021) proposes a relation-guided model to extract relational facts. The authors consider relations as prior knowledge and acquire special relation representations as guidance in the identification of relevant objects. However, their model is unidirectional, and knowledge is not applied for the identification of triples.

3. Methodology

Our BRASK is an end-to-end triple extraction model with a bidirectional relation-guided attention network. For extracting entities, we adopt the binary tagging strategy, which can transform the extraction problem into sequence tagging tasks. First, we encode texts by the language model BERT and then adopt the bidirectional extraction framework to obtain target relational triples. Bidirectional extraction contains forward and backward triple extraction, which are based on multitask learning. For forward triple extraction, we detect subjects and then identify the corresponding objects with semantic relations. However, it is not enough to obtain final relational triples only by forward triple extraction, and there may be some invalid triples in the predicted results. We still need backward triple extraction to constrain our model, which plays a complementary role. For backward triple extraction, we extract all possible objects and then use the extracted objects to identify the relevant subjects with knowledge relations. Forward and backward triple extraction are parallel, which jointly determine the final relational triples. Our model framework is shown in Fig. 3.

3.1. Entity tagging scheme

To extract both subjects and objects from a sentence, we adopt a binary tagging strategy that considers the entity extraction problem as sequence tagging tasks. Each word in the sentence is tokenized to tokens with WordPiece embeddings (Wu et al., 2016), i.e., subwords. Therefore, an entity contains one or more tokens. For each entity, we define e_s and e_e as the start token of an entity and end token of an entity, respectively. Therefore, we represent an entity in the form of (e_s, e_e). To distinguish entities from the sentence, the two labels e_s and e_e are assigned to each token of the sentence, which represents the start position of an entity and end position of an entity, respectively. In addition, we assign a binary label 1 or 0 to both e_s and e_e, which represents whether the token corresponds to the first or last token of an entity. An entity is extracted if the start and end tokens of an entity are distinguished. To extract multiple entities from a sentence, we apply the principle that each e_s that is labeled 1 and its nearest e_e that is labeled 1 jointly dete[redacted]
scheme.

3.2. BERT encoder

We use the langu[redacted]tations. BERT applies large-scale corpora for pretraining, which can capture semantic features to be fine-tuned on downstream tasks. BERT adopts multilayer transformers (Vaswani et al., 2017) to pretrain deep bidirectional representations. The sentence representations are acquired as follows:

$$h_i = Trm(h_{i-1}), i \in [1, n] \tag{1}$$

The inputs of our BERT encoder are token embedding and positional embedding. Here, we do not need segmentation embedding compared with the original BERT because the text of the datasets is a single sentence. For token embedding, BERT utilizes WordPiece embedding (Wu et al., 2016), which tokenizes the sentence using a 30000 token vocabulary. h_i is the ith hidden state vector. $Trm()$ is the Transformer block and n represents the total number of Transformer blocks.

3.3. Forward triple extraction

We extract all possible subjects of triples and then identify corresponding objects under all relations, which is referred to as forward triple extraction. The direction of forward triple extraction is from subjects to objects. We utilize semantic relations to guide extracted subjects for identify objects. Semantic relations are acquired by the pretrained BERT model. We obtain a relational triple set with forward triple extraction.

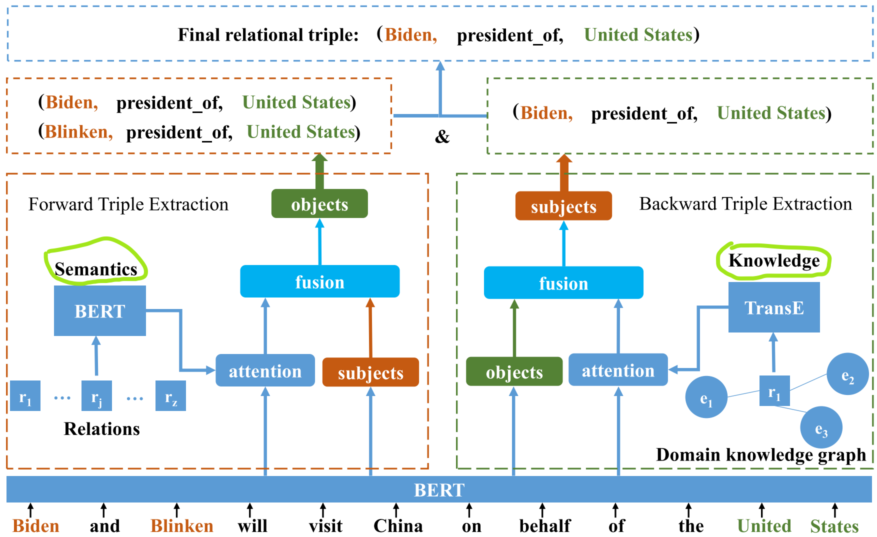

Fig. 3. Overview of our BRASK for relational triples extraction. First, we feed the sentence into BERT to obtain encoded representations. Second, our model adopts bidirectional extraction to obtain target triples, which contains forward and backward triple extraction. We use the intersection of the predicted triple sets obtained by forward and backward triple extraction to obtain the final relational triples.

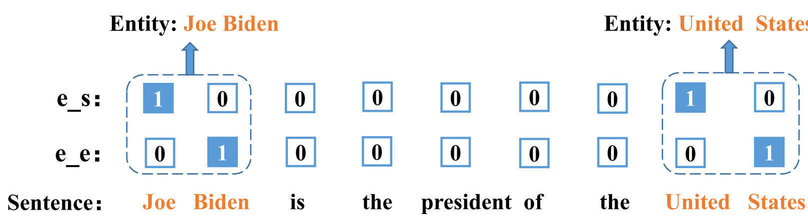

Fig. 4. An example of an entity tagging scheme. e_s and e_e indicate the start tag of an entity and end tag of an entity, respectively.

3.3.1. Extraction of subjects

Our model extracts subjects via the binary tagging strategy introduced above. Based on this strategy, we feed sentence representations encoded by BERT into the fully connected neural network to obtain probabilities of distinguishing each token in the input sequence as a subject's start or end token. The detailed operations are presented as follows:

$$p_{i,sub\_s} = Sigmoid(W_{sub\_s}x_i + b_{sub\_s}) \tag{2}$$

$$p_{i,sub\_e} = Sigmoid(W_{sub\_e}x_i + b_{sub\_e}) \tag{3}$$

where x_i indicates the encoded representation of the ith token, which is the input of the fully connected neural network. W_{sub_s}, W_{sub_e}, b_{sub_s} and b_{sub_e} are the parameters, and the activation function is sigmoid.

A subject set S is obtained by subject extraction, and we use s_k to represent the kth extracted subject. We average the encoded representations of the kth subject's start and end tokens to represent s_k:

$$s_k = avg(x_{k,sub\_s},\ x_{k,sub\_e}) \tag{4}$$

where x_{k,sub_s} and x_{k,sub_e} indicate encoded representations of the kth subject's start token and end token, respectively, in the sentence.

3.3.2. Semantic relation guidance

After identifying all subjects, we extract the corresponding objects under each relation. Based on our method, identifying the relevant objects is a key and difficult step in extracting entire triples. Therefore, to better extract the objects, our model utilizes relations to guide each extracted subject for identifying objects. Relations are the bridges between subjects and objects. The relation set R is predefined, and we use r_j to represent the jth relation. To obtain relation embeddings, we utilize pretrained BERT to obtain its last two hidden state vectors of all tokens and then use average pooling. BERT is aimed at pretraining deep bidirectional representations from large-scale corpora, which can better learn semantic information and understand language. Therefore, we define relation embeddings by pretrained BERT as semantic relations. For each subject, our model can identify corresponding objects or no objects with semantic relation guidance. Triples are extracted if there are relevant objects. However, it is not applicable if we directly combine extracted subjects and semantic relations with sentence representations. Because the words of a sentence may play different roles in the identification of objects, our model adopts a relation-specific attention mechanism to compute attention scores that reflect a different role of each token, which can acquire fine-grained relation representations.

$$h_g = avg\{x_1, x_2, \ldots, x_l\} \tag{5}$$

$$e_{i,j} = V^T tanh(W_r r_j + W_g h_g + W_x x_i) \tag{6}$$

$$a_{i,j} = softmax(e_{i,j}) \tag{7}$$

$$c_j = \sum_{i=1}^{l} a_{i,j} x_i \tag{8}$$

where h_g represents the global sentence with average BERT encoded representations of all tokens in a sentence and l represents the length of the sentence. Eq. (6) uses the multilayer perceptron to compute attention scores that can measure a different role played by each token. r_j indicates the jth semantic relation embedding, and $a_{i,j}$ is acquired by the softmax to represent the weight coefficient. c_j indicates the fine-grained sentence representation, which is acquired by the weighted average between encoded representations of all tokens and $a_{i,j}$.

3.3.3. Extraction of objects

We also apply the binary tagging strategy to distinguish the corresponding objects. First, we fuse the subject representations and fine-grained sentence expressions of the semantic relations into the ith token's encoded representation.

$$H_{i,k} = W_s s_k + W_x x_i \tag{9}$$

$$H_{i,j} = c_j + x_i \tag{10}$$

$$H_{i,j,k} = H_{i,k} + H_{i,j} \tag{11}$$

where $H_{i,k}$ is the ith token's encoded representation x_i integrating the information of the subject. $H_{i,j}$ is the fusion of the fine-grained semantic relation representation, and x_i. Eq. (11) fuses $H_{i,k}$ and $H_{i,j}$ to obtain the specific expression integrating subject and relation representations. Second, we feed the special representation into the fully connected neural network to obtain the start and end probabilities of the corresponding object:

$$p_{i,obj\_s} = Sigmoid(W_{obj\_s} H_{i,j,k} + b_{obj\_s}) \tag{12}$$

$$p_{i,obj\_e} = Sigmoid(W_{obj\_e} H_{i,j,k} + b_{obj\_e}) \tag{13}$$

where W_{obj_s}, b_{obj_s} and W_{obj_e} and b_{obj_e} are parameters of the fully connected neural network.

3.4. Backward triple extraction

We obtain a relational triple set by forward triple extraction with semantic relation guidance. However, it is not enough to obtain the final results of triple extraction. As a result, we apply backward triple extraction to filter invalid relational triples extracted by forward triple extraction. Backward triple extraction indicates that we extract all possible objects at the beginning and then identify the corresponding subjects. The direction of backward triple extraction is from objects to subjects. In contrast, we use knowledge relation guidance in backward triple extraction. By backward triple extraction, we obtain more accurate relational triples.

In backward triple extraction, the processes of object and subject extraction are the same as those of forward triple extraction. Thus, we will not repeat it in this part.

3.4.1. Knowledge relation guidance

To identify the corresponding subjects according to extracted objects, we also utilize relations as guidance. Unlike forward triple extraction, our model adopts knowledge relation guidance in the process of backward triple extraction. Knowledge relations are acquired by the TransE, which is an approach to learn embeddings of knowledge graphs, focusing on the minimal parametrization of the model to primarily represent hierarchical relations. Because of the peculiarity of the triple extraction task, there are many relational triples in our datasets, which contain specific knowledge of these datasets. We utilize all relational triples of the training set to construct a domain knowledge graph and to apply TransE in this knowledge graph to obtain all relation embeddings. Our model can learn domain knowledge information from the knowledge graph by TransE, which plays a role in knowledge reasoning. We still adopt a relation-specific attention mechanism to acquire fine-grained sentence expressions for knowledge relations.

$$e'_{i,j} = V^T tanh(W'_r r'_j + W'_g h_g + W'_x x_i) \tag{14}$$

$$a'_{i,j} = softmax(e'_{i,j}) \tag{15}$$

$$c'_j = \sum_{i=1}^{l} a'_{i,j} x_i \tag{16}$$

where r'_j represents the relation embedding trained by TransE. We also obtain a relational triple set with backward triple extraction.

3.5. Bidirectional extraction

Our bidirectional extraction framework of relational triples is based on multitask learning, containing the forward and backward triple extraction introduced above. Forward and backward triple extraction are parallel and complementary, which can learn bidirectional dependencies between subjects and objects and capture more implicit features. Our model naturally combines semantic and knowledge relation guidance into bidirectional extraction. The relational triples are correctly extracted if they can be simultaneously extracted by forward and backward triple extraction. Therefore, we use the intersection of two predicted triple sets obtained by forward and backward triple extraction, which can filter triples that do not satisfy both the semantics and knowledge in a sentence. We acquire more accurate predicted triples by bidirectional extraction.

$$T_{final} = T \& T' \tag{17}$$

where T_{final} indicates the final relational triple set by bidirectional extraction. T and T' are relational triple sets obtained by forward triple extraction and backward triple extraction, respectively. & indicates the intersection operation.

3.6. Loss function

Our model simultaneously adopts forward and backward triple extraction. Thus, the total loss is composed of two parts: $\mathcal{L}$ for forward triple extraction and $\mathcal{L}'$ for backward triple extraction.

$$\mathcal{L}_{total} = \sum_{d=1}^{|D|} (\mathcal{L} + \mathcal{L}') \tag{18}$$

where $\mathcal{L}_{total}$ represents the total loss, $|D|$ is the size of the training set, and d indicates the dth sample. We minimize $\mathcal{L}_{total}$ with Adam stochastic gradient descent (Kingma & Ba, 2014) over shuffled mini-batches.

In forward triple extraction, the loss of subject extraction $\mathcal{L}_{sub}$ and the loss of object extraction $\mathcal{L}_{obj}$ under specific relations are provided to train the model. The loss of forward triple extraction is defined as follows:

$$\mathcal{L} = \mathcal{L}_{sub} + \mathcal{L}_{obj} \tag{19}$$

$$\mathcal{L}_{sub} = \sum_{s \in T_d} \log p(s|X_d) \tag{20}$$

where T_d indicates extracted relational triples of the dth sentence X_d and s represents the extracted subject. The likelihood function $p(s|X_d)$ is:

$$p(s|X_d) = \prod_{t \in \{sub\_s, sub\_e\}} \prod_{i}^{l} (p_i^t)^{I(y_i^t=1)} (1 - p_i^t)^{I(y_i^t=0)} \tag{21}$$

where y_i^t indicates the binary label of the subject's start or end token for the ith token. $I(y)$=1 if y is true and 0 otherwise. $\mathcal{L}_{obj}$ is defined in the same way.

For backward triple extraction, the loss $\mathcal{L}'$ contains the losses of object and subject triple extraction: $\mathcal{L}'_{obj}$, $\mathcal{L}'_{sub}$, which is defined as the same as $\mathcal{L}$.

4. Experiments

4.1. Datasets

We evaluate our proposed method on two public datasets, NYT (Riedel, Yao, & McCallum, 2010) and WebNLG (Gardent, Shimorina, Narayan, & Perez-Beltrachini, 2017). NYT is constructed by distant supervision, which aligns the knowledge base and raw data to produce training data. There are 24 pretrained relations in total. The

Table 1
Statistics of the datasets.

Relations	NYT		WebNLG	
	Train	Test	Train	Test
	24		216	
Normal	37013	3266	1596	246
SEO	14735	1297	3406	457
EPO	9782	978	227	26
RO	4846	456	455	53
ALL	56195	5000	5019	703

NYT dataset provides 56195 instances for training, 5000 instances for validation, and 5000 instances for testing. WebNLG was proposed by Gardent et al. (2017) for natural language generation tasks. Zeng et al. (2018) preprocesses this dataset for the extraction of relational triples, and the dataset contains 216 relation types. In WebNLG, 5019 instances are selected as the training set, 500 instances are selected for validation, and the remaining 793 instances are employed as the test set. According to overlapping patterns, we divide the sentences of these two datasets into four types: Normal, SEO, EPO and RO. A sentence holds Relation Overlap (RO) if some of its triples have overlapped relations. An example of the RO case is shown in Fig. 1. We observe that there are two different triples in a sentence that share the same relation 'President'. Table 1 illustrates the statistics of the NYT and WebNLG datasets.

4.2. Baselines and evaluation metrics

We adopt the following baselines to compare our model:

- NovelTagging (Zheng et al., 2017) presents a novel tagging framework that combines both the entity roles and relation role, converting the triple extraction problem to a tagging task. However, their model cannot address the triple overlapping problem.
- CopyRE (Zeng et al., 2018) applies a sequence-to-sequence method based on a copy mechanism to directly produce relational triples. This copy mechanism can only copy the last token of an entity, which is not suitable for solving the triple overlap problem.
- Graphrel (Fu, Li, & Ma, 2019) presents an end-to-end approach for extracting relations and adopts graph convolutional networks (GCNs) (Kipf & Welling, 2016) to identify relational triples, which can extract the inherent features among all word pairs of sentences.
- CopyMTL (Zeng, Zhang, & Liu, 2020) introduces a multitask learning method to enhance the capability of addressing multi-token entities.
- Rsan (Yuan et al., 2020) learns the fine-grained sentence expressions of relations to guide entity recognition and controls the relation information for decoding entities with a relational gated mechanism.
- CasRel (Wei et al., 2020) is a novel binary labeling framework derived from a principled problem formulation to jointly detect entities and relations, and their model naturally solves the overlap problem.
- RGAM (Yang et al., 2021) extracts relational triples with a relational-guided attention mechanism and identifies implicit relational triples from sentences.
- TpLinker (Wang et al., 2020) presents a one-stage method for relational triple extraction, which narrows the gap between learning and inference.

We follow the evaluation metrics of prior works and adopt the standard precision (Prec), recall (Rec) and F1 scores to evaluate the results. A triple is regarded as correct if the entity pair and its relation are exactly matched.

4.3. Implementation details

In our experiments, the maximum length of the sentence is set to 100. The encoder BERT contains 100 M parameters, which is the BERT-Base-Cased version. The embedding dimension of the semantic relation is 768. The dimension of knowledge relation embedding trained by TransE is 768. We train our model with a learning rate of 1e-5 and a batch size of 4. The training will end if the results on the validation set do not gain any improvement within 9 epochs, preventing the training from overfitting. The tag threshold is set to 0.5, which is used to identify the start and end tokens of entities. For TransE, the margin is 1, and the batch size is set to 300.

4.4. Main results

Table 2 presents the main results of our approach against other baseline methods. We observe that our model, BRASK, outperforms all other models, which achieves state-of-the-art performance. On the NYT and WebNLG datasets, the precision, recall and F1 score of our method have improved compared with the baselines. In particular, our model has significantly improved in precision, which reflects the effectiveness of our proposed method. We attribute the performance gains of BRASK to its three advantages:

(1) Our model adopts multitask learning, which contains tasks of forward and backward triple extraction (bidirectional extraction). Bidirectional extraction has the ability to capture bidirectional features between subjects and objects. Our BRASK is constrained by these two tasks, which can extract more accurate predicted triples. In addition, our method can effectively address the triple overlap problem.

(2) The semantic and knowledge relations are integrated into bidirectional extraction, which play a role in guidance. The semantic relations are learned by pretrained BERT, which contains knowledge of common fields. The knowledge relations are produced with TransE, which is applied in the knowledge graph constructed with all relational triples of the training set, which can learn the domain knowledge and related information about entities. Semantics and knowledge enhance the performance of our model.

(3) We adopt relation-based attention to obtain fine-grained semantic representations, which can measure different roles played by different words in a sentence and focus more on relevant words in the extraction of objects.

4.5. Results of bidirectional extraction under different relation guidance

Our model is a bidirectional extraction framework that contains forward triple extraction with semantic relation guidance and backward triple extraction with knowledge relation guidance. We conduct extended experiments on the NYT dataset to observe the results on bidirectional extraction under different relation guidance. Forward and backward triple extraction can utilize relation or knowledge relation guidance. For example, in Table 3, '$Forward + Semantics, Backward + Knowledge$' indicates that forward triple extraction uses semantic relation guidance, while backward triple extraction adopts knowledge relation guidance. Table 3 presents the results of these experiments. Bidirectional extraction achieves outstanding performance under any relation guidance. Forward extraction with semantic guidance and backward extraction with knowledge guidance achieve the best results. We suggest that this results is attributed to the difference in the difficulty of forward and backward extraction in extracting relational triples and the relatively greater difficulty of backward extraction. In addition, knowledge relations are trained by TransE, which has learned more correlative information about entities than semantic guidance. Knowledge relation guidance is more conducive to identifying the corresponding entities. If the relatively simpler forward extraction uses knowledge guidance, our model may lead to overfitting. It is more suitable for forward extraction and backward extraction to use semantic relation guidance and knowledge relation guidance, respectively.

Table 2
Main results of our model and the compared methods on NYT and WebNLG.

Model	NYT			WebNLG		
	Pre	Rec	F1	Pre	Rec	F1
NovelTagging (Zheng et al., 2017)	62.4	37.1	42	52.5	19.3	28.3
CopyRE (Zeng et al., 2018)	61	56.6	58.7	37.7	36.4	37.1
GraphRel (Fu et al., 2019)	63.9	60	61.9	44.7	41.1	42.9
CopyMTL (Zeng et al., 2020)	75.7	68.7	72	58	54.9	56.4
Wdec (Nayak & Ng, 2020)	88.1	76.1	81.7	84.8	64.9	73.5
Rsan (Yuan et al., 2020)	85.7	83.6	84.6	80.5	83.8	82.1
CasRel (Wei et al., 2020)	89.7	89.5	89.6	93.4	90.1	91.8
RGAM (Yang et al., 2021)	90.6	92	91.3	93.5	91.9	92.6
TPLinker (Wang et al., 2020)	91.3	**92.5**	91.9	91.8	92	91.9
BRASK	**93**	91.5	**92.2**	**94.8**	**92.2**	**93.5**

Table 3
Results of bidirectional extraction under different relation guidance.

Model	Pre	Rec	F1
Forward+Semantics, Backward+Knowledge	93	91.5	92.2
Forward+Knowledge, Backward+Semantics	92.2	91.2	91.7
Forward+Semantics, Backward+Semantics	92.4	91.6	92
Forward+Knowledge, Backward+Knowledge	92.6	91	91.8

4.6. Ablation study

We conduct ablation experiments on the NYT and WebNLG datasets to explore the effectiveness of bidirectional extraction, semantic relation guidance, knowledge relation guidance and relation-guided attention. To verify the validity of each component, we remove one particular part at a time to observe the influence on the results. As shown in Table 4, '$-Bidirectional\ extraction$' refers to removing the backward triple extraction and only adopts the forward triple extraction method to obtain relational triples. '$-Semantic\ relation\ guidance$' indicates that our model does not use the semantic relations to guide extracted subjects for identifying the objects in forward triple extraction. '$-Knowledge\ relation\ guidance$' indicates that backward triple extraction does not utilize knowledge relations as guidance and that the subjects can map to the corresponding objects with no relation guidance. '$-Relation-guided\ attention$' indicates that our model does not adopt relation-guided attention for learning fine-grained semantic expressions.

From these results, we observe that the evaluation metrics of these experiments have declined. Semantic and knowledge relation guidance have the greatest impact on the results in the NYT dataset. However, for the WebNLG dataset, knowledge relation guidance has a greater impact than semantic relation guidance on the results since the WebLNG dataset contains more relations than the NYT dataset. The model may predict more invalid relational triples without bidirectional extraction. Bidirectional extraction is composed of both forward triple extraction and backward triple extraction, which are complementary and jointly determine the final relational triples. It is not credible enough if we only use forward triple extraction to extract target relational triples. Bidirectional extraction can filter triples that do not satisfy both the semantics and knowledge in a sentence. Semantic relation guidance can guide extracted subjects to distinguish objects under different relations. Semantic relations are obtained by pretrained BERT, which can learn general semantic representations. Knowledge relations are trained by TransE on specific knowledge graphs, which contain domain knowledge and are suitable for playing a role in guidance. Moreover, using a relation-guided attention mechanism can focus more on effective tokens under specific relations, which leads to a more significant distinction between positive relations and negative relations. These experiments demonstrate the advantages of our proposed methods in relational triple extraction.

Table 4
Ablation study of BRASK on NYT and WebNLG datasets.

Model	NYT			WebNLG		
	Pre	Rec	F1	Pre	Rec	F1
BRASK	93	91.5	92.2	94.8	92.2	93.5
–Bidirectional extraction	91.3	91.5	91.4	92.9	92.1	92.5
–Semantic relation guidance	92.3	89.9	91.1	93.9	91.7	92.8
–Knowledge relation guidance	92.8	89.7	91.2	93.1	91.3	92.2
–Relation-guided attention	92.2	91.2	91.7	93.8	92.1	93

Table 5
F1 score of our method and other baselines on sentences with different patterns and different number of triples.

Model	NYT								
	Normal	EPO	SEO	RO	N=1	N=2	N=3	N=4	N≥ 5
GraphRel	69.6	58.2	51.2	–	71	61.5	57.4	55.1	41.1
CasRel	87.3	92	91.4	80.6	88.2	90.3	91.9	94.2	83.7
RGAM	89.4	93.2	92.8	87	89.4	92.7	93.4	94.4	88.6
TPLinker	90.1	94	93.4	–	90	92.8	93.1	**96.1**	90
BRASK	**90.5**	**94**	**93.7**	**90.1**	**90.4**	**93**	**93.7**	95.3	**91.1**

4.7. Detailed results on different types of sentences

Based on previous works, we conduct further experiments to investigate the capability of BRASK in extracting overlapping and multiple triples on the NYT dataset. We divide the sentences of the test set into three types based on different overlapping patterns, i.e., normal, EPO, SEO and RO. The first experiment validates the effects of our method and baselines in extracting entities and relations under all overlapping patterns. As shown in Table 5, BRASK outperforms all other models in overlapping patterns, especially for the RO class, reflecting the importance of relation guidance. In addition, we explore BRASK's ability to detect different numbers of relational triples from sentences. We divide the instances of the NYT dataset into 5 types, which indicate 1, 2, 3, 4, and more than 5 triples in the sentence. From Table 5, our model achieves much higher performance in learning different numbers of relational facts from sentences. These experimental results demonstrate that BRASK is effective in solving overlapping and multiple triple problems.

4.8. Analysis of tag threshold

In this subsection, we analyze the effects of our approach under different tag thresholds on the NYT dataset. The tag threshold is set for identifying both subjects and objects from the sentence. Fig. 5 presents the results of our method under different tag thresholds. With an increase in the threshold, the precision of our model gradually rises, while the recall gradually falls. The F1 score increases and then decreases, but

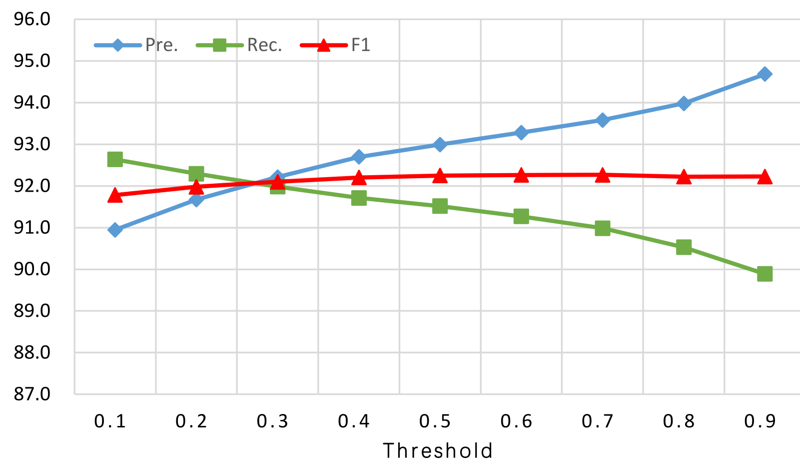

Fig. 5. Results of our method under different tag thresholds.

Table 6
Results of relational triple elements.

Element	NYT			WebNLG		
	Prec	Rec	F1	Prec	Rec	F1
S	95.8	93.7	94.7	98.9	95.2	97
O	95.4	94.3	94.9	98.4	94.9	96.6
R	96.8	94.4	95.6	97.2	93.8	95.4
(S, R)	95	92.7	93.8	95.8	92.8	94.3
(R, O)	94.4	92.8	93.6	96.2	93.2	94.7
(S, O)	93	91.8	92.4	96.5	93.8	95.1
(S, R, O)	93	91.5	92.2	94.8	92.2	93.5

Table 7
Case study for our method.

Instance
#1 Sentence: Scott Schwind has joined Fulbright & Jaworski as an energy partner in Houston.
Ground Truth: (Jaworski, place_founded, Houston)
forward triple extraction: (Schwind, company, Jaworski), (Jaworski, place_founded, Houston)
BRASK(bidirectional extraction): (Jaworski, place_founded, Houston)
#2 Sentence: The mass attempt by several hundred Africans to enter the enclave on Thursday follows a sharp increase over the last two months in the number of migrants to attempt crossing the two razor wire fences that separate Morocco from Ceuta and Spain 's other North African enclave, Melilla.
Ground Truth: (Melilla, country, Spain), (Spain, administrative_divisions, Melilla) (Spain, administrative_divisions, Ceuta), (Spain, contains, Melilla) (Ceuta, country, Spain)
BRASK(bidirectional extraction): (Melilla, country, Spain), (Spain, administrative_divisions, Melilla) (Spain, administrative_divisions, Ceuta), (Spain, contains, Melilla) (Ceuta, country, Spain)

it is relatively stable compared with precision and recall. These results indicate that our model will extract different subjects and objects under different tag thresholds. With an increase in the threshold, the model will extract more accurate entities, which indicates that the precision of our model increases in triple extraction. Our model achieves the best results on the NYT dataset with thresholds of 0.5, 0.6 and 0.7. Even when the threshold is set to 0.1, the F1 of BRASK is still close to 92%, which proves that BRASK can extract relational triples with high confidence.

4.9. Error analysis

To examine the factors that influence the predicted relational triples of our method, we analyze the results of predicting different components in relational triple *(S, R, O)*, where S indicates the subject, O indicates the object and R is the corresponding relation. S is correctly extracted if the subject of the predicted triple is true, as are O and R. If both the subject and relation of the extracted triple are true, the component *(S, R)* is considered correct, regardless of the validity of the extracted object.

Table 6 presents the results of different relational triple elements. The performances of extracting the subject S, relation R and object O are excellent, which demonstrates the effectiveness of our method. For the NYT dataset, we observe that our model more accurately identifies relations of triples than both subjects and objects. The performance on *(S, O)* is the same as that on the entire triple *(S, R, O)*, reflecting that extracted entity pairs can correctly identify the corresponding relations. However, there is a small performance gap between *(S, R)* and relational triple *(S, R, O)*. This finding implies that a few extracted subjects still fail to identify the corresponding objects.

In contrast, for WebNLG, the F1 score on S and O is higher than that on R. The F1 gap between *(S, O)* and the entire triple *(S, R, O)* is relatively larger than that between *(S, R)/(R, O)* and the entire triple *(S, R, O)*, which reflects that identifying the relations of entity pairs is harder than identifying the entities of relational triples. The reason for this difference is that the number of predefined relations in the WebNLG dataset is 10 times that in the NYT dataset, and it is more challenging to extract relations in WebNLG.

4.10. Case study

The case study of our method is shown in Table 7. For the first case, '*Forward Extraction*' indicates that we just utilize the forward triple extraction method to obtain relational triples without backward triple extraction, which is a unidirectional model. We observe that the forward triple extraction predicts two triples, one of which is marked red, which indicates that the predicted triple is invalid. However, BRASK correctly extracts the target triple, which proves that bidirectional extraction can filter invalid triples and obtain more accurate predicted results.

In the second case, there are overlapping and multiple triples in this sentence. This case contains SEO, EPO and RO classes, and it is comparatively difficult to extract all triples. However, our model successfully identifies these target relational triples, which demonstrates the validity of our method in solving overlapping and multiple triple problems.

5. Conclusion

In this paper, we present a bidirectional relation-guided attention network with semantics and knowledge for joint entity and relation extraction. Our model is a bidirectional extraction framework for identifying relational triples, which contains forward and backward triple extraction. In forward triple extraction, we obtain the semantic relations by pretrained BERT to guide extracted subjects for distinguishing objects. For backward triple extraction, we utilize the knowledge embedding algorithm TransE to learn the knowledge relations, which guide all possible objects to identify the relevant subjects. The semantic and knowledge relations correspond to the general and domain knowledge, and we incorporate them into bidirectional extraction. Bidirectional extraction can learn bidirectional dependencies between subjects and objects and capture more implicit features. Forward triple extraction and backward triple extraction are complementary and can filter predicted triples with low confidence. Experiments on the NYT and WebNLG datasets illustrate that our BRASK achieves significant improvements. The additional experiments demonstrate the superiority of handling overlapping and multiple triple extraction scenarios.

CRediT authorship contribution statement

Yi Yang: Conceptualization, Methodology, Software, Validation, Formal analysis, Visualization, Writing – original draft. **Shangbo Zhou:** Supervision, Resources, Writing – review & editing. **Yuxuan Liu:** Investigation, Data curation.

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.

Data availability

Data will be made available on request.

Acknowledgments

We thank all anonymous reviewers for their constructive comments. This work is supported by the National Key R&D Program of China (No. 2017YFB1402405-5), the Fundamental Research Funds for the Central Universities, China (No. 2020CDCGJSJ0042) and the Natural Science Foundation of China (Grant No. 62272070).

References

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. *Advances in Neural Information Processing Systems, 26*.

Chan, Y. S., & Roth, D. (2011). Exploiting syntactico-semantic structures for relation extraction. In *Proceedings of the 49th annual meeting of the Association for Computational Linguistics: Human language technologies* (pp. 551–560).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

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* (pp. 1409–1418).

Gardent, C., Shimorina, A., Narayan, S., & Perez-Beltrachini, L. (2017). Creating training corpora for nlg micro-planning. In *55th Annual meeting of the Association for Computational Linguistics*.

Gormley, M. R., Yu, M., & Dredze, M. (2015). Improved relation extraction with feature-rich compositional embedding models. arXiv preprint arXiv:1505.02419.

Kate, R., & Mooney, R. (2010). Joint entity and relation extraction using card-pyramid parsing. In *Proceedings of the fourteenth conference on computational natural language learning* (pp. 203–212).

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)* (pp. 917–928).

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

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)* (pp. 402–412).

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 AFNLP* (pp. 1003–1011).

Miwa, M., & Bansal, M. (2016). End-to-end relation extraction using lstms on sequences and tree structures. arXiv preprint arXiv:1601.00770.

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* (pp. 1858–1869).

Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, *30*(1), 3–26.

Nayak, T., & Ng, H. T. (2020). Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In *Proceedings of the AAAI conference on artificial intelligence, vol. 34, no. 05* (pp. 8528–8535).

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

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Wang, Y., Yu, B., Zhang, Y., Liu, T., Zhu, H., & Sun, L. (2020). Tplinker: Single-stage joint extraction of entities and relations through token pair linking. arXiv preprint arXiv:2010.13415.

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* (pp. 1476–1488).

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

Yang, Y., Li, X., & Li, X. (2021). A relation-guided attention mechanism for relational triple extraction. In *2021 International joint conference on neural networks* (pp. 1–8). IEEE.

Yu, X., & Lam, W. (2010). Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach. In *Coling 2010: Posters* (pp. 1399–1407).

Yu, B., Zhang, Z., & Su, J. (2019). Joint extraction of entities and relations based on a novel decomposition strategy. arXiv preprint arXiv:1909.04273.

Yuan, Y., Zhou, X., Pan, S., Zhu, Q., Song, Z., & Guo, L. (2020). A relation-specific attention network for joint entity and relation extraction. In *International joint conference on artificial intelligence 2020* (pp. 4054–4060). Association for the Advancement of Artificial Intelligence (AAAI).

Zelenko, D., Aone, C., & Richardella, A. (2003). Kernel methods for relation extraction. *Journal of Machine Learning Research*, *3*(Feb), 1083–1106.

Zeng, X., Zeng, D., He, S., Liu, K., & Zhao, J. (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)* (pp. 506–514).

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 AAAI conference on artificial intelligence, vol. 34, no. 05* (pp. 9507–9514).

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. arXiv preprint arXiv:1706.05075.

Zhou, G., Su, J., Zhang, J., & Zhang, M. (2005). Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting of the Association for Computational Linguistics* (pp. 427–434).