Systematic Analysis of Joint Entity and Relation Extraction Models in Identifying Overlapping Relations

Yuchen Luo[1], Zhenjie Huang[1], Kai Zheng[2](✉), and Tianyong Hao[1]

[1] School of Computer Science, South China Normal University, Guangzhou, China
{2019022611,2020022962,haoty}@m.scnu.edu.cn
[2] Network Center, South China Normal University, Guangzhou, China
david@scnu.edu.cn

Abstract. Named entity recognition and relation extraction are two fundamental tasks in the domain of natural language processing. Joint entity and relation extraction models have attracted more and more attention due to the performance advantage. However, there are difficulties in identifying overlapping relations among the models. To investigate the differences of structures and performances of joint extraction models, this paper implements a list of state-of-the-art joint extraction models and compares their difference in identifying overlapping relations. Experiment results show that the models by separating entity features and relation features work better than the models with feature fusion in identifying overlapping relations on three publicly available datasets.

Keywords: Named entity recognition · Relation extraction · Overlapping relation · Feature separation

1 Introduction

With the development of natural language processing, research in information extraction has attracted more and more attention. The research of information extraction are usually divided into three main tasks: named entity recognition, relation extraction, and event extraction. Named entity recognition, as the core foundation of other information extraction, aims to identify the mentions that represent named entities from a natural language text and label their locations and types [1]. The main purpose of relation extraction is to extract semantic relation between named entities from natural language text, i.e., to determine the classes of relations between entity pairs in unstructured text based on entity recognition, as well as to form structured data for computation and analysis [2]. After named entity recognition and relation extraction, the extracted entities and relations can be represented in the form of a triple (*Entity 1*, *Relation*, *Entity 2*). For example, in the sentence “He was somewhat agitated, so his Keppra was switched to Topiramate”, “Agitated” and “Topiramate” are named entities in “Reason” and “Drug” types, respectively. The entities have a certain relation “Reason-Drug”. Thus they can be expressed as a triple (“Agitated”, “Reason-Drug”, “Topiramate”).

H. Zhang et al. (Eds.): NCAA 2021, CCIS 1449, pp. 17–31, 2021.
https://doi.org/10.1007/978-981-16-5188-5_2

The traditional entity-relation extraction models frequently utilize the pipeline manner, which is performed in two independent steps, named entity recognition and then relation extraction sequentially. The pipeline manner completely separates the named entity recognition and relation extraction tasks, while neglecting the relevant information between the two tasks, leading to an error transfer problem [3]. Unlike the pipeline manner, joint extraction models deal with both named entity recognition and relation extraction tasks simultaneously for solving the error transfer problems. Recent studies have shown that joint learning methods can effectively integrate entity and relation information to obtain better performance in both tasks [4, 5]. However, several joint models, such as SPTree [6] and Novel Tagging [7], although they can improve the performance for solving the error transfer problem, still suffer from the problem of ineffective identifying overlapping relations.

Overlapping relations mean that there are multiple relations in a natural language sentence, and there may be an overlap between multiple relations. Zeng et al. [8] first carefully classified overlapping relations into two types: EntityPairOverlap (EPO) in which two entities are overlapping in two relations, as well as SingleEntityOverlap (SEO) in which only one entity is overlapping in two relations. Despite the increasing number of overlapping relation studies in recent years, identifying overlapping relations is still a difficult problem to overcome for the relation extraction research.

This paper focuses on comparing and analyzing the performance of recent joint extraction models and the capability of identifying overlapping relations. Four state-of-the-art joint extraction models are implemented: Novel Tagging [7], Multi-head [9], Two are better than one [10], and TPLinker [11]. In terms of feature processing, Novel Tagging and TPLinker represent a fusion of entity and relation features based on an annotation scheme, while Multi-head and Two are better than one use a form of feature separation and information interaction between entity and relation features via a specific structure. The models except Novel Tagging are able to identify overlapping relations via different strategies. This paper analyzes these models on the following three datasets: NYT, CoNLL04, and N2C2 2018 (Track2). By comparing the performance of the four models, this paper investigates the differences in identifying overlapping relations, and analyzes the advantages and disadvantages of feature fusion and feature separation strategies. Experiment results show that the joint models with identifying overlapping relation strategies outperform the other models, and the feature separation strategy has better performance than feature fusion in identifying overlapping relations.

The main contributions of this paper are the following three aspects: 1) implemented the four latest joint extraction models and compared the differences of model structures and performance; 2) compared and analyzed the performance of four joint extraction models in identifying overlapping relations on multiple datasets; 3) discovered a new finding that the feature separation strategy has advantage than the feature fusion strategy in identifying overlapping relations.

2 Related Work

In recent years, researchers have proposed many models to extract entities and relations. In terms of traditional pipeline manner, Hoffimann et al. [12] utilized external components and knowledge-based approaches to extract entities and relations, while a large number of researchers [13–16] adopt deep neural network approaches. In terms of joint extraction models, Miwa and Bansal [6] firstly used Bi-LSTM to implement a named entity recognition model, and then combine Bi-TreeLSTM as well as dependency analysis trees to obtain relations between entities, which is the first using neural networks for joint entity and relation extraction. However, this approach needs to rely on external resources, such as the need to perform dependency analysis on sentences in the extraction of entities and relations. To reduce the dependency on external resources, Bekoulis et al. [9] proposed to treat relation extraction as a multi-head selection problem. After identifying entities, this method assumed that each entity might be the head entity of a certain relation, and then recognized relations with all other identified entities. Based on the framework of multi-head recognition, Zhao et al. [17] introduced a feature representation of relative positions of entities and a global optimization function GRC to express overall semantic information of the sentence for improvement. Zhao et al. [18] believed that traditional models treated the tasks of named entity recognition and relation extraction differently, which would lose closely related information between the two tasks. Therefore, they proposed a CMAN framework that could apply feature information to the two tasks. Benefiting from the performance improvement brought by the interactions of the two tasks of named entity recognition and relation extraction, the Two are Better than One proposed by Wang and Lu [10] used sequence and table features to replace entity features and relation features respectively, as well as provided a mechanism to enable sufficient interaction between the two features.

The sequence-to-sequence models are an implementation way in the joint extraction models. These methods treated the entity relation extraction task as a translation-like task, by taking text sequences as input and entity-relation triplet sequences as output. Nayak and Ng [19] designed a new representation of entity-relation triples based on the encoder-decoder structure to solve the task of entity and relation extraction at one time. However, this basic sequence-to-sequence problem is prone to a kind of exposure error, which refers to the use of real data values during training, and the use of prediction at the previous moment as input during testing. There is a deviation issue between the distributions learned during training and during testing. Therefore, Zhang et al. [20] proposed a structure of seq2UMTree to reduce the exposure error, that is, the decoding part was no longer decoded in the form of a sequence but in a tree structure. Sui et al. [21] utilized a non-auto regressive model for decoding and proposed Bipartite Matching Loss to eliminate the dependence on the order of entity-relation triples from original sequence to current sequence. Gupta et al. [22] proposed to use lower triangle part of tables to implement the entity-relation extraction task. The features on diagonal line were used to realize named entity recognition, while the features on off-diagonal line were used to implement the relation extraction. Nevertheless, this model was transformed into a sequence for traversal, so there was still a long-term dependency problem. To overcome this problem, Ma et al. [23] used the form of tensor dot product to extract relations.

Tagging is another type of joint extraction model. The design of this method focused on how to design a reasonable tagging to characterize entities and relations, and how to obtain feature expression of tagging through neural networks. Zheng et al. [7] first proposed a novel tagging method, each word had a corresponding tag. A tag was composed of the position of the entity where a word was located, the type of relation that a word participated in, and whether the relation between words was a head entity or a tail entity. Under this tagging, sequence annotation could be used. The method identified a tag for each word and then combines all identified tags to extract entity relations using the principle of nearest matching. Wang et al. [11] proposed a structure of TPLinker, which constructed a tag of entire sequence for each word. This method marked the words that form entity boundary with current word in the sequence. All entities and relations related to current word was identified via the tags of this sequence.

Although the joint extraction models were considered to be utilized to alleviate the gradient error propagation, many scholars had shown that some joint models mixed the features of entities and the characteristics of relations. The mixture of these two features might brought a negative impact to the performance of the models. Zhong and Chen [24] used a simple pipeline model for entity-relation extraction, and finally achieved SOTA results on several datasets. They believed that the pipeline model could avoid the problem of feature fusion and both tasks could be improved if entity feature was properly integrated into the relation extraction task.

Therefore, this paper compares and analyzes two types of joint extraction models with feature fusion and feature separation strategies, as well as investigates the method differences in identifying overlapping relations.

3 Joint Extraction Model Comparison

The structure of joint extraction models is often complex and diverse and can be divided into two types of strategies according to the way the models handle entity features and relation features: feature fusion and feature separation. Different feature processing strategies can affect the overall performance of the models and the recognition performance of overlapping relation identification. To investigate their differences, this paper implements four latest state-of-the-art models, compares and analyzes their structural feature strategies, and analyzes the traits of the models.

3.1 Model Differences

In terms of model structure, the major difference between the two strategies lies in the way of handling entity features and relation features while separating or mixing the features may leads to different performance the models. The strategy of feature separation makes the two tasks of named entity identification and relation extraction clear, but it may lead to gradient error propagation. The feature-fusion strategy is able to extract triples at once, but ignores the information interaction between the two tasks. Figure 1 shows a common joint extraction framework with the two feature processing strategies.

To investigate the influencing factors of the performance of joint extraction models, this paper conducts a comparison of the four models including Multi-head, Two are Better than One, Novel Tagging, and TPLinker. Table 1 shows the difference and relevance of the four models. The comparison consists of feature processing strategy, the way of information interaction, and the applicability of SEO and EPO. Multi-head and Two are

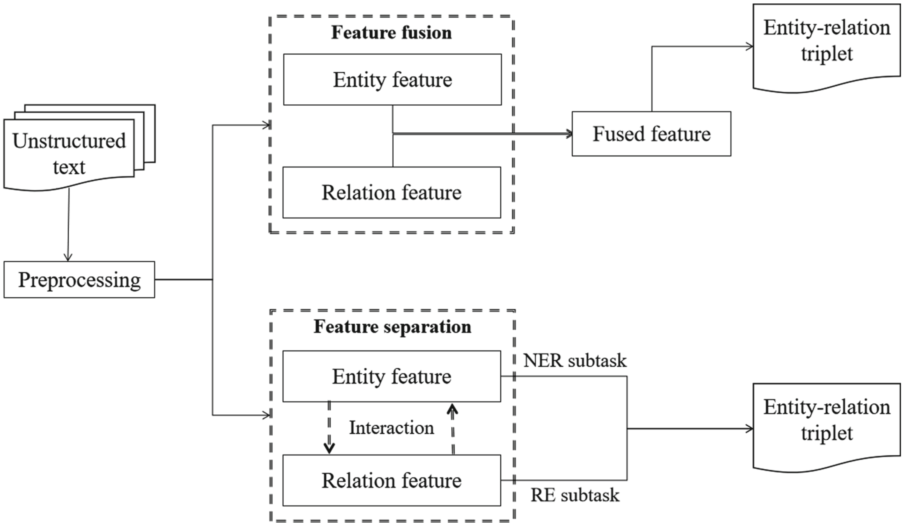

Fig. 1. The illustration of joint extraction framework with feature separation and feature fusion strategies

Better than One are two models based on feature separation strategy. Both models can identify the SEO overlap issue, but cannot identify the EPO overlap issue. Multi-head can only carry out one-direction information exchange, while Two are Better than One can carry out two-direction information exchange. Novel Tagging and TPLinker are two models based on feature fusion strategy. Novel Tagging cannot identify the problem of overlapping relations, while TPLinker can identify the problem of SEO and EPO issues at the same time.

Table 1. Comparison of the four state-of-the-art joint extraction models

Model	Strategy	Interaction	SEO	EPO
Multi-head	Feature separation	One-direction	√	×
Two are better than one	Feature separation	Two-direction	√	×
Novel tagging	Feature fusion	–	×	×
TPLinker	Feature fusion	–	√	√

3.2 Feature Separation Strategy

Feature separation is a strategy to treat entity features and relation features separately in a model. The fact that entities and relations share a complete set of feature representations respectively does not mean that they are unrelated to each other. Through means such as neural network parameter sharing, a model can enhance information interaction between the two features, while the interaction is usually beneficial. Since the two tasks of named entity recognition and relation extraction are related, the performance of a joint model can be improved if information interaction is appropriate.

This paper selects the Multi-head and Two are Better than One models with feature separation strategy. The Multi-head model can only transfer the interactive information of entity features to the relation extraction task in one direction due to its structure limitation. The entity feature information to be transferred is extracted by a Bi-LSTM during named entity recognition. Multi-head regards the task of relation extraction as a multi-head selection problem. The model assumes that any entity may be the head entity of a certain relation, and it is judged whether the relation is established through a Sigmoid function and received entity interaction information. This method can identify situations where there are multiple relations between two entities. Multi-head model consists of five layers: word embedding layer, Bi-LSTM layer, CRF layer, label embedding layer and Sigmoid layer. Word embedding layer maps each token to a word vector using a word2vec model. Bi-LSTM layer encodes information from left to right and right to left by taking the word embedding as input. CRF layer labels the word using a BIO encoding scheme to identify all entity arguments. After that, the label embedding layer maps the entity label of each token to a label vector, and then it concats the label embedding and the output of Bi-LSTM. At last, the Sigmoid layer takes the output of CRF layer to identify for all tokens, the most probable head word of the head entity and the most probable corresponding relation.

Two are Better than One model uses sequence features and table features to replace entity features and relation features respectively. With the four-dimensional features of the table structure, the model can transfer entity features and relation features in both directions. Assuming that the input of each encoding unit in the table structure is the initial input *S0*, the input corresponding to the sequence structure is *Sl-1*. *Xl, i, j* is the input of the sequence feature, and *Tl-1* is the output of the previous table feature unit. According to the four-dimensional features, *l, i, j* represents the different directions. The received interactive information is shown in Eq. (1) and (2).

$$T_{l,i,j} = GRU(X_{l,i,j}, T_{l-1,i,j}, T_{l,i-1,j}, T_{l,i,j-1}) \tag{1}$$

$$X_{l,i,j} = ReLU(Linear([S_{l-1,i}; S_{l-1,j}])) \tag{2}$$

The table encoder takes the sequence feature as input, and concat it as a table. The element in row *i* and column *j* of the table represents the relations of the i-th word and the j-th word in the sentence. Next the Multi-Dimensional Recurrent Neural Network encodes the table feature and output hidden feature. The structure of sequence encoder is similar to the structure of the encoder of transformer. However the self-attention is replaced with a table-guided attention which converts the table feature to sequence feature.

3.3 Feature Fusion Strategy

Feature fusion refers to a strategy in which the model fuses entity features and relation features in a specific way to form a new feature representation. The models with the strategy can complete the task of joint extraction only by extracting and decoding fused features. However, the models usually lose interactive information between entities and relations. The focus of feature fusion is how to blend entity features and relation features, while tagging is one of the solutions.

Novel Tagging and TPLinker are two feature fusion models based on Tagging schemes. Novel Tagging firstly introduces the Tagging scheme to the joint extraction models. It combines relation features and entity features into joint labels. The compositions of a label are three pieces of information: the position of an entity where a word is located, the type of relation that a word may participate in, and whether a relation where the word is located is a head entity or a tail entity. This means that the entire task can be finally transformed into a sequence labeling task as long as the joint labels are recognized. The model structure of Novel Tagging is an end-to-end model, consisting of an encoder Bi-LSTM and a decoder LSTM. However, Novel Tagging cannot identify the problem of overlapping relations since it binds a pair of entities to a relation.

The TPLinker model constructs a tag about the entire sequence for each word through a handshaking tagging scheme. This kind of tagging marking words that form the boundary of an entity with current word in the sequence, the words that form the beginning of the entity in the same relation with current word, and the words that form the end of the entity in the same relation with current word. All entities and relations related to current word can be identified via tagging of this sequence. Therefore, TPLinker can recognize both SEO and EPO issues. The goal of TPLinker model is to identify the link of each token pair in three types: 1) Entity head to entity tail (EH-to-ET). This link type indicates that the two tokens form a head word and a tail word of the same entity. 2) Subject head to object head (SH-to-OH). This type indicates that two tokens are the start token of a paired subject entity and object entity. 3) Subject tail to object tail (ST-to-OT). Similar to that of subject head to object head, this type indicates the end word of a paired subject entity and object entity. In order to find out all links, TPLinker construct three sequences for each token in a sentence, namely EH-to-ET sequence, SH-to-OH sequence and ST-to-OT sequence. Among these three sequences, all corresponding links with current word are recognized. For convenience of tensor calculation, TPLinker concats SH-to-OH sequences of all tokens in a sentence, as ST-to-OT sequences. Without recognizing entity types, all words in a sentence share the same EH-to-ET sequence. To address the problem of overlapping relation, TPLinker constructs a ST-to-OT sequence and a SH-to-OH sequence for each relation. Thus, TPLinker constructs $2N + 1$ sequences for a sentence if there are N relation types.

However, the above both models obliterate entity features and thus they cannot identify entity types. To address this problem, the TPLinker-plus model adds a module for entity type recognition without changing the original tagging scheme.

4 Results and Analysis

4.1 Datasets

Three publicly available data sets are used including NYT, CoNLL04, and N2C2 2018 (Track2). The NYT dataset was released by Riedel et al. [25] in 2010 containing texts derived from New York Times. Named entities were annotated using the Stanford NER tool and combined with the Freebase knowledge base. The relations between named entity pairs were obtained by linking and referring to the relations in the external Freebase knowledge base combined with a remote supervision method. The CoNLL04 dataset was the data with entity and relation recognition corpora [26]. The N2C2 2018 (Track2) dataset was from the shared track 2 of the 2018 National NLP Clinical Challenges competition. It was oriented to medication and adverse drug events extraction in EHR [27]. Table 2 shows the statistical characteristics of the tree datasets. The NYT dataset has 1297 SEO cases and 978 EPO cases, while the CoNLL04 and N2C2 2018 (Track2) datasets contain only a small number of EPO cases. The SEO overlap cases of N2C2 2018 (Track2) accounts for 68.9%. Thus, the N2C2 2018 (Track2) is unbalanced dataset.

Table 2. The statistical characteristics of the three datasets, where SEO denotes "SingleEntityOverlap" and EPO denotes "EntityPairOverlap"

Dataset	#Normal	#SEO	#EPO	Total
NYT	3266	1297	978	5541
CoNLL04	795	1245	2	2042
N2C2 2018 (Track2)	18583	41195	32	59810

4.2 Evaluation

Three widely used indicators including precision (P), recall (R) and F1 score (F1) are used to measure the performance of each model. The calculation of the indicators are shown in Eq. (3), (4), and (5). True Positive (TP) means that the prediction of a sample is positive and it is actually positive. False Positive (FP) refers to the prediction of a sample is positive but it is actual negative. False Negative (FN) refers to the prediction of a sample is negative but it is actual positive. F1 score is a balanced score of precision and recall.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$F_1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{5}$$

The evaluation is based on a strict matching method. Specifically, an entity is considered correct if both the boundary and type of the entity are correct. A relation is considered correct if both the type of the relation and the associating entities are correct.

4.3 Results

Four experiments were conducted to compare the differences of the models. The first experiment compared the performance of each model on the task of named entity recognition. The result is shown in Table 3. Multi-head obtained a F1 score of 0.841 and 0.870 on the CoNLL04 and N2C2 2018 (Track2) datasets, respectively. Two are Better than One model acquired an F1 score of 0.866 on the CoNLL04 dataset, which was higher than that of Multi-head. However, its performance on the N2C2 2018 (Track2) dataset was lower than that of the Multi-head model. The F1 score of TPLinker-plus on the N2C2 2018 (Track2) data was higher than Two are Better than One model but lower than the Multi-head model.

Table 3. The performance comparison of the models on the named entity recognition task

Model	CoNLL04			N2C2 2018 (Track2)		
	P	R	F1	P	R	F1
Multi-head	0.835	0.847	0.841	**0.879**	**0.860**	**0.870**
Two are better than one	**0.860**	**0.871**	**0.866**	0.711	0.759	0.734
Novel tagging	/	/	/	/	/	/
TPLinker-plus	0.662	0.603	0.631	0.799	0.845	0.821

The second experiment compared the performance of each model on the relation extraction task. The result is shown in Table 4. The Multi-head model obtained an F1 score of 0.738 on the N2C2 2018 (Track2) dataset, which was higher than that of the other three models. Two are Better than One obtained an F1 score of 0.690 on the CoNLL04 dataset, which was higher than Multi-head. TPLinker achieved 0.820 and 0.488 F1 values on NYT and N2C2 2018 (Track2) datasets, respectively, which were higher than that of the Novel Tagging model.

Table 4. The performance comparison of the models on the relation extraction task

Model	NYT			CoNLL04			N2C2 2018(Track2)		
	P	R	F1	P	R	F1	P	R	F1
Multi-head	/	/	/	0.679	0.561	0.614	**0.768**	0.710	**0.738**
Two are better than one	/	/	/	**0.750**	**0.639**	**0.690**	0.551	0.544	0.547
Novel tagging	0.451	0.259	0.329	/	/	/	0.412	0.315	0.357
TPLinker	**0.849**	**0.793**	**0.820**	/	/	/	0.348	**0.814**	0.488

The third experiment compared the recognition performance of Multi-head and Two are Better than One models on different entity types on the N2C2 2018 (Track2) dataset. The result is shown in Fig. 2. Among them, the entity types with poor recognition performance were ADE (Adverse drug event), Duration, and Reason. The recognition performance of ADE entities was the worst, and the F1 scores by the two models were less than 40%. The recognition performance of Multi-head model for all entity types was higher than that of the Two are Better than One model.

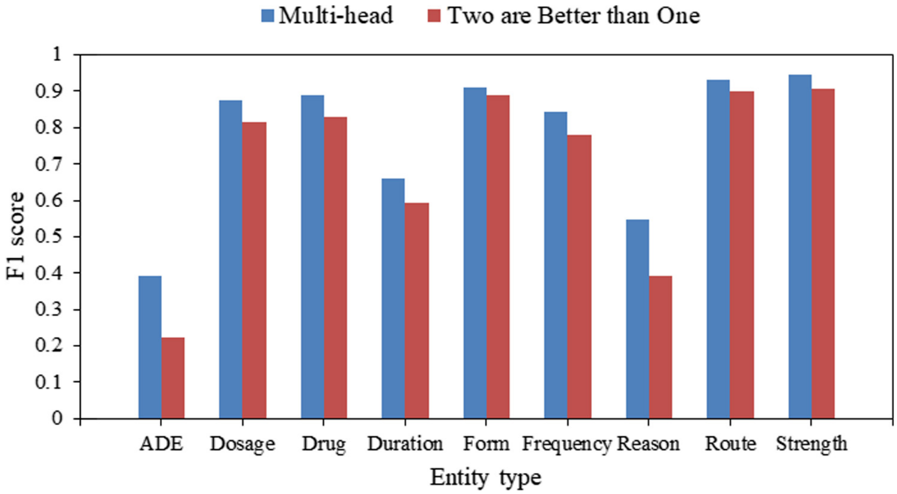

Fig. 2. The recognition performance of two models for different entity types on the N2C2 2018 (Track2) dataset

The fourth experiment compared the extraction of the Multi-head and Two are Better than One models on the N2C2 2018 (Track2) dataset for different relation types. Among them, the relation types with low extraction performance are ADE-Drug, Duration-Drug, and Reason-Drug. The two models both achieved F1 scores lower than 30% on the extraction of the ADE-Drug relation. For the extraction of the Duration-Drug relation, the F1 scores obtained by the Two are Better than One model was nearly 40% lower than that of the Multi-head model, reflecting their significant performance difference. The result is reported as Fig. 3.

4.4 Analysis

In the first experiment, the F1 scores of the Multi-head and TPLinker-plus models were higher on the N2C2 2018 (Track2) dataset than that on the CoNLL04 dataset. It can be seen that for these two models, increasing the amount of data can effectively improve the performance of the named entity recognition task, while both models show strong robustness to sentence length on the task of named entity recognition. In particular, TPLinker-plus improved the F1 score by nearly 20%, and it is speculated that the possible reason is that a large number of sentences without entities in the N2C2 2018 (Track2) dataset acts as negative samples.

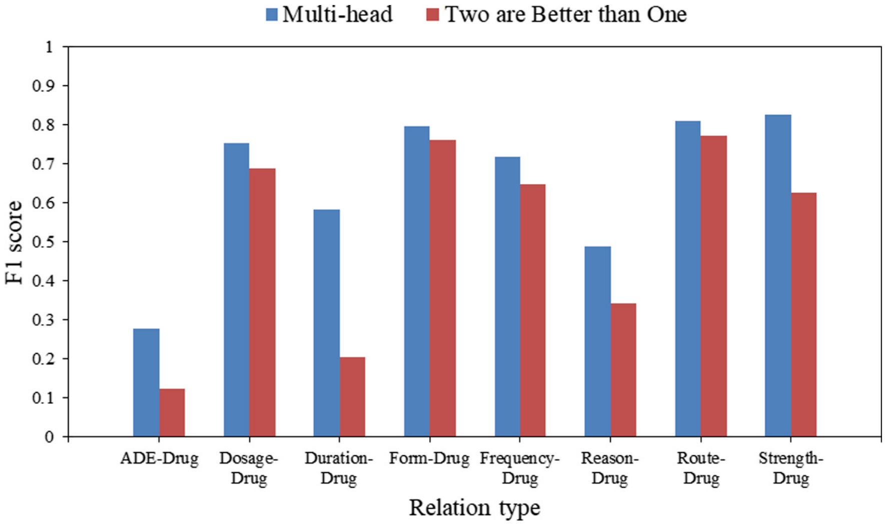

Fig. 3. The extraction performance of the two models for different relation types on the N2C2 2018 (Track2) dataset

In the second experiment, the models that could identify overlapping relations had higher F1 scores than that of the Novel Tagging model, indicating the effectiveness of overlapping relation identification. In the comparison of models with different feature processing strategies, the F1 scores of the Multi-head and Two are Better than One models were higher than those of Novel Tagging and TPLinker models, indicating that the models with feature separation strategies had better performance than that with feature fusion strategy in the task of overlapping relation identification. The interaction information of entity features in the Two are Better than One model had a positive effect on the relation extraction task.

The Multi-head model assumed that any entity could be a head entity of a certain relation for solving the SEO issue. In the experiment, the Multi-head model performed better on the N2C2 2018 (Track2) dataset containing a large number of SEO cases. On the more balanced CoNLL04 dataset, Two are Better than One model with two-direction feature interaction made better use of interaction information than the Multi-head model with one-direction interaction, thus achieving better performance on both tasks. The overall sentence length of the N2C2 2018 (Track2) dataset was longer than that of the CoNLL04 dataset. Two are Better than One model used table feature for entity relation extraction, while the size of table was proportional to the quadratic of the length of sentences. Therefore, the longer the sentences were, the larger the tables were. Longer sentences resulted in a more serious long-term dependency problem when Two are Better than One model traversed tables, which decreased the performance of the model on the N2C2 2018 (Track2) dataset.

In the third and fourth experiments, the entities in ADE type in the N2C2 2018 (Track2) dataset contained complex medical information, and the ADE and Reason entity types had very similar representations. Both were easily confused in the recognition task, resulting in lower recognition performance of both ADE and Reason entities than the others. Improving the recognition performance of ADE and Reason entities had positive implications for the models on the N2C2 2018 (Track2) dataset. In addition, the eight relation types in the dataset were all related to Drug entities, and this specificity leaded to more serious SEO issue. However, both the Multi-head and Two are Better than One model could identify the SEO issue, and most of the relation types were extracted correctly. In the relation extraction task, ADE-Drug, Duration-Drug, and Reason-Drug were the relation types with poor extraction performance. This mainly caused by gradient error propagation problem due to the poor recognition performance of ADE, Duration, and Reason entities.

The performance of the TPLinker model on the N2C2 2018 (Track2) dataset was lower than that on the NYT dataset. It was speculated that this might be caused by the different distributions of relations between the two datasets. After slicing the N2C2 2018 (Track2) dataset into sentences, there were a large number of sentences without relations, and a large number of relations were concentrated on certain sentences. The tagging of the TPLinker model was based on sentence-level annotation. Thus, the exits of different of relations in the sentences had an impact on the performance of the models. To analyze the effect of the number of relations in one sentence to the performance of the TPLinker model, an additional comparison was conducted on the N2C2 2018 (Track2) dataset and the NYT dataset. The two datasets were sliced into testsets containing different numbers of relations in the same sentences for performance comparison. The NYT dataset contained much fewer sentences with multiple relations, thus five relations in the same sentence was utilized, while ten relations in the same sentences were utilized for the N2C2 2018 (Track2). The comparison result is shown in Fig. 4.

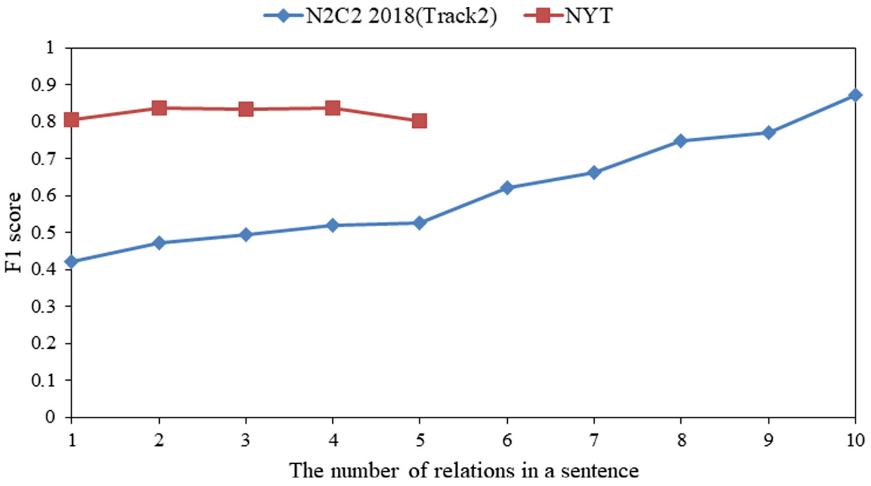

Fig. 4. The performance of TPLinker model for different number of relations in a sentence

For the N2C2 2018 (Track2) dataset, the performance was improving with the increasing of the number of relations. For the NYT dataset, the performance of the

TPLinker model tended to be stable. It indicated that the TPLinker model had advantage on more relations in the same sentences. This showed that the tagging in the model by marking words into sequence could better represent relation information. The sentences with multiple relations in the N2C2 2018 (Track2) dataset were generally long. Different sentence split methods would dramatically affect the sliced sentences thus affecting the performance of the TPLinker model. Thus, appropriate sentence split methods should be paid attention to ensure the performance of the model on long sentences with multiple relations.

5 Conclusions

This paper implemented the latest four joint extraction models and compared the structure and performance differences of the models. Particularly, the performance the four models were evaluated in identifying overlapping relations on three standard datasets. The models with feature separation strategy had better performance than that with feature fusion strategy in the overlapping relation identification task. This indicated the interactions between entity and relation features had positive affect for the joint models. The number of relations in the same sentences had an impact on model performance, thus appropriate sentence split methods were necessary for tagging-based joint extraction models.

Acknowledgements. This work was supported by National Natural Science Foundation of China (No. 61772146) and Natural Science Foundation of Guangdong Province (2021A1515011339).

References

1. Liu, L., Wang, D.: A survey of named entity recognition. Inf. Sci. **37**(3), 329–340 (2018)
2. Xu, K., Feng, Y., Huang, S., Zhao, D.: Semantic relation classification via convolutional neural networks. Comput. Sci. **71**(7), 132–137 (2015)
3. Qi, L., Ji, H.: Incremental joint extraction of entity mentions and relations. In: The 52th Annual Meeting of the Association of Computational Linguistics, pp. 402–412 (2014)
4. Dai, D., Xiao, X., Lyu, Y., Dou, S., Wang, H.: Joint extraction of entities and overlapping relations using position-attentive sequence labeling. In: The AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 6300–6308 (2019)
5. Tan, Z., Zhao, X., Wang, W., Xiao, W.: Jointly extracting multiple triplets with multilayer translation constraints. In: The AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 7080–7087 (2019)
6. Miwa, M., Bansal, M.: End-to-end relation extraction using LSTMs on sequences and tree structures. In: The 54th Annual Meeting of the Association of Computational Linguistics, pp. 1105–1116 (2016)
7. Zheng, S., Wang, F., Bao, H., Zhou, P., Xu, B.: Joint extraction of entities and relations based on a novel tagging scheme. In: The 55th Annual Meeting of the Association of Computational Linguistics, pp. 1227–1236 (2017)
8. Zeng, X., Zeng, D., He, S., Kang, L., Zhao, J.: Extracting relational facts by an end-to-end neural model with copy mechanism. In: The 56th Annual Meeting of the Association of Computational Linguistics, pp. 506–514 (2018)

9. Giannis, B., Johannes, D., Thomas, D., Chris, D.: Joint entity recognition and relation extraction as a multi-head selection problem. Expert Syst. Appl. **114**(Dec), 34–45 (2018)
10. Wang, J., Lu, W.: Two are better than one: Joint entity and relation extraction with table-sequence encoders. In: The 2020 Conference on Empirical Methods in Natural Language Processing, pp. 1706–1721 (2020)
11. Wang,Y., Yu, B., Zhang, Y., Liu, T., Sun, L.: TPLinker: single-stage joint extraction of entities and relations through token pair linking. In: The 28th International Conference on Computational Linguistics, pp. 1572–1582 (2020)
12. Hoffmann, R., Zhang, C., Xiao, L., Zettlemoyer, L. S., Weld, D. S.: Knowledge-based weak supervision for information extraction of overlapping relations. In: The 49th Annual Meeting of the Association of Computational Linguistics, pp. 541–550 (2011)
13. Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J.: Relation classification via convolutional deep neural network. In: The 25th International Conference on Computational Linguistics, pp. 2335–2344 (2014)
14. Vu, N.T., Adel, H., Gupta, P., Schütze, H.: Combining recurrent and convolutional neural networks for relation classification. In: The 2016 Conference of the North American Chapter of the Association for Computational Linguistics, pp. 534–539 (2016)
15. Vashishth, S., Joshi, R., Prayaga, S.S., Bhattacharyya, C., Talukdar, P.: RESIDE: improving distantly-supervised neural relation extraction using side information. In: The 2018 Conference on Empirical Methods in Natural Language Processing, pp. 1257–1266 (2018)
16. Nayak, T., Ng, H.T.: Effective attention modeling for neural relation extraction. In: The 23rd Conference on Computational Natural Language Learning (CoNLL), pp. 603–612 (2019)
17. Zhao, T., Yan, Z., Cao, Y., Li, Z.: Entity relative position representation based multi-head selection for joint entity and relation extraction. In: Sun, M., Li, S., Zhang, Y., Liu, Y., He, S., Rao, G. (eds.) CCL 2020. LNCS (LNAI), vol. 12522, pp. 184–198. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-63031-7_14
18. Zhao, S., Hu, M., Cai, Z., Liu, F.: Modeling dense cross-modal interactions for joint entity-relation extraction. In: The 29th International Joint Conference on Artificial Intelligence, pp. 4032–4038 (2020)
19. Nayak, T., Ng, H.T.: Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In: The AAAI Conference on Artificial Intelligence, vol. 34, no. 05, pp. 8528–8535 (2020)
20. Zhang, R.H., Liu, Q., Fan, A.X., Ji, H., Kurohashi, S.: Minimize exposure bias of seq2seq models in joint entity and relation extraction. In: The Association for Computational Linguistics: EMNLP 2020, pp. 236–246 (2020)
21. Sui, D., Chen, Y., Liu, K., Zhao, J., Liu, S.: Joint entity and relation extraction with set prediction networks. CoRR arXiv:2011.01675 (2020)
22. Gupta, P., Schütze, H., Andrassy, B.: Table filling multi-task recurrent neural network for joint entity and relation extraction. In: The 26th International Conference on Computational Linguistics (COLING2016), pp. 2537–2547 (2016)
23. Ma, Y., Hiraoka, T., Okazaki, N.: Named entity recognition and relation extraction using enhanced table filling by Contextualized Representations. CoRR arXiv:2010.07522 (2020)
24. Zhong, Z., Chen, D.: A frustratingly easy approach for joint entity and relation extraction. CoRR arXiv:2010.12812 (2020)
25. Riedel, S., Yao, L., McCallum, A.: Modeling relations and their mentions without labeled text. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010. LNCS (LNAI), vol. 6323, pp. 148–163. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15939-8_10

26. Roth, D., Yih, W.: A linear programming formulation for global inference in natural language tasks. In: The 8th Conference on Computational Natural Language Learning (CoNLL), pp. 1–4 (2010)
27. Sam, H., Kevin, B., Michele, F., Amber, S., Ozlem, U.: 2018 n2c2 shared task on adverse drug events and medication extraction in electronic health records. J. Am. Med. Inform. Assoc. **27**(1), 3–12 (2020)