Neural Processing Letters
https://doi.org/10.1007/s11063-022-11101-3

Case Element Joint Extraction Based on Case Field Correlation and Dependency Graph Convolutional Network

Yu Zhang[1,2] · Shengxiang Gao[1,2] · Zhengtao Yu[1,2] · Chengding Zhao[1,2] · Peilian Zhao[1,2] · Peifu Han[1,2]

Accepted: 7 December 2022

Abstract
The case element is a brief description of the case-related events. Extracting the case elements in the news text has great significance for downstream case field natural language processing tasks. In view of the case field relevance and intrinsic relevance of the case elements, this paper proposes a joint case element extraction method based on case domain correlation and graph convolutional network: modeling sentence contextual information by bi-directional long short-term memory networks, then using it to predict the case field correlation for guarantying the elements' relevance of cases by joint learning; and modeling the dependency relationship of candidate elements by graph convolutional network to capture its intrinsic relevance. The experiments show that the method proposed in this paper improves accuracy rate by 6. 6% in extracting case elements.

Keywords Case element · Case field correlation · Joint learning · Graph convolutional network

1 Introduction

In recent years, the natural language processing of texts in the legal field has received more and more attention from scholars. The current research object of natural language processing in the legal field is mainly focused on texts in legal documents. How to automatically handle public opinion texts related to cases is an urgent scientific question. For natural language texts in the field of the case, the case element is a brief description of the case-correlated event. This paper defines the case element as five categories of time, place, person, action, and action object when the event occurred. The case elements reflect the core content of

✉ Shengxiang Gao
179958170@qq.com

1 Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, Yunnan, China

2 Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming 650500, Yunnan, China

Published online: 27 December 2022

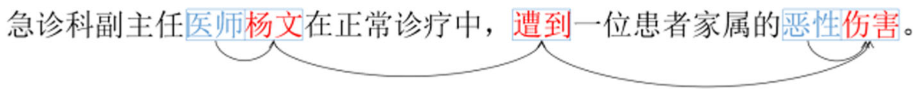

Fig. 1 An example of Dependency in news sentences

the case-correlated event, so it has a guiding and restrictive role in the subsequent natural language processing tasks such as text classification and abstracts in case-correlated fields.

The case element extraction task can be regarded as a special keyword extraction task. Given a news text, several keywords that meet the definition of case elements are extracted. According to the definition of the case element, the case element has two features: the correlation of the case field and the relevance between the case elements. These features are not found in a general keyword. Keyword extraction is a research focus in natural language processing. At present, there are two main methods for keyword extraction: unsupervised method and supervised method. Unsupervised keyword extraction research mainly uses statistical methods to calculate the importance of words in statistical significance [1, 2], but statistical methods cannot use the semantic relationship of context and the semantic information of the words themselves, so the extracted keywords and the sentence where they are located cannot be determined whether is correlated to the case field. For sentences correlated to case field and sentences not correlated to case field, it is clear that it is more likely that case elements are extracted from sentences correlated to the case field. Therefore, whether the sentence in which the element is located is relevant to the case domain will obviously affect whether the element is considered as the case element.

Supervised keyword extraction research usually focuses on machine learning methods, training keyword extraction models through labeled corpora. Some researchers convert the keyword extraction task into a binary classification problem of words [3, 4], and some researchers regard it as a task to generate keywords from texts [5]. Although these methods can take advantage of the characteristics of words and texts, they ignore the relationship between keywords. As shown in Fig. 1, "杨文(Yang Wen)", "遭到(Encountered)" and "伤害(Injury)" are three candidate elements that exist on a dependency tree, which describe the person and action respectively. It is at the same time as whether "杨文", "遭到" and "伤害" are case elements. Based on this feature, this paper converts the case element extraction into a group-level binary classification problem, that is, determines whether a group of candidate elements are case elements.

To solving above problems, this paper proposes a method for joint extraction of case elements based on case domain correlation and dependency graph convolutional network. The bidirectional recurrent neural network is used to model the context of the candidate elements to determine whether it has a case field correlation. The graph convolutional neural network is used to model the dependency between the candidate elements to extract their associated features, and finally determine whether this group of elements is case elements. In order to obtain a large amount of training corpus, this paper adopts the method of distance supervision to obtain labeled case elements in the news through co-occurrence information [6].

The contribution of this paper includes the following two aspects: 1. A case element extraction method with joint learning of case field correlation prediction is proposed. By modeling the sentence context, it is judged whether the candidate elements meet the correlate characteristics of the case field, and the accuracy of the extracted case elements is improved;

2. The idea of extracting candidate elements by group is proposed. The associated features between candidate elements in the group is extracted by using the graph convolutional neural network. The accuracy of the case element extraction is improved by integrating the associated features between the candidate elements in the group.

2 Related Work

Case element extraction can be regarded as a special keyword extraction task, which is mainly reflected in correlation of case elements to case fields and interrelationship between case elements.

There are two main methods for keyword extraction: unsupervised method and supervised method. Unsupervised keyword extraction uses statistical methods to calculate the statistical characteristics of words to get their criticality. Li et al. [1] extracted the keywords in news documents by statistically predicting the term frequency-inverse document frequency (tf-idf) feature of the words. Rose et al. [2] predicted the keyword probability of the word by using the statistical co-occurrence frequency in the corpus to obtain the statistical characteristics of the phrase. Li et al. used TextRank algorithm based on graph sorting to extract keywords [3]. Unsupervised method does not need to label data and has better performance in keyword extraction task. However, it is not applicable for case field due to statistical characteristics of words without semantic information and the uncertainty between extracted keywords and the case field.

In order to extract case semantics, the supervised keyword extraction method can be used. Supervised keyword extraction methods generally train machine learning models through labeled corpora. And there are two different understandings in keyword extraction tasks. One is to perform the binary classification task of converting keyword extraction into words. FRANK et al. trained naive Bayesian classifier through some artificially selected features [3]. Turney has trained a decision tree classifier for keyword dichotomies of words [4]. Some use neural network and conditional random field (CRF) model to mark the sequence of words is essentially based on this idea [7, 8]. And the other understanding of keyword extraction tasks is using the method of sequence generation. Meng et al. integrated the copy mechanism into the sequence-to-sequence model (seq2seq) to extract key phrases [9]. Zeng et al. also used seq2seq model to transform keyword extraction into keyword generation task and extract keywords for legal issues [4]. This method enables the model to learn the semantic information of the sentence through the neural network, but it is difficult to closely combine the semantics of keywords with each other because of the temporal characteristics of cyclic neural network. However, case elements do not appear in the sentence in turn, so this method is not suitable for modeling the correlation semantics among case elements.

In order to capture the relationship between the case elements, we model them in the form of graph structure. In recent years, relevant researches on neural network have been applied to the data of graph structure [10, 11], so that neural network can directly calculate the graph. Bruna et al. [10] first transferred the spectral clustering to deep learning and proposed graph neural network. Kpif et al. applied the convolutional neural network (CNN), which is often used in the image field in deep learning, to the graph structure and proposed the graph convolutional neural network (GCN) [11]. And Niepert et al. connected the nodes in the graph in the spatial domain, constructing the nodes of the hierarchical structure [12]. Graph convolution networks have been widely used in natural language processing tasks [13, 14]. Yao et al. trained a deep text classification model by using both words and documents as node compositions [13], and some scholars have also improved the GCN to get better results on some tasks [15, 16]. Zhang et al. used graph convolutional neural network to model the dependency tree, which significantly improved the accuracy of relational extraction [14]. Therefore, we also use this idea for reference to model the dependency tree through GCN to extract the relationship between candidate elements in this paper.

Dependency tree is an important tool for natural language processing tasks, which can directly connect words that are far away but have similar semantic relationships to build a tree, and model the words in the sentence in turn, resulting in information clutter. Although many scholars use dependency tree to model and learn word relations with good results, dependency tree has certain limitations due to the sentence context information outside the dependency tree. Contextual information outside the dependency tree is helpful for judging whether candidate elements have case field relevance, which is of great significance for case element extraction. Therefore, we incorporate contextual information to determine whether the sentence is related to the case for joint learning while using GCN to model the dependency relationship, so as to improve the model effect.

Large-scale data is necessary but difficult to obtain for neural network training. Since Mintz et al. have been proposing the method of obtaining marked entity-relationship corpus through distance supervision [19], many scholars' studies have also confirmed the effectiveness of distance supervision for annotating relational data [6, 20]. Therefore, we also use the idea of distance supervision to establish training data due to the case elements are related to describe the same event from the definition of case element.

3 Method

In this section, we describe the proposed model for extracting case elements. Because legal documents are normative, this paper extracts case elements from legal documents based on rules to build a knowledge base of case elements, where case elements are stacked in groups. Then it collects the news data based on the case element knowledge base, and labels the case elements in the news text according to the co-occurrence information. At the same time, because the case elements have the case field correlation, the case field relevance label of the sentence can be labeled at the same time. During training the model, the contextual encoding of the sentence is obtained by modeling the entire sentence, which is used to predict whether the sentence has a case field correlation, and the candidate element components are partially modeled by dependency graph convolutional networks to obtain a dependency encoding, which is then combined with the contextual encoding information to predict whether the candidate element group is a group of case elements. Figure 2 shows the core process of case element extraction in this paper.

As shown in Fig. 3, the case element extraction model proposed in this paper contains five layers: pre-processing layer, word embedding layer, context modeling layer, dependency modeling layer and predicting layer. Suppose that we input a sentence of news $D = \{ \mathrm{w}_1 ... \mathrm{w}_\mathrm{n} \}$ in which w means a word in the news document. The pre-processing layer constructs the dependency syntactic tree of the sentence through dependency analysis, and then obtains the core syntactic part of the sentence according to the part-of-speech pruning to obtain the candidate elements of the sentence describes as { w*1...w*m} , and get candidate elements are formed into adjacency matrix A through mutual syntactic relationship. The word embedding layer converts words into word vectors described as { e1...en} through a word vector matrix. The context modeling layer uses bidirectional long short-term memory

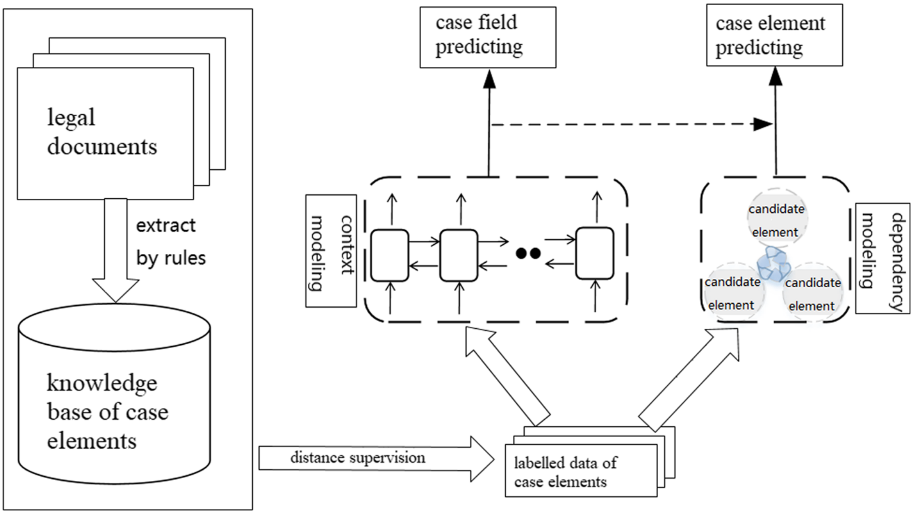

Fig. 2 The core process of case element extraction

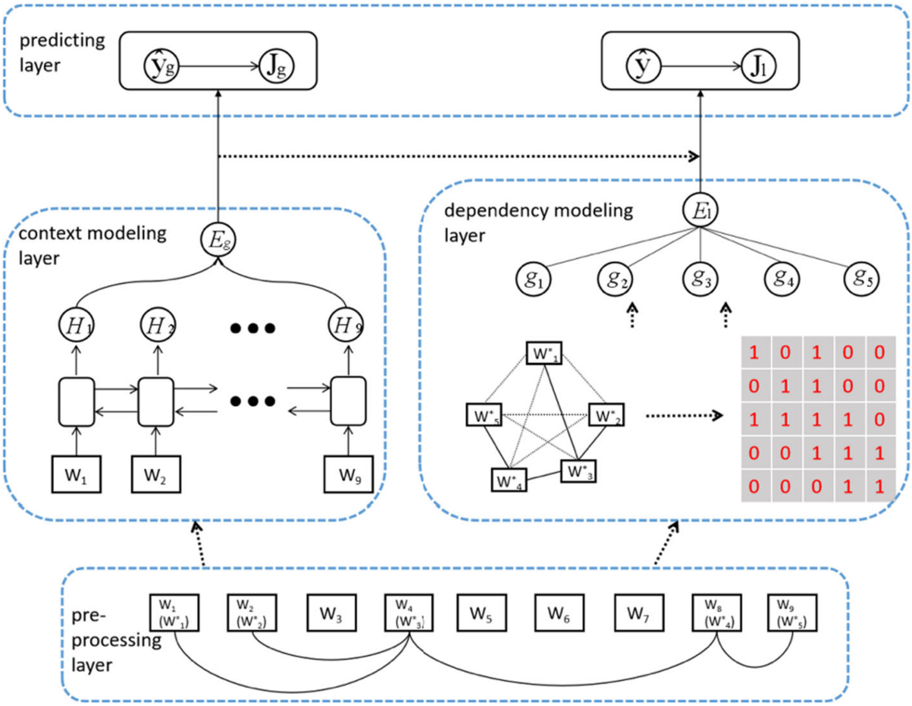

Fig. 3 Model of case element extraction in this paper

network(BiLSTM) to model the entire word sequence to obtain the vector of the sentence E_n. The dependency modeling layer models the adjacency matrix A of the candidate elements to obtain the dependency vector E1. The predicting layer predicts the case field correlation of this sentence through the vector representation E_n of the sentence and predicts whether candidate elements are case elements through E_n and E_1, and calculate the losses separately. Detail is described in this section below.

3.1 Pre-processing Layer

In order to extract the dependent syntactic information for the dependency modeling layer, the pre-processing layer first analyzes the input sentence to its dependency relationship through dependency syntactic analysis, then prunes through the part-of-speech and inter-word dependencies, and intercepts components that have a first-order relationship with subject-predicate object as a subtree. Suppose that the input sentence of the news text is $D = \{ w1...wn\}$, pre-processing layer processes it to get words { wi...wk} in dependency subtree. For ease of description, these words are represented by { w*1...w*m} .

3.2 Word Embedding Layer

The input word sequence $\{w1, w2...wn\} \in R^{V}$ represents the sentence which is to be extracted, V is the size of the vocabulary. We pre-train a word vector matrix $M \in R^{V*d}$ through a large number of referee documents.

$$ei = wi \cdot M \tag{1}$$

(1) can map each word wi into a d-dimensional vector $ei \in R^d$. In the same way, candidate element word w*i in { w*1...w*m} extracted by dependency analysis is also mapped to vector $e{*}i \in R^d$.

3.3 Context Modeling Layer

Natural language texts have temporal characteristics so that recurrent neural networks can effectively model them. In particular, long short-term memory networks(LSTM) has a better result because it can solve the problem of long-distance dependence [21]. In this paper, we use bi-directional long short-term memory networks(BiLSTM) to model the word sequence { w1...wn} of the input sentence. Suppose that wt is the word inputted at moment t, BiLSTM work at time t like the formula as follows:

$$i_t = \sigma(W_i \times w_t + U_i \times h_{t-1} + b_i) \tag{2}$$

$$f_t = \sigma\left(W_f \times w_t + U_f \times h_{t-1} + b_f\right) \tag{3}$$

$$o_t = \sigma(W_o \times w_t + U_o \times h_{t-1} + b_o) \tag{4}$$

$$\widetilde{c}_t = \tanh(W_c \times w_t + b_c) \tag{5}$$

$$c_t = i_t \odot \widetilde{c_t} + f_t \odot c_{t-1} \tag{6}$$

$$h_t = o_t \odot tanh(c_t) \tag{7}$$

$$Ht = [\vec{h}t; \overleftarrow{h}\, t] \tag{8}$$

where it, ft and o_t denote input gate, forget gate and output gate separately, deciding what information the LSTM cell stores, discards, and outputs. W and b and U are trainable parameters, h_{t-1} represents the external state of the last time,$\odot$ denotes vector element product, [;] denotes the concatnating operation. $\vec{h}t$ and $\overleftarrow{h}\, t$ denote the hidden layer tensor of the forward and backward process separately. By processing the word sequence recurrently, BiLSTM encodes the sentence to $H1...H\text{n} \in R^{2q}$ where 2q is the hidden layer dimension of BiLSTM. We concatnate the final results which the forward and backward operations BiLSTM makes respectively as a final vector representation of the sentence, the formula is as follows:

$$E_n = [H_1[: q]; H_n[q :]] \tag{9}$$

where Eg denotes the result of context modeling layer.

3.4 Dependency Modeling Layer

For the dependency subtree { w*1...w*m} built in section A, we treat these words as nodes of the graph, the dependency syntactic relations between words as edges of nodes, to build an undirected graph. To generate the adjacency matrix A of this dependency tree, we set Aij = 1 to indicate that there is an edge between w*i and w*j. Learning from the method of Marcheggian et al. [22], the self-loop of each node is also added to the graph as an edge, which means Aii = 1. In this paper, we use graph convolutional neural networks (GCN) to model the graph. In layer k, GCN calculates the feature $\text{g}_\text{i}^{(k)}$ of node i like the formula as follows:

$$g_\text{i}^{(k)} = RELU\left(\sum_{j=1}^{m} AijW^{(k)}g_j^{(k-1)} + b^{(k)}\right) \tag{10}$$

where W^k and b^k denote the weights matrix and bias respectively for layer k of GCN, $RELU$ denotes a non-liner activation function. h is the input word vector e*i when k = 0.

At the top layer of the GCN, we affine all nodes into a tensor through a layer of fully connected network, the formula is as follows:

$$E_l = tanh(W_l \times G + b_l) \tag{11}$$

where

$$G = [g_1^{(l)}; g_2^{(l)}...; g_m^{(l)}] \tag{12}$$

l denotes the depth of GCN, tanh denotes a non-liner activation function, El is the output of the dependency modeling layer.

3.5 Predicting Layer

The predicting layer is divided into two sub-layers: the case field correlation predicting layer and the case element predicting layer.

3.6 Case Field Correlation Predicting Layer

Context encoding Eg is a semantic space vector representation of the entire input sentence D. In this paper, we give Eg to a softmax classifier to predict whether the sentence is correlated to the case field. The formula is as follows:

$$\widehat{p_g}\left(y_g|D\right) = softmax(W_s \times E_n + b_s) \tag{13}$$

$$\hat{\mathrm{y}}\mathrm{g} = \arg\max_{y}(\hat{p}g(y\mathrm{g}|D)) \tag{14}$$

where ŷg is the predicted result of case field. The training goal is to minimize the cross-entropy loss between the predicted value and the real label, and the loss Jg is formulated as:

$$\mathrm{J}g = -\frac{1}{2}(y_{g-0}\log\hat{y}_{g-0} + y_{g-1}\log\hat{y}_{g-1}) \tag{15}$$

3.7 Case Element PredictingLayer

Both the results of context modeling and dependency modeling are decisive for the predicting of candidate elements. We concatenate Eg and El as input to the case element predicting layer, formulated as:

$$E = [E_n; E_l] \tag{16}$$

Then, we use a softmax classifier to predict whether the candidate element group is a group of case elements. The formula is as follows:

$$\hat{\mathrm{p}}(y|D) = soft\max(We * E + be) \tag{17}$$

$$\hat{\mathrm{y}} = \arg\max_{y}(\hat{p}(y|D)) \tag{18}$$

where ŷ is the predicted result. If ŷ is 1, the model predicts that the candidate element group is a case element group, and components in the case elements are supplemented by a complete dependency tree in the original sentence. The reason to operate supplement in the tree is that the pruning operation may lose some information in the oraginal sentence.

The training goal of the case element predicting layer is also to minimize the cross-entropy loss between the predicted value and the label. The loss is calculated as follows:

$$\mathrm{Jl} = -\frac{1}{2}(y_0\log\hat{y}_0 + y_1\log\hat{y}_1) \tag{19}$$

Finally, we will jointly train the two prediction tasks, that is, the total loss of the model will be added, as follows:

$$J = J1 + Jg \tag{20}$$

4 Experiments

In this section, we introduce the data set, the experimental hyperparameters and the baseline model we used. The effectiveness of the proposed method is verified by contrast experiments, and the effectiveness of the parts of the method is verified by ablation experiments.

4.1 Data Set

Experimental corpus is the basis of training model, which directly determines the accuracy of the experiment. At present, no relevant public standard corpus has been found. In order to verify the effectiveness, two parts of experimental corpus need to be prepared: the corpus of legal documents and the corpus of news texts.

There are many ways of news reporting, but basically the news is uploaded in the form of Internet. Therefore, the carrier of news is web pages, and the basic composition of web pages is HTML language. By analyzing the HTML code of the news website, writing a network crawling program to automatically grab the news text information from the Internet, crawling the news text from the Han Yue news website using the web crawler technology based on XPath, and writing different templates for different websites using the scratch framework. Through the template, you can locate the title, time and time of the news Text and other detailed information, so as to search for accurate and detailed information.

Legal documents are normative, which is very suitable for obtaining the knowledge base needed by the research. Through the processing of large-scale documents collected in the judicial documents network, the judicial documents corpus needed for the research is obtained. Firstly, the structure of the legal document obtained from the judgment document network is the same as that of the relevant documents. Through simple structured processing, it can be expressed in the form of a dictionary, and the content of the legal document can be directly parsed into different components according to the fields. Due to the provisions of the document, the wording of the case description sentence in the legal document is relatively standardized, which is generally the structure of "time, defendant + place + behavior + behavior object". Through the word segmentation and part of speech tagging of the case description sentence, the rules are constructed based on the drafting rules of the legal document, so as to extract the time, place, task Verbs and verb objects are the five elements of the case.

After crawling to the corresponding document data and news data from the judgment document network or news website, this paper selects JSON data format, processes the data in the form of 'key name: key value', stores the corpus with local files, and obtains the local corpus file.

Based on the correlation between case elements, this paper adopts a way of labeling based on distant supervision.

The usual sequence tagging task is carried out manually, because words have different ambiguities based on different contexts. If the corpus is searched according to a word, its meaning in the corpus can not be accurately labeled.

The idea of distant supervision is that when two words with mutual relationship appear together, their meanings will be constrained by each other. The case element annotation process adopted in this paper is as follows:

The first step is to extract the case element knowledge base from the judgment document data. Due to the relevance of case elements, this paper saves the elements of each case in groups.

Second, when dealing with news corpus, if multiple elements of a group of elements in the case element knowledge base appear in a sentence at the same time, we believe that the group of words in this sentence is a group of case elements.

Thus, we can mark the corpus of news text and case elements we need.

By analyzing and cleaning 17,191 documents in the judgment document network through rules, 4311 groups of case elements are obtained and constructed as the case element knowledge base. 3449 news documents are crawled through distant supervision, and the dataset is constructed by sentence.

Finally, the quality of the data set is checked by manual evaluation.

4.2 Hyperparameter Settings

There are five nodes pruned by the dependency tree and replaced by UNK for the components that do not appear in the tree. The dimension of the word embedding layer we set is 300 dimensional, the context modeling layer dropout is set to 0.25, and the dependency modeling layer dropout is set to 0.4, the Adam optimizer is used for optimization, and the batch_size is 20.

4.3 Baseline Models

Due to the few researches on the task of case element extraction in the academic field at present, this paper designs a comparative experiment for the task data set according to the following methods:

(1) Considering the case element extraction task as a Chinese sequence tagging task. There are two main experimental methods: Using the method of combining BiLSTM with CRF (hereinafter referred to as BiLSTM + CRF) and the method of Lattice LSTM (Zhang et al.,2018) (hereinafter referred to as Lattice).
(2) Considering the case element extraction task as a keyword generation task, that is, given a document to generate the case elements. Based on this idea, the seq2seq model based on copy mechanism (Meng et al.,2017)(hereinafter referred to as copyRnn) for sequence generation.

4.4 Evaluation Indicator.

We use the precision rate (hereinafter referred to as p), the recall rate (hereinafter referred to as r), and the F1 value (hereinafter referred to as F) as the evaluation index where F is calculated as:

$$F = 2 * p * r/(p + r) \tag{21}$$

4.5 Efficiency Analysis

By using different baseline models on the proposed data sets, the results are as follows:

The copyRNN only uses the copy mode, that is, it is trained completely by extracting the original text, and stops when decodes five time steps at the decoder.

From Table 1 it can be clearly observed that our method improves the accuracy by 6. 6% and the recall rate by 12. 4%, compared with the case element extraction task which is regarded as the Chinese sequence tagging task. This shows that the idea of using the relationship between candidate elements and the case relevance of the sentence to be extracted is valid for this task.

Table 1 Efficiency Analysis of different methods

Methods	*P*	R	F1_score
Our method	0.8299	0.7907	0.8098
BiLSTM + CRF	0.7637	0.6587	0.7073
Lattice	0.7811	0.6664	0.7192
CopyRNN	0.6284	0.3717	0.4671

When using copyRNN to predict, it's accuracy and recall rate are low, which is because copyRNN is more inclined to produce words other than the original sentence when used for keyword extraction, and the relation between the candidate elements and the case filed correlation of the sentence to be extracted is not used. However, since the case elements in our dataset appear in the original sentence, copyRNN is not applicable to this task.

4.6 Ablation Experiment

In order to verify the validity of each part of the model, each layer is eliminated and compared separately.

When the context modeling layer is not used, the case field correlation predicting is not used at the same time. Ablation experiment without dependency modeling uses BiLSTM instead of GCN to model candidate elements.

By deleting the dependency modeling, the model becomes the BiLSTM method. There is a large gap between the performance of the model and the BiLSTM + CRF model, mainly because the CRF layer can learn the constraints of sentences and add some constraints to ensure that the final prediction result is effective. These constraints can be automatically learned by CRF layer when training data. With these useful constraints, wrong prediction sequences will be greatly reduced.

As can be seen from Table 2, each modeling layer we proposed has a practical effect on case element extraction. When the context modeling layer is not used, the accuracy and recall rate are reduced obviously. Inferring from this, the semantic information modeling of the whole sentence makes the model utilize the effective information other than the dependency tree. The recall rate was slightly reduced by 1. 5% and the accuracy by 6% when the dependencies were modeled without GCN, from which we know that the dependency modeling layer proposed in this paper directly captures the semantic information of the candidate elements, which can obviously improve the accuracy of the model.

Joint learning of the correlation of the case field, at the expense of a small number of recall rates, has improved the accuracy by 8. 7%. It shows that joint learning of the correlation of the case field effectively biased the model towards predicting the case elements in the sentence

Table 2 Result of Ablation Experiment

Methods	*P*	R	F1_score
Our method	0.8299	0.7907	0.8098
Context modeling	0.7219	0.7469	0.7341
Dependency modeling	0.7695	0.7753	0.7723
Case field predicting	0.7426	0.8100	0.7748

associated with the case field as a positive example, which is consistent with our definition of the case elements and has obvious effect.

5 Conclusion

Aiming at the fact that the defined case elements have case-related characteristics and the correlation between case elements, we propose a method for joint prediction of case field correlation while extracting case elements. At the same time, we propose the idea of extracting candidate elements by candidate elements, using graph convolutional neural networks to extract the association relationship of candidate elements in the candidate elements, and improving the accuracy of case element extraction by integrating the characteristics of the association relationship of candidate elements in the candidate elements.

Further research can be carried out by processing the case news text based on the case elements, such as using the case elements as the guidance information to obtain the core information of the case text, and then processing the case field summary of the news text.

Acknowledgements The work was supported by National Natural Science Foundation of China (Grant Nos. 61972186, U21B2027, 61732005), National Key Research and Development Plan (Grant Nos. 2018YFC0830105, 2018YFC0830101, 2018YFC0830100), Yunnan provincial major science and technology special plan projects (Grant No. 202002AD080001-5, 202103AA080015), Yunnan high-tech industry development project (Grant No. 201606, 202105AC160018), Yunnan Basic Research Project (Grant No. 202001AS070014).

References

1. Li J, Zhang K (2017) Keyword extraction based on tf/idf for Chinese news document. J Wuhan Univ J Natl Sci 12(5):917–921
2. Rose S, Engel D, Cramer N et al (2010) Automatic keyword extraction from individual documents. J Text Mining: Appl Theory 1:1–20
3. Frank E, Paynter GW, Witten IH et al (1999) Domain-specific keyphrase extraction. In: Frank A (ed) International joint conference on artificial intelligence. Morgan Kaufmann Publishers, San Francisco, vol 2, pp 668–673
4. Turney PD (2000) Learning algorithms for keyphrase extraction. J Inform Retriev J 2(4):303–336
5. Daojian Z, Guowei T, Yuan D et al (2019) Keyphrase extraction for legal questions based on a sequence to sequence model. J Tsinghua Univ (Sci Technol) 59(4):256–261
6. Clark, Kevin, Manning et al (2016) Deep Reinforcement Learning for Mention-Ranking Coreference Models, A. Clark. Empirical Methods in Natural Language Processing, C. Austin, Texas: Association for Computational Linguistics, 2256–2262 (2016)
7. Waleed Ammar, Matthew Peters, Bhagavatula Chandra, et al (2017) The AI2 system at SemEval-2017 Task 10 (ScienceIE): semi-supervised end-to-end entity and relation extraction. In: Ammar A (ed) The 11th International Workshop on Semantic Evaluations. International Workshop on Semantic Evaluations, Vancouver, pp 592–596
8. Shenhao Jiang, Animesh Prasad, Min-Yen Kan et al (2018) Identifying emergent research trends by key authors and phrases. In: Journal of international conference on computational linguistics, pp 259–269
9. Rui M, Sanqiang Z, Shuguang H et al (2017) Deep keyphrase generation. In: Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, volume 1, pp 582–592
10. Bruna J, Zaremba W, Szlam A et al (2014) Spectral networks and locally connected networks on graphs. In: Bruna A (ed) Journal of international conference on learning representations. International Conference on Learning Representations, Banff
11. Kipf T, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: Thomas A, Kipf N (eds) International conference on learning representations. International Conference on Learning Representations, Toulon

12. Mathias Niepert, Mohamed Ahmed, Konstantin Kutzkov. Learning Convolutional Neural Networks for Graphs, A. Niepert. International Conference on Machine Learning, C. New York City, NY: ACM, 2014–2023 (2016)
13. Yao L, Mao C, Luo Y, et al. Graph convolutional networks for text classification, A. Yao L. AAAI Conference on Artificial Intelligence, C. Honolulu, Hawaii: AAAI, 33(01): 7370–7377 (2019)
14. Zhang Y. , Qi P. , Manning C. D. . Graph convolution over pruned dependency trees improves relation extraction, A. Zhang Y. Empirical Methods in Natural Language Processing, C. Brussels, Belgium: Association for Computational Linguistics, 2205–2215 (2018)
15. Veličković P, Cucurull G, Casanova A, et al. Graph attention networks, A. Veličković P. International Conference on Learning Representations, C. Vancouver, BC: International Conference on Learning Representations (2018)
16. Guo Z, Zhang Y, Lu W, et al. Attention Guided Graph Convolutional Networks for Relation Extraction, A. Guo Z. meeting of the association for computational linguistics, C. Florence, Italy: Association for Computational Linguistics, 241–251 (2019)
17. Xu K, Feng Y, Huang S, et al. Semantic relation classification via convolutional neural networks with simple negative sampling, A. Xu K. Empirical Methods in Natural Language Processing, C. Lisbon, Portugal: Association forComputational Linguistics, 536–540 (2015)
18. Peng N, Poon H, Quirk C et al (2017) Cross-sentence n-ary relation extraction with graph lstms. J Trans Assoc Comput Linguist 5(1):101–115
19. Mintz M, Bills S, Snow R, et al. Distant supervision for relation extraction without labeled data, A. Mintz M. International Joint Conference on Natural Language Processing, C. Singapore: Association for Computational Linguistics, 1003–1011 (2009)
20. Zeng D, Liu K, Chen Y et al (2015) Distant supervision for relation extraction via piecewise convolutional neural networks. In: Zeng A (ed) D. Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Lisbon, pp 1753–1762
21. Hochreiter S, Schmidhuber J (1997) Long short-term memory. J Neural Comput 9(8):1735–1780
22. Marcheggiani D, Titov I (2017) Encoding sentences with graph convolutional networks for semantic role labeling. In: Marcheggiani A (ed) Empirical methods in natural language processing. Association for Computational Linguistics, Copenhagen, pp 1506–1515
23. Zhang Y, Yang J (2018) Chinese NER using lattice LSTM. In: Zhang Y. Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Melbourne, pp 1554–1564