

## Research Article

# Joint Extraction of Long-Distance Entity Relation by Aggregating Local- and Semantic-Dependent Features

Xiao Wei  and Yongqi Chen 

*School of Computer Engineering and Science, Shanghai University, Shanghai, China*

Correspondence should be addressed to Xiao Wei; [xwei@shu.edu.cn](mailto:xwei@shu.edu.cn)

Received 20 January 2022; Accepted 18 February 2022; Published 16 March 2022

Academic Editor: Deepak Kumar Jain

Copyright © 2022 Xiao Wei and Yongqi Chen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The relations among long-distance entities are important for text understanding. Long-distance entities mean a pair of entities separated by more words in the text. Most current work of the joint extraction of entity relation is between entities that are close together. These methods are not suitable for long-distance entity relation extraction for they usually ignore the interaction between long-distance entities. In this paper, we propose Long-Distance Feature Aggregation Model (LDFAM), an end-to-end long-distance entity relation joint extraction model, which uses a Bi-GRU and weighted graph neural network to model text and learn its local- and semantic-dependent features. The long-distance dependence between entities is obtained by aggregating these two types of features. For entity recognition, we use conditional random fields to further obtain the dependency information between tags. For relation extraction, we fuse entity tag information for biaffine pairwise scoring. Experimental results on NYT and SemEval2010-Task8 datasets show that our model has achieved significant improvements compared to the baseline model. Finally, we apply the model to the field of material science to extract entity relationship knowledge from material literature. Experimental results show that benefiting from the long-distance entity relation extraction, the proposed method can get more useful knowledge from material literature and improve the knowledge coverage of material domain knowledge acquisition.

## 1. Introduction

Entity and relation extraction is an important research topic in the field of information extraction and natural language processing. The traditional method is mainly aimed at the short-distance entity relation extraction within a sentence, and this task is regarded as a pipeline of two independent tasks, named entity recognition [1] and relation extraction [2]. In fact, in order to better understand the text, the long-distance entity relation is just as important as the short-distance entity relation.

The long-distance entities may be in the same sentence or in different sentences. In either case, the text is generally a long text. For example, as shown in Figure 1, there is a “/location/location/contains” relation between “Florida” and “Miami.”

We can see that the head entity “Florida” is at the end of the sentence, and the tail entity “Miami” is at the beginning

of the sentence, and they are far apart. The pipeline method [3, 4] usually first identifies the entity mentions “Florida” and “Miami” and then judges the relation between “Florida” and “Miami.” These methods ignore interaction between entity recognition and relation extraction tasks. Joint extraction models [5–7] are proposed to establish the association between two tasks. Although these methods reflect the advantages of joint modelling, these are all feature-based systems and therefore rely heavily on feature engineering.

With the successful development of deep neural networks, more and more models automatically obtain the semantic features of texts through neural networks to realize entity recognition and relation extraction. These methods use CNN, LSTM, or Tree-LSTM to model text to obtain local semantic features [8, 9], global sequence features [10], and syntactic dependency features [11, 12].

However, these methods cannot well capture the long-distance dependence between entities in a long text. When

*But Joe Garcia, a Democratic political strategist in Miami, said that even if the race was only slightly competitive, national Democratic leaders would come to campaign for Mr. Davis because winning the governor's seat would help them capture Florida's 27 electoral votes in the 2008 presidential race.*

FIGURE 1: Example: there is a “/location/location/contains” relation between “Florida” and “Miami.”

the distance between entities gradually increases, their performance will decrease significantly. The reasons are as follows: first, it is difficult to obtain complete text semantic information using only a single neural network on long text, which will result in the lack of partial semantic information. For example, using CNN can better capture local semantic information, but it is difficult to obtain global information. Using RNN can obtain sequence semantic features but ignores the syntax and semantic features. Second, their methods can cause the propagation of errors. The identified entities may be wrong, but they are directly used as the input of relation extraction, which leads to errors in relation extraction. And the appearance of redundant wrong entities will lead to the decline of recall rate.

In this paper, we propose Long-Distance Feature Aggregation Model (LDFAM), an end-to-end model to extract long-distance entity relation. First, the word embedding representation is automatically obtained through BERT, and the global sequence feature and the syntactic dependence feature of the text are captured by Bi-GRU and graph neural network, respectively. Then, these two kinds of features are aggregated to obtain the long-distance dependence between entities. Finally, CRF is used to further obtain the dependency rules between entity mention tags and perform entity mention identification and then predict the entity relation through biaffine paired scoring. Our contribution has three parts:

- (i) We integrate the global semantic information of the text and the long-distance grammatical dependence feature to generate the long-distance entity dependence feature
- (ii) We map the initial entity mentions through two independent potential spaces and aggregate similar entity mentions into entity representations, avoiding data redundancy problems caused by pairs of similar entity mentions
- (iii) We evaluate the method on two public relation extraction datasets NYT and SemEval2010-Task8, and we apply the model to the field of materials to extract entity relation in material literature. Experimental results show that our method significantly improves the performance of joint extraction of long-distance entity relation and achieves a new state-of-art on both datasets

## 2. Related Work

**2.1. Extraction of Entity Relation.** The early joint extraction methods of entity relations are all based on feature engineering. Collins [5] proposed an incremental joint framework that uses structure perceptron [13, 14] and effective cluster

search to extract mentioned entities and relations at the same time. Unlike traditional label-based methods, they use a segmented decoder based on the idea of a semi-Markov chain. Miwa and Sasaki [6] proposed a structured learning method based on history, which maps the entity relation extraction task to a simple form filling problem by integrating global features, appropriate learning methods, and search order. These methods require the design of complex features and rely heavily on NLP tools, which may lead to error propagation.

Most of the current methods use deep neural networks to achieve joint extraction. Miwa and Bansal [15] extract entity relation based on word sequence information and dependency tree structure information by using bidirectional LSTM and Tree-LSTM, and the dependency layer focuses on the shortest path of the two entities in the dependency tree. The shortest path has been proved to be very effective in the research of relation classification. However, the long dependency of the label is ignored when predicting the entity. The LSTM decoder [16] is optimized to solve the long dependency problem of tags. Zheng et al. [10] provide a new idea of joint extraction, which converts the extraction problem into a sequence labelling problem. The label of the entity object is composed of three parts: word position, relation type, and relation role. The correlation of the entity label is enhanced through the LSTM decoding layer with bias objective function. However, using the nearest combination method when merging the tags will cause some problems. Based on this tagging strategy, an attention mechanism [17] is introduced into the model to enhance the ability to encode relational words in the text. Adversarial training is adopted in the model training process.

The above methods all use a single neural network to model the text, which makes it difficult to obtain complete semantic information, resulting in the lack of semantic information. This problem becomes more serious when modelling long texts. The limitation of the model itself makes the model's performance on long texts not as good as on short texts. For example, the RNN network has gradient disappearance. This makes it more difficult to obtain the full-text semantics.

**2.2. Long-Distance Relation Extraction.** Cross-sentence relation extraction is the research that has only been put forward in recent years. Zhu et al. [18] construct a fully connected graph for the entities in the text and use the features of GNN with generated parameters to automatically learn edges to realize multihop relation inference. This alleviates the long-distance dependence problem of entity relation extraction to a certain extent. Christopoulou et al. [19] use heuristic rules to create different types of edges for different nodes, thereby using different types of nodes and edges to create document-level graphs. The reasoning mechanism at the edge of the graph uses multi-instance learning to learn

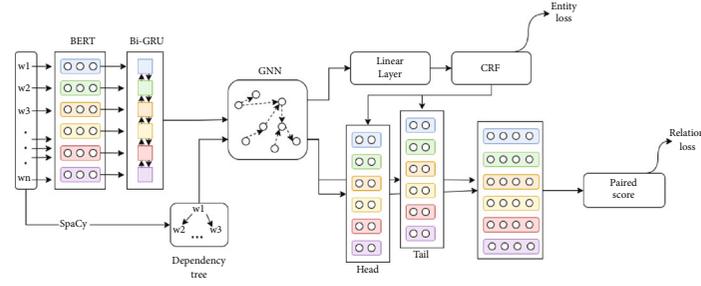


FIGURE 2: The overall architecture of the proposed joint extraction model of long-distance entity relation.

the relations within and between sentences. Wang et al. [20] encode both the overall and local information of the document and then concatenate the local semantic and global semantic representation of the entity to obtain the representation of the entity pair, which is combined with the document theme to classify relation. Nan et al. [21] regards the graph structure as a latent variable. Based on structural attention, a variant of the matrix tree theorem is used to generate a specific-dependent structure to capture the nonlocal interactive information between entities. They also proposed an iterative strategy to dynamically construct a latent structure based on the previous iteration which can improve the aggregation of information in the document and better perform multihop reasoning.

Most of these methods all use graph neural network to achieve cross-sentence relation extraction. They pay more attention to relation extraction, such as multihop relation inference, but ignore the importance of entity information for relation extraction. Making full use of entity semantic information can effectively improve the performance of relation extraction.

### 3. Model

The Figure 2 illustrates the overall architecture of the proposed joint extraction model of long-distance entity relation. Firstly, we use BERT [22] to represent the initial embedding of the text. With the initial embedding representation of words, we use Bi-GRU and weighted GNN to extract global sequence- and local-dependent features, respectively, then further obtain the dependent features of the entity tag through CRF, then map each word to two different vector spaces and predict the relation of each word pair.

**3.1. Encoding.** We use BERT as the initial encoder. After BERT was recently proposed, it has achieved excellent performance in all directions in the NLP field. BERT uses transformer as the main framework of the model, which can better capture the bidirectional semantic relation in sentences. We segment sentences according to words.

$$\text{Text} = \{w_1, w_2, w_3, \dots, w_n\}, \quad (1)$$

where  $w_i$  represents the  $i$ th word in the input text. According to the BERT pretraining language model, the initial embedding representation  $e_i$  of the word is obtained. The input text can be represented by several word vectors.

$$\text{Text}_{\text{embedding}} = \{e_1, e_2, e_3, \dots, e_n\}. \quad (2)$$

**3.2. Feature Acquisition.** As shown in Figure 2, we use Bi-GRU and graph neural network to obtain the global semantic features and local grammatical dependency features of words, respectively. Bi-GRU [23] is a variant of Bi-LSTM [24], which replaces the forget gate and input gate in LSTM with update gates and merges the cell state and the hidden state. Bi-GRU can well obtain the global sequence semantic features in text. Taking the initial text encoding as input, we use Bi-GRU to further encode the text.

$$\text{Feature}_{\text{seq}} = \{h_1^0, h_2^0, h_3^0, \dots, h_n^0\}, \quad (3)$$

where  $h_i^0$  represents the global sequence semantic features of the  $i$ th word.

Graph neural networks [25, 26] were first proposed in Scarselli et al. [27]. The graph neural network can obtain local dependency features very well, and it is also helpful to obtain remote dependency features. Since our task is to extract long-distance entity relation, long text contains rich syntax and semantic information. By acquiring syntactic and semantic features and disseminating them in different nodes, remote semantic features can be acquired, which is more conducive to long text modelling. Since the original input text is a sequence structure, not a graph structure, we use spaCy to grammatically analyze the text and build a grammatical dependency tree. In order to avoid the lack of semantic information of the central node in the propagation of graph neural network, we add reflexive dependency to each node. Using the dependency relation in the dependency tree as the adjacency matrix, we assign different weight values to different dependency relation and then use the weighted graph neural network to extract local dependency features.

$$h_u^l = \sigma \left( \sum_{v \in N(u)} (W^l h_v^l + b^l) \right), \quad (4)$$

where  $h_u^l$  represents the hidden layer feature of the word  $u$  in the  $l$ th layer and  $N(u)$  represents a word that has a dependency relation with  $u$ , and  $\sigma$  represents the activation function.  $W^l$  and  $b^l$ , respectively, represent the weight and bias of layer  $l$ , which are both learnable neural network parameters.

TABLE 1: The specifics of the experimental settings.

	Settings
Bi-GRU (2 layers)	400
GNN (2 layers)	256
Linear layer	256
FFNN (2 layers)	256
Dropout	0.5
Learning rate	0.001
Optimizer	Adam

3.3. *Entity Recognition.* Based on the sequence- and local-dependent semantic features extracted by Bi-GRU and the weighted graph neural network, we use a two-layer linear layer to identify the entity reference of the word and obtain the initial tag score of the word.

$$P_u = \text{softmax}\left(W^{\text{linear}}h_u^l + b^{\text{linear}}\right), \quad (5)$$

where  $P_u$  represents the initial tag score of word  $u$ . Then, use CRF [28] to further obtain the dependency relation between entity mention tags. We learn the transition probability  $A$  between entity mention tags through the CRF layer and calculate the score of each possible tag sequence.

$$s(\text{Text}, y) = \sum_{i=0}^n A_{y_i, y_{i+1}} + \sum_{i=1}^n P_{i, y_i}, \quad (6)$$

where  $y$  represents the label sequence predicted by Text and  $A_{i,j}$  represents the transition probability from tag  $i$  to tag  $j$ .

3.4. *Relation Extraction.* We use entity tag information to guide relation extraction. In order to reduce the error propagation between entity recognition and relationship extraction, we will map entity references into two different latent spaces to generate entity-level representations. First, the hidden layer feature and the mention label feature are spliced.

$$h_u^{l+1} = \left[ h_u^l ; P_u \right]. \quad (7)$$

Then, use two feed-forward neural networks (FFNN) for each word to project into two independent latent spaces, corresponding to the head entity and tail entity of the entity pair, respectively.

$$x_u^{\text{head}} = \sigma\left(W_{\text{head}}h_u^{l+1} + b_{\text{head}}\right), \quad (8)$$

$$x_u^{\text{tail}} = \sigma\left(W_{\text{tail}}h_u^{l+1} + b_{\text{tail}}\right), \quad (9)$$

where  $\sigma$  represents the activation function.  $W_{\text{head}}$  and  $W_{\text{tail}}$  are the parameters of the two FFNNs, respectively, and  $x_u^{\text{head}}$  and  $x_u^{\text{tail}}$  represent the resulted head/tail representations for the word  $u$ .

TABLE 2: Statistics of datasets.

	NYT		SemEval2010-Task8	
	Train	Test	Train	Test
10~20	18016	1336	3022	496
20~30	13776	648	2024	339
30~40	5568	296	1053	159
40~	2640	120	949	174
Total	40000	2400	7048	1168

Then, a classifier calculates the reference-level paired confidence score and obtains the entity-level paired score.

$$\text{score}\left(e^{\text{head}}, e^{\text{tail}}\right) = \log \sum_{i \in E^{\text{head}}, j \in E^{\text{tail}}} \left( W_{\text{rel}} \left[ x_i^{\text{head}} ; x_j^{\text{tail}} \right] + b_{\text{rel}} \right), \quad (10)$$

where  $E^{\text{head}}$  and  $E^{\text{tail}}$ , respectively, represent the set of mentions of entities  $e^{\text{head}}$  and  $e^{\text{tail}}$ .

3.5. *Train Details.* We use two kinds of loss in the model: entity loss and relation loss. For the entity loss  $\text{loss}_e$ , we use (PAD, CLS, Begin, Inside, Out, X) tagging for the ground-truth labels, and the label of each word belongs to one of the five classes. We use the cross-entropy loss function in training. For the relation loss  $\text{loss}_r$ , we use the one-hot relation vector as the ground truth label of each entity pair. Similarly, we use the cross-entropy loss function in training. In addition, we set a trainable relation weight in the experiment to learn the interaction between entity and relation. Finally, the total loss is calculated as the sum of the entity loss and the relation loss:

$$\text{loss}_{\text{total}} = \text{loss}_e + \alpha \text{loss}_r. \quad (11)$$

## 4. Experiments

In this section, we present the experimental results of the proposed model. We first describe the relevant settings of the experiment, the datasets, and the baselines for our comparison. Then, we do a quantitative analysis on the accuracy of the relation extraction between entities at different distances. Through the displayed experimental results, the improvement of the proposed model on the joint extraction of long-distance entity relations is verified.

4.1. *Experimental Settings.* In our implementation, we first split the text into words. Each word has three values: inputID, mask, segment, where inputID represents the number of the word in the BERT dictionary, mask represents which sentence the word belongs to, and segment represents which sentence the word belongs to. The word embedding representation (800 d) of each word is generated through the BERT pretraining language model and used as the input embedding of each word. For the construction of the text dependency tree, we use the spaCy grammatical analysis tool

TABLE 3: Results for NYT.

	NovelTagging			MultiDecoder			Our model		
	<i>P</i>	<i>R</i>	<i>F1</i>	<i>P</i>	<i>R</i>	<i>F1</i>	<i>P</i>	<i>R</i>	<i>F1</i>
10~20	0.599	0.453	0.516	0.645	0.492	0.558	<b>0.731</b>	<b>0.648</b>	<b>0.687</b>
20~30	0.408	0.238	0.301	0.585	0.329	0.421	<b>0.695</b>	<b>0.664</b>	<b>0.679</b>
30~40	0.398	0.201	0.267	0.401	0.286	0.334	<b>0.625</b>	<b>0.517</b>	<b>0.566</b>
40~	0.317	0.25	0.28	0.367	0.263	0.306	<b>0.549</b>	<b>0.434</b>	<b>0.485</b>

TABLE 4: Results for SemEval2010-Task8.

	NovelTagging			MultiDecoder			Our model		
	<i>P</i>	<i>R</i>	<i>F1</i>	<i>P</i>	<i>R</i>	<i>F1</i>	<i>P</i>	<i>R</i>	<i>F1</i>
10~20	0.492	0.217	0.301	0.436	0.392	0.413	0.582	0.426	0.492
20~30	0.401	0.202	0.269	0.426	0.408	0.417	0.511	0.397	0.447
30~40	0.379	0.174	0.239	0.312	0.329	0.320	0.436	0.314	0.365
40~	0.320	0.152	0.206	0.295	0.281	0.288	0.391	0.309	0.345

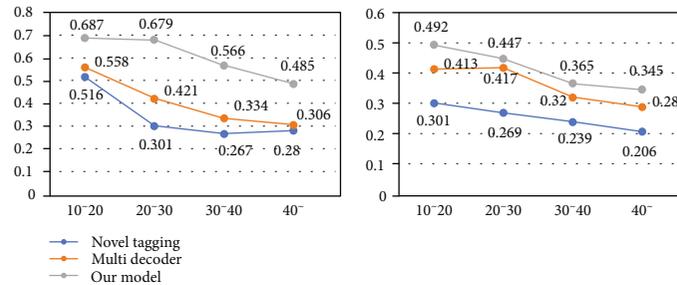


FIGURE 3: Results by different distance.

to obtain the grammatical dependency of the entire text word and construct the dependency matrix.

Since we are targeting distant entities, the text we choose is longer. We use 2-layer Bi-GRU with 400 units to represent the text. Then, we use 2-layer GNN with 256 feature sizes to further model the text. In the entity recognition part, we use a linear layer with a feature size of 256. In the relation prediction part, we also use two two-layer feedforward neural networks with 256 feature sizes. During training, we set the dropout rate to 0.5 and the learning rate (lr) to 0.001. We use the Adam optimizer to train the model and implement it under TensorFlow. More specifics of the experimental settings are provided, as are shown in Table 1.

**4.2. Datasets.** We use NYT and SemEval2010-Task8 datasets to evaluate the proposed model. There are a lot of short texts in NYT and SemEval2010-Task8 datasets. These texts cannot be verified by our experiment. Therefore, we chose those longer texts as the dataset. Due to the different amounts of data in different relation in the datasets, we select 7 relations with a larger amount of data, in order to reduce the interference error of small samples. We divide the datasets into 4 categories according to the distance of the entities: between 10 and 20 (10~20), between 20 and 30 (20~30), between 30 and 40 (30~40), and more than 40 (40~). Between 10 and 20 means the distance between entities differs by 10 to

20 words. Between 20 and 30 means the distance between entities differs by 20 to 30 words. Between 30 and 40 means the distance between entities differs by 30 to 40 words. More than 40 means the distance between entities is greater than 40 words. The statistics of NYT and SemEval2010-Task8 are shown in Table 2. In the detailed analysis, we discussed the different categories of results.

**4.3. Baseline and Evaluation Metrics.** NovelTagging: Zheng et al. [10] proposed a new tagging scheme, which focuses on extracting the triples composed of two entities and the relation between these two entities. By directly modelling the triples, the entity relation joint extraction is converted into a sequence labelling problem, and each word in the sentence is labelled with the entity and relation category. They also make use of the global sequence feature of text, but lack the long-distance syntax dependency feature based on graph. The purpose of comparison with them is to show the importance of using graph to obtain long-distance semantic features.

MultiDecoder: Zeng et al. [29] proposed an advanced method that treats relation extraction as a seq-seq problem and used multiple dynamic decoders to extract duplicate relational triples to solve the problem of repeated combination of entity references. The purpose of comparison with them is to show that it is superior to using multiple dynamic

TABLE 5: Datasets of materials.

Dataset	Train	Test	Valid
Quantity	701	101	90

TABLE 6: Results for material dataset.

	$P$	$R$	$F1$
Bi-GRU (2 layers)	0.381	0.390	0.372
GNN (2 layers)	0.781	0.768	0.771
Our model	0.756	0.810	0.778

decoders to aggregate similar entity references for relation extraction. The advantage of their method is that they can aggregate the representation of similar entities and reduce the emergence of redundant entities to a certain extent. However, their method cannot obtain the dependency features between remote entities and cannot well identify the remote entity relations in the text.

We use precision, recall, and standard  $F1$  scores to evaluate the results. If and only if both entities and relation are the same as the ground truth, the predicted triplet is considered correct.

**4.4. Quantitative Results.** Table 3 shows the precision, recall, and  $F1$  scores of NovelTagging, MultiDecoder, and our model on the NYT dataset. On 10~20, our model outperforms NovelTagging by 0.171 and MultiDecoder by 0.129. The distance is still relatively close between 10 and 20, so the improvement in this range is not as good as the long distance. With the increase of the distance between entities, our model is still greatly improved compared with the baseline model. When the distance is greater than 40, it appears too large, and the performance will decrease to a certain extent at this time.

Similar results can be found on the SemEval2010-Task8 dataset. Table 4 shows the precision, recall, and  $F1$  scores of NovelTagging, MultiDecoder, and our model on the SemEval2010-Task8 dataset. There is a significant improvement on all four divided datasets with different distances.

We summarize the results of entity-to-relation predictions at different distances. Figure 3 shows the extraction results of different distances on NYT (left) and SemEval2010-Task8 (right) datasets. From the general trend in Figure 3, we can see that  $F1$  gradually decreases with the increase of the distance between entities. This means that the distance between entities does increase the difficulty of relation extraction. Both NovelTagging and MultiDecoder use bidirectional RNN network to obtain global sequence semantic features in the encoding stage, while MultiDecoder uses multiple dynamic decoders to obtain overlapping relations in decoding. Therefore, NovelTagging and MultiDecoder perform similarly on the single relational dataset NYT. On the dataset SemEval2010-Task8 with overlapping relationship, the performance of MultiDecoder is better than NovelTagging.

Based on the global sequence semantic features, our model further constructs the syntax dependency structure and obtains the long-distance dependency features between entities through graph neural network, so it has better performance on the single relational dataset NYT. Further, we aggregate entity references by mapping them to two different representation spaces and extract relation only after generating entity representations. Therefore, our model also has excellent performance on the overlapping dataset SemEval2010-Task8.

**4.5. Application.** In order to prove the practicability of our model, we apply the model to the field of material science and use it to extract the entity relation knowledge from the material literature. By analysing the material literature from special steel, aluminium matrix composite, and thermal barrier coating, we divide the entities in the field of material science into three types: component entity, process entity, and performance entity. The component entity represents the constituent of the material, and the process entity represents the manufacturing of the material. In the field of materials, we mainly explore the relation between components and properties, process, and properties, so as to judge which components and processes can improve the performance indicators of materials.

Firstly, we construct the dataset of material science based on the patent text and abstract of research paper of the above three material domains. The division of the dataset is shown in Table 5. We train the model on the training set and testing set, so that the model can fully learn the entity and relational semantic features in the material field. And then use the learned semantic features to verify on the verification set to judge the performance of the model. Similarly, we use precision, recall, and  $F1$  scores to measure model performance.

In the selection of comparison models, we choose Bi-GRU model which only considers the global sequence features of entities and GNN model which only considers dependency features. However, our model combines global sequence features and long-distance dependence features. In order to eliminate the influence of the number of network layers on the results, the two-layer structure is adopted in both comparison models.

Table 6 shows the experimental results on the material dataset. Because material dataset aims at a specific domain and has single semantic features and few types of relationships, compared with the public datasets NYT and SemEval2010-Task8, experimental results on the material domain dataset we constructed have higher performance. It can be seen from the data in Table 6 that Bi-GRU network and GNN network only consider one semantic feature and do not integrate the two features. Our model fully integrates these two features, so it has better performance. Further analysis on the whole, by comparing  $F1$  scores, we can find that GNN has a greater impact on long-distance entity relation extraction than Bi-GRU network. This shows that for the task of long-distance entity relation extraction, GNN network can better obtain the features of long-distance dependence, so as to judge the entity relation more accurately. We can also find that the

experimental results have a greater improvement in the recall rate. This is because in the relation extraction part, we aggregate mention into entity representations through two independent latent spaces, which effectively reduces the generation of wrong results, so we can get a higher recall rate and obtain more relation between components and properties and process and properties from the material literature. The experimental results also show that benefiting from the long-distance entity relation extraction, the proposed method can get more useful knowledge from material literature and improve the knowledge coverage of material domain knowledge acquisition.

## 5. Conclusion

In this paper, we present Long-Distance Feature Aggregation Model (LDFAM), an end-to-end long-distance entity relation joint extraction model. We use Bi-GRU and graph neural networks to obtain text sequence features, local features, and grammatical- and semantic-dependent features, respectively. Then, the conditional random field is used to further obtain the transfer dependency between the mention tags, so as to improve the accuracy of entity recognition. In the relation prediction, we combine the mention label information of the words to map each word to two different potential independent spaces, corresponding to the head entity and the tail entity, and finally use the paired scoring function to score the entity-level relation. We evaluated the proposed method on NYT and SemEval2010-Task8 datasets. The results show that the performance of our method is better than the previous methods in the joint extraction of long-distance entity relations. Finally, the proposed method is verified on the dataset in the field of materials. The results show that the use of GNN is more helpful to improve the precision of the model, and the latent mapping space is more helpful to improve the recall rate.

## Data Availability

The data that support the findings of this study are available from the second author upon reasonable.

## Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## Acknowledgments

This work was supported by the National Key Research and Development Program of China (no. 2018YFB0704400).

## References

[1] L. Ratinov and R. Dan, "Design challenges and misconceptions in named entity recognition," in *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL)*, pp. 147–155, Boulder, Colorado, 2009.

[2] D. Zelenko, C. Aone, and A. Richardella, "Kernel methods for relation extraction," *Journal of Machine Learning Research*, no. 3, pp. 1083–1106, 2003.

[3] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Linguisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

[4] G. Zhou, J. Su, J. Zhang, and M. Zhang, "Exploring various knowledge in relation extraction," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pp. 427–434, Ann Arbor, MI, USA, 2005.

[5] L. Qi and H. Ji, "Incremental joint extraction of entity mentions and relations," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 402–412, Baltimore, MA, USA, 2014.

[6] M. Miwa and Y. Sasaki, "Modeling joint entity and relation extraction with table representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1858–1869, Doha, Qatar, 2014.

[7] X. Yu and W. Lam, "Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach," in *Coling 2010: Posters*, pp. 1399–1407, Beijing, China, 2010.

[8] C. Santos, X. Bing, and B. Zhou, "Classifying relations by ranking with convolutional neural networks," *Computer Science*, vol. 86, no. 86, pp. 132–137, 2015.

[9] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in *Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers*, pp. 2335–2344, Dublin, Ireland, 2014.

[10] S. Zheng, F. Wang, H. Bao, Y. Hao, P. Zhou, and B. Xu, "Joint extraction of entities and relations based on a novel tagging scheme," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 1227–1236, Vancouver, Canada, 2017.

[11] J. Li, M. T. Luong, D. Jurafsky, and E. Hovy, "When are tree structures necessary for deep learning of representations," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2304–2314, Lisbon, Portugal, 2015.

[12] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng, and Z. Jin, "Classifying relations via long short-term memory networks along shortest dependency paths," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1785–1794, Lisbon, Portugal, 2015.

[13] M. Collins, "Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1–8, Philadelphia, PA, USA, 2002.

[14] C. Michael and R. Brian, "Incremental parsing with the perceptron algorithm," in *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pp. 111–118, 2004.

[15] M. Miwa and M. Bansal, "End-to-end relation extraction using LSTMs on sequences and tree structures," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1105–1116, Berlin, Germany, 2016.

[16] S. Zheng, Y. Hao, D. Lu et al., "Joint entity and relation extraction based on a hybrid neural network," *Neurocomputing*, vol. 257, pp. 59–66, 2017.

- [17] P. Huang, X. Zhao, Y. Fang, H. Zhu, and W. Xiao, "End-to-end knowledge triplet extraction combined with adversarial training," *Journal of Computer Research and Development*, vol. 56, no. 12, pp. 2536–2548, 2019.
- [18] H. Zhu, Y. Lin, Z. Liu, J. Fu, T. S. Chua, and M. Sun, "Graph neural networks with generated parameters for relation extraction," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1331–1339, Florence, Italy, 2019.
- [19] F. Christopoulou, M. Miwa, and S. Ananiadou, "Connecting the dots: document-level neural relation extraction with edge-oriented graphs," in *Proc. of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4925–4936, Hong Kong, China, 2019.
- [20] D. Wang, W. Hu, E. Cao, and W. Sun, "Global-to-local neural networks for document-level relation extraction," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3711–3721, 2020.
- [21] G. Nan, Z. Guo, I. Sekuli, and W. Lu, "Reasoning with latent structure refinement for document-level relation extraction," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1546–1557, 2020.
- [22] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, Minneapolis, MN, USA, 2019.
- [23] J. Chung, C. Gulcehre, K. H. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Workshop on Deep Learning*, pp. 1–9, Montreal, Canada, 2014.
- [24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [25] D. Marcheggiani and I. Titov, "Encoding sentences with graph convolutional networks for semantic role labeling," in *Proc. of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1506–1515, Copenhagen, Denmark, 2017.
- [26] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. V. Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *European Semantic Web Conference*, pp. 593–607, Springer, Cham, 2018.
- [27] F. Scarselli, M. Gori, Ah Chung Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.
- [28] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: probabilistic models for segmenting and labeling sequence data," in *Proceedings of the 18th International Conf. on Machine Learning*, pp. 282–289, Williamstown, USA, 2001.
- [29] X. Zeng, D. Zeng, S. He, K. Liu, and J. Zhao, "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*, pp. 506–514, Melbourne, Australia, 2018.