WLINKER: MODELING RELATIONAL TRIPLET EXTRACTION AS WORD LINKING

Yongxiu Xu[1,2], Chuan Zhou[2,3], Heyan Huang[4], Jing Yu[1,2], Yue Hu[1,2],*

[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3]Academy of Mathematics and Systems, Science Chinese Academy of Sciences, Beijing, China
[4]School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China

ABSTRACT

Relational triplet extraction (RTE) is a fundamental task for automatically extracting information from unstructured text, which has attracted growing interest in recent years. However, it remains challenging due to the difficulty in extracting the overlapping relational triplets. Existing approaches for overlapping RTE, either suffer from exposure bias or designing complex tagging scheme. In light of these limitations, we take an innovative perspective on RTE by modeling it as a word linking problem that learns to link from subject words to object words for each relation type. To this end, we propose a simple but effective multi-task learning model, WLinker, which can extract overlapping relational triplets in an end-to-end fashion. Specifically, we perform word link prediction based on multi-level biaffine attention for leaning the word-level correlations under each relation type. Additionally, our model joint entity detection and word link prediction tasks by a multi-task framework, which combines the local sequential and global dependency structures of words in sentence and captures the implicit interactions between the two tasks. Extensive experiments are conducted on two benchmark datasets NYT and WebNLG. The results demonstrate the effectiveness of WLinker, in comparison with a range of previous state-of-the-art baselines.

***Index Terms*—** Relational triplet extraction, Overlapping relations, Multi-task learning, Text mining

1. INTRODUCTION

Extracting relational triplets from unstructured text is a crucial technology for constructing knowledge graph, which aims to extract both entities and semantic relations in the form of $(subject, relation, object)$. Traditional approaches [1, 2, 3] solve this task in a pipeline manner: (1) detecting all possible entity mentions by an entity model, and then (2) extracting relations between the recognized entities by a relation model. Such pipeline method, while flexible and intuitive, suffers from the disadvantages of error cascading, high computational cost and the lack of interaction between the two tasks.

To tackle these issues, subsequent works are dedicated to adopt neural joint learning method [4, 5, 6]. to extract entity span and relation simultaneously Despite achieving better performance than pipeline method, the complicated relation structures still pose great challenges for RTE task. Zeng et.al. [7] defines three types of triplets according to triplet overlap degree, including: *Normal* where triplets have no overlapping entities, *SingleEntityOverlap(SEO)* where two triplets share only one overlapping entity with each other, and *EntityPairOverlap(EPO)* where two triplets share both subject and object entities with each other. See the examples in the upper part of Fig. 1,

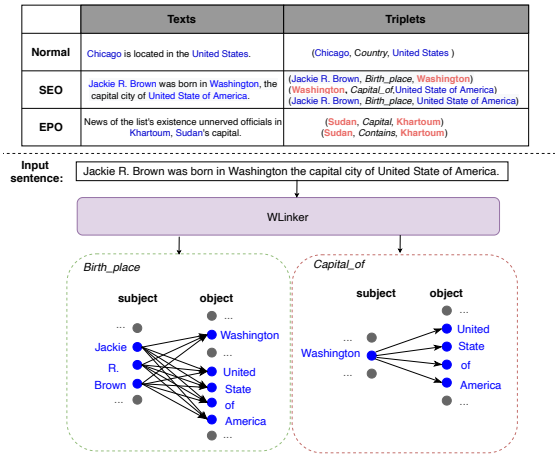


Fig. 1. Examples of Normal, SEO, and EPO triplet cases and the process of our model for the second example sentence. Entities in text are highlighted with blue color and the overlapping entities are marked in bold with red color. Notice that three relational triplets in SEO case are overlapped with each other, but we only highlight one group.

in *SEO* case, triplet *(Jackie R. Brown, Birth_palce, Washington)* only shares entity *Washington* with triplet *(Washington,Capital_of, United States of America)*, while in *EPO* case, two triplets share the entity pair *(Sudan, Khartoum)* with each other.

More recently, various joint neural models were proposed to extract overlapping relational triplets, such as seq2seq-based [7, 9, 10], decomposition-based [11, 12, 13, 14] and handshaking tagging-based models [15, 8]. Although these methods have achieved promising performance, there remains several key limitations associated with them. Typically, seq2seq-based and decomposition-based models have difficulty with the same issue: exposure biass [16, 17]. Specifically, seq2seq-based models use the gold triplets of previous step to guide decoder to generate triplets for current step during the train process, while in the testing phase, the triplets of previous step are generated by the trained model, leading to a gap between training and testing. Similarly, for decomposition-based models, at training time, the gold subject entity as specific input to guide the model extract object entities and relations, while the input subject is given by the trained model at the testing time. Exposure bias could generate accumulated errors especially when a sentence contains multiple triplets, since a skewed prediction will further deviate the prediction of the follow-up process. In contrast, handshaking tagging-based model [15], decomposes RTE

*Corresponding author: Heyan Huang (E-mail: hhy63@bit.edu.cn)

into several uncoupled subtasks to avoid the exposure bias. However, they need designing complex tagging scheme for each subtask and deploying complex inference processes. Therefore, how to effectively extract overlapping relational triplets in an end-to-end manner is still an open problem.

In this paper, we cast a new perspective by viewing RTE as a word linking problem, motivated by the link prediction problem in graph leaning which aims to predict potential edges between arbitrarily two unconnected vertices in graph. Here, we consider each word in text as a node in graph, then the relation extraction of *(subject-word, object-word)* pairs is a sort of link prediction from the graph. Considering the same word pair could have multiple relations, we perform word linking prediction in each relation space, as shown at the bottom of Fig. 1. To this end, we propose a simple but effective neural joint model for extracting relational triplets (potentially overlapping) in an end-to-end manner named **WLinker**. Benefiting from end-to-end architectures and the novel perspective, WLinker is capable of avoiding from the exposure bias issue without designing complex handshaking tagging scheme.

Techinically, WLinker has the following two advantages: (1) It captures the distinctive word-level correlations under different relation type by a multi-label classifier based on multi-level biaffine attention. Biaffine attention has been widely used for syntactic or semantic word-to-word dependency parsing [18]. (2) It incorporates the local and global word-level dependency structures by a multi-task learning framework, to enhance the performance. We conduct extensive experiments on two widely used relation extraction datasets that support evaluating the ability of extracting multiple overlapping relational triplets in a single sentence. Experimental results demonstrate that WLinker significantly outperforms a range of state-of-the-art baselines.

2. METHODOLOGY

In this paper, instead of modeling relations as classes of entity pairs, we model relations as labels of word pairs, i.e. further deconstructing RTE into a multi-label classification problem for all word pairs in a sentence. Fig. 2 shows the overall neural architecture of WLinker model. The architecture consists of three components: (1) the Contextual Encoder module transforms the input words into the contextual word representations which are shared across the following task modules. (2) the Entity Detection module is a auxiliary task to recognize the entities contained in sentence by a sequence labeling method, which captures the local dependencies of word sequence. (3) the Word Linking module is the core component of our model, which captures the word-level correlation semantics for each relation type from global aspect. By jointly learning entity detection and word linking in a unified multi-task learning framework, our model can fuse the local linear and global dependency structures of word sequences, as well as capture the implicit interaction between the two tasks.

2.1. Contextual Encoder Module

In this module, we implement two kinds of encoders as LSTM [19] and BERT [20] respectively. Specifically, suppose that an input sentence S consists of N words, i.e., $S = [w_1, w_2, ..., w_N]$. For LSTM encoder, we use $\boldsymbol{w}_i$ to denote the concatenation of the pre-trained Glove word embeddings [21], the bidirectional LSTM (BiLSTM) character embeddings [22] and the position embeddings for w_i. We further utilize a BiLSTM to produce the contextualized word embedding sequence $\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_N$:

$$\boldsymbol{h}_i = [\overrightarrow{LSTM}(\boldsymbol{w}_i, \overrightarrow{\boldsymbol{h}}_{i-1}), \overleftarrow{LSTM}(\boldsymbol{w}_i, \overleftarrow{\boldsymbol{h}}_{i+1})] \tag{1}$$

where each $\boldsymbol{h}_i \in \mathbb{R}^{d_w}$, d_w is the dimension of hidden states for BiLSTM.

For BERT encoder, we first tokenize each word w_i into word pieces $\tilde{w}_i$, and then feed them into a pre-trained BERT module to project the input into a sequence of contextual vectors $\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_N$:

$$\boldsymbol{h}_i = BERT([CLS], \tilde{w}_i, [SEP]) \tag{2}$$

Note that we do not combine the character embeddings and position embeddings because we assume that they have already been encoded in the BERT embeddings. The contextualized word embeddings are shared across the next two task modules.

2.2. Entity Detection Module

Here, we take entity detection as a sequence labeling task to capture the local linear structures of word sequence. We expect that the lower layers to extract task-universal representations and the upper layers to extract more task-specific representations [23]. Thus, we first couple a BiLSTM layer on the top of the Contextual Encoder, to obtain task-aware word representation sequence $\boldsymbol{h}^E = [\boldsymbol{h}_1^E, \boldsymbol{h}_2^E, ..., \boldsymbol{h}_N^E] \in \mathbb{R}^{N \times d_e}$ for Entity Detection module:

$$\boldsymbol{h}_i^E = \mathrm{BiLSTM}^E(\boldsymbol{h}_i) \tag{3}$$

Subsequently, $\boldsymbol{h}^E$ is fed into a Conditional Random Field (CRF) layer to predict the most probable tag for each word. Let $\boldsymbol{V} \in \mathbb{R}^{k \times k}$, $\boldsymbol{P} \in \mathbb{R}^{N \times k}$ be the transposition matrix and be the state score matrix respectively, where k is the number of possible labels based on BIOE scheme[1]. Given the sentence S, the conditional probability of target label sequence $L^* = [l_1, l_2, ..., l_N]$ can be computed as follows:

$$\begin{aligned} p(L^*|S) &= \frac{exp(score(L, S))}{\sum_{L' \in L_S} exp(score(L', S))} \\ score(L, S) &= \sum_{i=0}^{N} V_{l_i, l_{i+1}} + \sum_{i=1}^{N} P_{i, l_i} \end{aligned} \tag{4}$$

where L_S denotes all possible tag sequences, $V_{l_i, l_{i+1}}$ indicates the transition score from tag l_i to tag l_{i+1}, P_{i,l_i} is the score of tag l_i for the input w_i.

During training, we use the cross entropy loss here as the Entity Detection module loss, denoted as $\mathcal{L}_{ed}$:

$$\mathcal{L}_{ed} = \sum_{L \in L_S} log\, p(L|S) \tag{5}$$

2.3. Word Linking Module

We further take the word linking problem under each relation type as a multi-label classification problem. Concretely, we calculate correlation intensities of each word pair under each relation type by investigating a multi-level baiffine attention. We expect that multi-layer encoders can gradually transform from a low-level word representations (with shallow semantics) into a more abstract high-level representations (with deep semantics), according to [24]. Thus, we

[1]Here, we use the BIOE (Beginning, Inside, Others,Ending) tagging scheme for entity detection

firstly apply a deep BiLSTM of L-levels to a sequence of contextual embeddings $\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_N$:

$$\begin{aligned} \boldsymbol{h}_{i,0}^{W} &= \boldsymbol{h}_i \\ \boldsymbol{h}_{i,l}^{W} &= \text{BiLSTM}_i^{W}(\boldsymbol{h}_{1,l-1}^{W}, ..., \boldsymbol{h}_{N,l-1}^{W}) \end{aligned} \tag{6}$$

where $\boldsymbol{h}_{i,l}^{W}$ indicates the i-th hidden embeddings of the BiLSTM of Word Linking module at the l-th layer. Here, we set L to be 3 with the best performance on the validation set.

Considering the subject and object words appear in different positions in a sentence, i.e. the contexts of them are different. We project the hidden representation for each l-th layer $\boldsymbol{h}_{i,l}^{W}$ through two separate MLPs to generate **subjec/object-sensitive entity representations**. In doing so, the pair matrix output by the biaffine attention operation is asymmetric and directional-aware. The subject representation and object representation of i-th word for each l-layer are calculated as follows:

$$\begin{aligned} \boldsymbol{h}_{i,l}^{(s)} &= \text{MLPs}^{(\text{s})}(\boldsymbol{h}_{i,l}^{W}) \\ \boldsymbol{h}_{i,l}^{(e)} &= \text{MLPs}^{(\text{e})}(\boldsymbol{h}_{i,l}^{W}) \end{aligned} \tag{7}$$

We use the concatenate operation to aggregate the subject and object representations of all levels:

$$\begin{aligned} \boldsymbol{h}_{i}^{(s)} &= [\boldsymbol{h}_{i,1}^{(s)}; \boldsymbol{h}_{i,2}^{(s)}; \boldsymbol{h}_{i,3}^{(s)}] \\ \boldsymbol{h}_{i}^{(e)} &= [\boldsymbol{h}_{i,1}^{(e)}; \boldsymbol{h}_{i,2}^{(e)}; \boldsymbol{h}_{i,3}^{(e)}] \end{aligned} \tag{8}$$

Then, we apply the biaffine attention on the $\boldsymbol{h}_{i}^{(s)}$ and $\boldsymbol{h}_{i}^{(e)}$, as follows:

$$link(i, r, j) = (\boldsymbol{h}_{j}^{(e)})^{T} \boldsymbol{U}_r \boldsymbol{h}_{i}^{(s)} + \boldsymbol{V}_r \boldsymbol{h}_{i}^{(s)} \tag{9}$$

where $link(i, r, j)$ indicates the score of the dependency between word pair (w_i, w_j) under relation $r \in \mathcal{R}$, $\mathcal{R}$ is the predefined relations set. The matrices $\boldsymbol{U}_r \in \mathbb{R}^{d \times d}$ and $\boldsymbol{V}_r^{\mathrm{T}} \in \mathbb{R}^{d}$ are biaffine parameters for relation r, d is the dimension of the subject and object representations. Owing to the biaffine attention, we can directly model both the compatibility of $\boldsymbol{h}_{i}^{(s)}$ and $\boldsymbol{h}_{j}^{(e)}$ by a bi-linear attention operation $(\boldsymbol{h}_{j}^{(e)})^{T} \boldsymbol{U}_r \boldsymbol{h}_{i}^{(s)}$, and the prior likelihood of $\boldsymbol{h}_{i}^{(s)}$ having a dependency link by $\boldsymbol{V}_r \boldsymbol{h}_{i}^{(s)}$.

Finally, we employ $sigmoid$ function on the $link(i, r, j)$, yielding $p(i, r, j)$ which represents the probability of the existence of dependency link between the i-the and the j-th word under r-th relation.

$$p(i, r, j) = sigmoid(link(i, r, j)) \tag{10}$$

We use the binary cross entropy loss here as the Word Linking module loss, denoted as $\mathcal{L}_{wl}$:

$$\begin{aligned} \mathcal{L}_{wl} = -\sum_{r=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{N}[y(i, r, j) log p(i, r, j) + \\ (1 - y(i, r, j)) log(1 - p(i, r, j))] \end{aligned} \tag{11}$$

Where $y(i, r, j) \in \{0, 1\}$, and $y(i, r, j) = 1$ denotes the fact that the word pair (w_i,w_j) have the relation r.

During training, Entity Detection module and Word Linking module are trained jointly, which on one hand captures the implicit interactions between two tasks and more importantly on the other hand fuses the local linear and global dependency structures of word sequence. Thus, we combine the above two task losses as the training loss:

$$\mathcal{L} = \alpha \mathcal{L}_{ed} + (1 - \alpha) \mathcal{L}_{wl} \tag{12}$$

where α ia a hyper-parameter for controlling the weight of the two tasks.

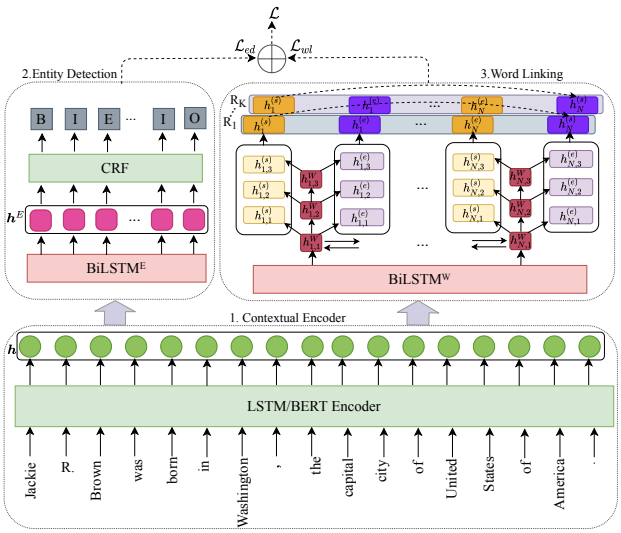


Fig. 2. The architecture of our WLinker model which cinsists of the Contextual Encoder, Entity Recognition and Word Linking modules.

3. EXPERIMENTS

3.1. Experimental Setting

Datasets & Evaluation Metrics. Following previous works, we conducted comprehensive experiments on two popular used datasets **NYT** [25] and **WebNLG** [26]: NYT is first produced by distant supervised relation task (DSRE), which contains 1.18M sentences and 24 predefined relation types; WebNLG was originally created by [26] for Natural Language Generation task and modified by [7] for Relational Triplet Extraction task. We adopt precision (Prec.), recall (Rec.) and standard micro-F1 (F1) to evaluate the ability of relational triplet extraction.

Implementation details. We implement our model using PyTorch, and all the models are trained on a personal workstation with Intel Xeon E5 2.2GHz CPU, NVIDIA GeForce GTX 1080 Ti GPUs and 128 GB memory. We train the model using Adam with learning rate 0.001, the learning rate decay 0.95 and batch size 10. We initialize the word embeddings with the 300-dimension Glove 840B embeddings [21]. The character embeddings and position embeddings are randomly initialized with 100-dimensions and 50-dimensions respectively. We set the dimension of hidden states for BiLSTM to 200. For BERT ecoder, we use the base cased English model with 768-dimension. Each MLP layer has 2 hidden layers with the hidden state of 200-dimensions and the activation function is set to ReLU. The dropout rate is set to 0.4 for regularization. The loss weight α is set to 0.3.

3.2. Main Results

We compare our model with several state-of-the-art (SoTA) models on both datasets, including: **NovelTagging** [6] proposes a novel tagging scheme which transforms the RTE task into a sequence labeling problem; **CopyMTL** [10] and **WDec** [9] are two seq2seq-based models with copy mechanism; **ETL-Span** [12] and **CasRel** [13] are two decomposition-based models for extracting overlapping relational triplets; **TPLinker** [15] proposes a multi-task learning model for overlapping RTE by designing a complex handshaking tagging schema.

Model	NYT			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1
NovelTagging* [6]	32.8	30.6	31.7	52.5	19.3	28.3
CopyMT [7]	75.7	68.7	72.0	58.0	54.9	56.4
WDec* [9]	88.1	76.1	81.7	88.6	51.3	65.0
ETL-Span [12]	85.5	71.7	78.0	84.3	82.0	83.1
CasRel$_{lstm}$* [13]	83.8	83.0	83.4	86.1	80.2	83.0
TPLinker$_{lstm}$ [15]	86.0	82.0	84.0	91.9	81.6	86.4
WLinker	86.3	88.0	**87.1**	88.4	92.2	**90.3**
CasRel$_{bert}$ [13]	89.7	89.5	89.6	90.1	88.5	89.3
TPLinker$_{bert}$ [15]	91.4	92.6	92.0	88.9	84.5	86.7
WLinker$_{bert}$	**91.8**	**93.2**	**92.5**	**90.5**	**92.4**	**91.4**

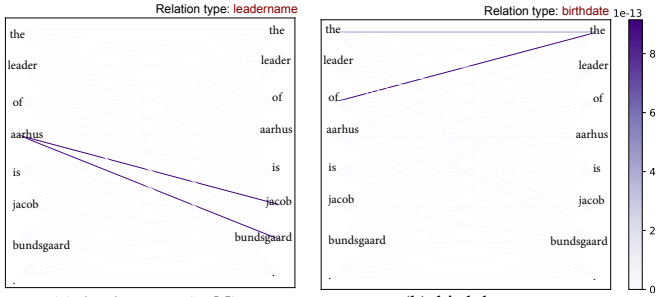
Table 1. Main results of different methods on NYT and WebNLG datasets. The results of the methods marked with $*$ are our re-implementation, and all improvements of our model are significant ($p \leq 0.05$).

Model	NYT(F1)	WebNLG(F1)
CopyMTL [10]	75.6	78.2
CasRel$_{lstm}$ [13]	89.7	93.5
WLinker w/o Word Linking	91.2	94.0
WLinker	**93.3**	**96.1**

Table 2. Results of entity detection on NYT and WebNLG.

Model	NYT		
	Prec.	Rec.	F1
WLinker	86.3	88.0	87.1
(a) w/o Entity detection	84.8	86.1	85.4(↓1.7)
(b) w/o MLPs	85.1	85.6	85.3(↓1.8)
(c) w/o biaffine attention	85.3	84.8	85.0(↓2.1)

Table 3. Ablation test on NYT test set.



(a) leadername (**gold**) (b) birthdate

Fig. 3. The visualization of attention matrices under two relation types. The gold triplet contain in this case is "(*aarhus, **leadername**, jacob bundsgaard*")

As shown in Table 1, overall, our proposed model outperforms all baselines on these datasets in term of F1-score. **WLinker**$_{bert}$ outperforms TPLinker$_{bert}$ and Cascade$_{bert}$ by 4.7% and 2.1% in F1-scores on WebNLG, respectively, and achieves comparable F1-socre with TPLinker$_{bert}$ on NYT. Additionally, **WLinker** achieves a vast amount of performance boost over TPLinker$_{lstm}$, i.e., +3.1%, +3.9%, respectively on NYT and WebNLG. We attribute the performance gains of **WLinker** into its three advantages: (1) Extracting relational triplets using end-to-end learning manner without error propagation; (2) Focusing more on the relation-related word pairs by the multi-level biaffine attention; (3) Fusing the local sequential and global dependency structures based on a multi-task learning framework. The performance gains from **WLinker** to **WLinker**$_{bert}$ demonstrate the importance of prior knowledge in a pre-trained language model.

We can see that **WLinker** outperforms the seq2seq-based and decomposition-based models on all metric results, which further demonstrates its ability of mitigating the issues of exposure bias. We can also observe that our model can obtain better Recall results on both NYT and WebNLG datasets. We conjecture this is largely due to that our model focus finer-grained (word-level instead of span-level) semantics for RTE.

Besides, we also investigate the performance of our model on entity detection task, and the results are shown in Table 2. It can be seen that **WLinker** outperforms CasRel$_{lstm}$ by 3.6% and 2.6% on NYT and WebNLG, respectively. CopyMTL is worst because of the limitation of their auto-regressive decoder. When the word linking module is removed, the results of **WLinker** are significantly reduced, which indicates that word linking module can provide additional information.

3.3. Detailed Analysis

Firstly, to explore the importance of the components of our model, we conduct ablation experiments on NYT dataset by removing one component at a time. And then, to show how our model effectively models the word-level correlation semantics under different relation types, we visualize two relation-specific biaffine attention matrices.

As shown in Table 2, each component of our model has a considerable contribute to the effectiveness of our model. We conjecture this is largely due to that they capture rich word semantic information for RTE task either from local aspect (e.g. Entity detection and MLPs components) or from global aspect (Multi-level biaffine attention component). As shown in Figure 3, our model can capture the correct word dependency under the correct relation type. We also find that the correlation intensities between words under the gold relation type (i.e. *leadername*) is much greater than that under the other relation type (e.g. *birthdate*), which further certifies that our model can effectively learn the dependencies between words through multi-level biaffine attention.

4. CONCLUSION

In this paper, we present a novel perspective that regards relational triplet extraction as word linking problem that predicts links from subject words to object words under each relation type. We propose a simple but effective end-to-end model which joints entity detection and word linking prediction using a multi-task learning framework. Based on this new paradigm, our model can fuse the local and global word dependency structures of word sequence, capture word-level correlation semantics under each relation type, and avoid from the exposure bias issue and complex tagging scheme. The experimental results demonstrate that our model outperforms a range of baselines.

5. ACKNOWLEDGMENTS

This work was supported by the National Key Research and Development Program of China (No.2021YFB3100600), the NSFC (No. 61872360), and the CAS Project for Young Scientists in Basic Research (No. YSBR-008).

6. REFERENCES

[1] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella, "Kernel methods for relation extraction," *J. Mach. Learn. Res.*, vol. 3, pp. 1083–1106, 2003.

[2] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky, "Distant supervision for relation extraction without labeled data," in *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*. 2009, pp. 1003–1011, The Association for Computer Linguistics.

[3] Yee Seng Chan and Dan Roth, "Exploiting syntactico-semantic structures for relation extraction," in *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*. 2011, pp. 551–560, The Association for Computer Linguistics.

[4] Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy, "Table filling multi-task recurrent neural network for joint entity and relation extraction," in *COLING*, Dec. 2016, pp. 2537–2547.

[5] Arzoo Katiyar and Claire Cardie, "Going out on a limb: Joint extraction of entity mentions and relations without dependency trees," in *ACL*, July 2017, pp. 917–928.

[6] Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu, "Joint extraction of entities and relations based on a novel tagging scheme," in *ACL*, 2017, pp. 1227–1236.

[7] Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao, "Extracting relational facts by an end-to-end neural model with copy mechanism," in *ACL*, July 2018, pp. 506–514.

[8] Yongxiu Xu, Heyan Huang, Chong Feng, Chuan Zhou, Jiarui Zhang, and Yue Hu, "A relation-aware attention neural network for modeling the usage of scientific online resources," in *International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021*. 2021, pp. 1–8, IEEE.

[9] Tapas Nayak and Hwee Tou Ng, "Effective modeling of encoder-decoder architecture for joint entity and relation extraction," in *AAAI*, pp. 8528–8535.

[10] Daojian Zeng, Haoran Zhang, and Qianying Liu, "Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning," in *AAAI*, 2020, pp. 9507–9514.

[11] Dai Dai, Xinyan Xiao, Yajuan Lyu, Shan Dou, Qiaoqiao She, and Haifeng Wang, "Joint extraction of entities and overlapping relations using position-attentive sequence labeling," in *AAAI*, pp. 6300–6308.

[12] Yu Bowen, Zhenyu Zhang, Jianlin Su, Yubin Wang, Tingwen Liu, B. Wang, and Sujian Li, "Joint extraction of entities and relations based on a novel decomposition strategy," in *ECAI*, 2020, pp. 2282–2289.

[13] Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang, "A novel cascade binary tagging framework for relational triple extraction," in *ACL*, July 2020, pp. 1476–1488.

[14] Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang, "A hierarchical framework for relation extraction with reinforcement learning," in *AAAI*, 2019, pp. 7072–7079.

[15] Yucheng Wang, Bowen Yu, Yueyang Zhang, Tingwen Liu, Hongsong Zhu, and Limin Sun, "TPLinker: Single-stage joint extraction of entities and relations through token pair linking," in *ICCL*, Dec. 2020, pp. 1572–1582.

[16] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba, "Sequence level training with recurrent neural networks," in *ICLR*, Yoshua Bengio and Yann LeCun, Eds., 2016.

[17] Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu, "Bridging the gap between training and inference for neural machine translation," in *ACL*, July 2019, pp. 4334–4343.

[18] Timothy Dozat and Christopher D. Manning, "Deep biaffine attention for neural dependency parsing," in *ICLR*, 2017.

[19] Klaus Greff, R. Srivastava, J. Koutník, Bas R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, pp. 2222–2232, 2017.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL:HTL*, June 2019, pp. 4171–4186.

[21] Jeffrey Pennington, Richard Socher, and Christopher Manning, "GloVe: Global vectors for word representation," in *EMNLP*, Oct. 2014, pp. 1532–1543.

[22] Cícero Nogueira dos Santos and Bianca Zadrozny, "Learning character-level representations for part-of-speech tagging," in *ICML*, 2014, vol. 32 of *JMLR Workshop and Conference Proceedings*, pp. 1818–1826.

[23] Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith, "Linguistic knowledge and transferability of contextual representations," in *NAACL*, 2019.

[24] Hsin-Yuan Huang, Chenguang Zhu, Yelong Shen, and Weizhu Chen, "Fusionnet: Fusing via fully-aware attention with application to machine comprehension," in *International Conference on Learning Representations*, 2018.

[25] Sebastian Riedel, Limin Yao, and Andrew McCallum, "Modeling relations and their mentions without labeled text," in *PKDD*, 2010, pp. 148–163.

[26] Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini, "Creating training corpora for NLG micro-planners," in *ACL*, July 2017, pp. 179–188.