applied sciences

MDPI

Article

Research on Chinese Medical Entity Relation Extraction Based on Syntactic Dependency Structure Information

Qinghui Zhang, Meng Wu, Pengtao Lv *, Mengya Zhang and Lei Lv

Henan Grain Big Data Analysis and Application Engineering Research Center, Henan University of Technology, Zhengzhou 450001, China
* Correspondence: pengtaolv@163.com

Abstract: Extracting entity relations from unstructured medical texts is a fundamental task in the field of medical information extraction. In relation extraction, dependency trees contain rich structural information that helps capture the long-range relations between entities. However, many models cannot effectively use dependency information or learn sentence information adequately. In this paper, we propose a relation extraction model based on syntactic dependency structure information. First, the model learns sentence sequence information by Bi-LSTM. Then, the model learns syntactic dependency structure information through graph convolutional networks. Meanwhile, in order to remove irrelevant information from the dependencies, the model adopts a new pruning strategy. Finally, the model adds a multi-head attention mechanism to focus on the entity information in the sentence from multiple aspects. We evaluate the proposed model on a Chinese medical entity relation extraction dataset. Experimental results show that our model can learn dependency relation information better and has higher performance than other baseline models.

Keywords: dependency information; graph convolutional networks; pruning operation; attention mechanism

check for updates

Citation: Zhang, Q.; Wu, M.; Lv, P.; Zhang, M.; Lei, L. Research on Chinese Medical Entity Relation Extraction Based on Syntactic Dependency Structure Information. *Appl. Sci.* **2022**, *12*, 9781. https://doi.org/10.3390/app12199781

Academic Editor: Yu-Dong Zhang

Received: 11 August 2022
Accepted: 26 September 2022
Published: 28 September 2022

1. Introduction

Obtaining useful information from the vast amount of medical resources is one of the main problems facing modern healthcare. Information extraction is a fundamental step in text analysis [1]. Information extraction includes named entity recognition, relation extraction, and event extraction [2]. Medical entity relation extraction is the classification of relation categories between entity pairs in unstructured medical texts. These relations exist in the form of triples (<subject, predicate, object>), which are called entity relations triples. Relation extraction is the key and difficult part of information extraction.

With the rapid development of biomedical text information extraction technology, there are more methods for relation extraction tasks [3]. Early studies used dictionary-based and medical domain-related knowledge bases to manually construct rule templates to accomplish relation extraction of medical entities [4]. Later, some scholars applied machine learning methods to medical text relation extraction and regarded the relation extraction task as a classification problem to recognize the relation between entities [5]. Recently, deep learning methods have been most widely applied in medical relation extraction, with recurrent neural networks (RNNs) [6], convolutional neural networks (CNNs) [7], and pre-trained language models being the mainstream neural networks currently used for relation extraction.

Although many methods have achieved good results in relation extraction tasks, there are still many difficulties in Chinese medical relation extraction tasks. Chinese medical texts have flexible expressions, complex sentence structures, and different methods of text analysis. Each Chinese sentence is not separated by a separator; rather, a series of consecutive Chinese characters are connected into a sentence. It is a crucial task to correctly divide words according to the semantics. Errors in Chinese word cutting can greatly affect the results of relation extraction. At present, Chinese word-cutting methods include classical mechanical word-cutting [8], statistical word-cutting [9], and neural network methods [10]. For example, Yuxuan Lai et al. [11] proposed a novel Chinese pre-training paradigm, Lattice-BERT, which explicitly combines word representations with characters so that sentences can be modeled in a multi-granularity manner.

Currently, relation extraction methods can be divided into two categories: sequence-based and dependency-based. Sequence-based approaches use only word embeddings as input to the model, while dependency-based models merge dependency trees into the model. Graph convolutional networks (GCNs) can extract spatial features on topological graphs to learn the information on the whole graph [12]. In existing studies, syntactic information is also widely used in relation extraction tasks. Syntactic dependency relations have better semantic guidance and entity relation information of sentences [13]. Compared with the sequence-based approach, the dependency-based approach can better obtain non-local entity relation information from sentences.

According to the above analysis, the Chinese medical relation extraction model needs to learn the sequence information of sentences. Apart from that, syntactic dependency information should be considered. Therefore, this paper proposes a Chinese medical relation extraction model, BAGCN (BiLSTM + Attention + GCN), based on syntactic dependency structure information. The model captures sentence-dependent structural information and sequence information through a graph convolutional neural network (GCN) and a bidirectional long short-term memory neural network (Bi-LSTM). In addition, we incorporate a new pruning operation in the model, considering the effect of noise on the dependency information. Finally, the model applies a multi-head attention mechanism to learn entity-related information from different perspectives. In this way, we make full use of the sequence information and dependency information of sentences to extract entity-relation triples.

The main contributions of this paper are as follows:

(1) The model constructs each sentence as syntactic dependency trees to learn the information of sentences. The dependency tree contains syntactic information and relation structures between words in a sentence. The hidden features of entity relations in the sentence can be fully explored by learning the dependency relations.

(2) The model combines BiLSTM and GCN to extract feature information together. BiLSTM can learn sentence sequence features at a shallow level, and GCN can fully learn node information in the dependency relation graph. By combining BiLSTM and GCN, the model can better learn the global feature information of the sentence.

(3) The model adopts a novel pruning strategy to remove the noise in the dependency tree. In this paper, the shortest dependency path between two entities in the dependency tree is constructed as the shortest path tree. The nodes connected to the head and tail entities form the local dependency tree. Then, the shortest path tree and the local dependency tree are combined to construct the final pruned tree. This pruning method both removes the redundant information in the sentence and retains the important information.

(4) The model introduces a multi-head attention mechanism to learn multi-perspective semantic information of sentences. The multi-head attention mechanism can automatically learn the importance and relevance of words in a sentence based on contextual information and multi-dimensional spatial information, further improving the performance of the relation extraction model.

2. Related Research

Relation extraction is one of the most important tasks in information extraction, and its purpose is to determine the relations between pairs of entities. Most of the relation extraction methods are applied in the general-purpose domain, but relation extraction tasks in the medical domain have different characteristics. Currently, the main methods of entity relation extraction in the medical field include rule-based approaches, machine learning approaches, and deep learning approaches.

2.1. Rule-Based Approach

Early methods of relation extraction mainly used rules formulated by experts in the relevant professional fields as the basis for extracting relations in texts, that is, a rule-based relation extraction method. However, this approach often requires many specialized domain personnel to spend a lot of time developing rules. In recent years, experts and scholars have also made some improvements in the generation of rules to address this deficiency. Leaman et al. [14] used a dictionary-based matching algorithm to add medical terminology terms from the National Center for Biotechnology Information disease corpus and the Unified Medical Language System to the lexicon for matching, achieving an F1 value of 63.7%. Rule-based relation extraction has high accuracy when the template design is accurate, but portability and scalability are poor. The rule-based approach has low robustness, requires manual construction for each relation, and does not yet recognize entity-pair relations beyond the template.

2.2. Machine Learning Approach

The traditional machine learning-based approach solves the relation extraction task as a classification problem. Supervised machine learning-based approaches train the parameters of the classifier with a labeled training dataset, and then test the model performance on a test dataset of unlabeled categories. Methods based on feature engineering and kernel function are machine learning methods. The feature engineering-based approach requires explicit conversion of specific relation instances into feature vectors that are acceptable to the classifier. The kernel function-based approach is a direct extraction of the instance structure tree and uses the kernel function instead of using the inner product of feature vectors when determining the spacing in entity relations. Rink et al. [15] proposed a supervised machine learning approach to discover the relations between medical problems, treatments, and tests mentioned in electronic medical records. The method was used to identify relations between concepts and assign their semantic types through a single support vector machine classifier. Alimova et al. [16] proposed a model of novel embedding features based on knowledge and BioSentVec. The method systematically investigates these features. In addition, the method investigates the effects of distance-based and word-based features. SVM and decision trees are also classical machine learning classification methods. SVM automatically finds those support vectors that have a better discriminatory power for classification, and the resulting classifier is constructed to maximize the district classification and class interval. Zhang et al. [17] proposed a text classifier construction method based on fuzzy support vector machines with decision trees. The method measures inter-sample relationships based on SVM and parallel to the plane of the classification plane as a tangent sphere, and it is combined with a decision tree to effectively solve the multi-classification problem. Decision trees are constructed using a top-down recursive approach to the tree, and test attributes are selected at each node of the tree using an information gain metric. Abu-halaweh et al. [18] proposed a fuzzy decision tree method. The method reduces the sensitivity of the generated decision tree to changes in attribute values through a fuzzy algorithm. In addition, a threshold value on the affiliation value of the object is introduced, thereby reducing the number of rules needed for decision-making. This method reduces the running time and improves classification accuracy. Fuzzy classification is widely used in machine learning. Fuzzy classification deals with linguistic uncertainty in instances based on fuzzy logic, in which case each instance is not explicit and has different affiliations in different classifications. Traditional fuzzy methods use each classifier separately for classification. Levashenko et al. [19] proposed a new fuzzy ID3 algorithm for generating comprehensible fuzzy classification rules. The method can accurately estimate the interaction of attributes by cumulative information estimation, thus finding a sequence of rules with close to minimum classification cost. Although traditional machine learning-based methods for biomedical relation extraction reduce the manual burden to some extent compared with rule-based methods, they require a large amount of feature engineering and need further improvement in recognition rate.

2.3. Deep Learning Approach

With the application of deep learning methods in various fields, researchers have started to apply them to biomedical relation extraction work. Currently, CNN, LSTM, and Transformer are the mainstream methods in the field of medical entity relation extraction. He et al. [20] proposed a convolutional neural network architecture with multi-pooling operations for medical relation classification of clinical records and explore a loss function with a category-level constraint matrix. Bai et al. [21] designed a new paragraph attention mechanism based on a convolutional neural network, extracted semantic local features through word embedding, and then connected different embedding features to classify relations. This method achieved good results in the Chinese Herbal Disease and Herbal Chemistry (HD–HC) dataset. Eberts et al. [22] proposed a hybrid model including a converter-based encoding layer, an LSTM entity detection module, and a reinforcement learning-based relation classification module, which greatly reduces the transmission of errors in relation extraction by deep learning methods. Yuan et al. [23] proposed the first model based on a recurrent neural network to classify relations in clinical records. This method also explored the differences between different contexts in sentences and evaluated the influence of word embedding on the performance of the LSTM model. Sangrak et al. [24] proposed an improved binomial tree LSTM model that combines word vectors with features such as location and syntactic information. The model provides multiple patterns for the detection and classification of drug interaction relations. The F1 value of relation detection reached 83.8%, and the F1 value of relation classification reached 73.5% in the DDI2013 evaluation data. Lin et al. [25] proposed a remotely supervised sentence-level attention-based convolutional neural network relation extraction model. The model uses CNN to convert each sentence of the input into a sentence vector, then adds an attention mechanism between sentences, and, finally, adds sentence information based on the removal of noise, significantly improving the performance of the model. Lee et al. [26] used BERT (pre-trained language model) as a bi-directional encoder pair to extract entities and relations from biomedical and clinical records, and their proposed approach achieved advanced performance on many biomedical and clinical datasets compared to other model systems. Sun et al. [27] introduced biomedical domain knowledge containing conceptual information such as proteins and compounds based on BERT, and assigned weights to feature representations through Gaussian probability distributions; the model achieved an accuracy of 76.56% on the CPI dataset. Wu et al. [28] used BERT for the relation extraction task. The method explored the way entities and entity locations are combined in the pre-trained model.

Graph neural network-related techniques can be combined with deep learning to efficiently process graph-type input data. In the biomedical field, research on relation extraction using graph neural networks for graph structure representation learning has emerged. The graph neural network can learn the dependency relations between words in a sentence to better explore the whole information of the sentence [29]. Song et al. [30] combined a graph recurrent neural network (GRN) based on BiLSTM, and the graph-based neural network architecture can better model biomedical sentences with complex hierarchical structures and effectively improve the feature extraction ability of the model. Yan Zhang et al. [31] proposed an attention-guided graph convolutional network (AGGCN) model with a full dependency tree as input, which makes full use of the information in the dependency tree in order to better extract the correlations. Geng et al. [32] proposed an end-to-end approach based on a bidirectional tree structure for long short-term memory. The method is used to identify word-based features and location information of entity pairs by BiLSTM. In our work, we not only learn sequence information through BiLSTM, but we also learn dependency information through GCN. We also perform pruning operations on the dependency tree. In addition, the multi-head attention mechanism is used to learn the multi-dimensional information of the sentence.

3. Model

Medical sentences are often complex, with a large specialized vocabulary and long-range dependencies between entities, and they often contain important syntactic information. To better learn dependencies between entities and use sentence structure information, this paper proposes a relation extraction model based on syntactic dependency structure information. First, the model performs vector embedding of the input text and transforms the sentences into three vectors (character vector, lexical feature vector, and entity feature vector) for stitching to better represent the sentence information in the semantic encoding stage. Second, the input vectors are encoded and learned by the BiLSTM layer to obtain the sequence information of the sentences. Third, the model constructs each sentence as a syntactic dependency tree by the LTP tool. Then, we perform a pruning operation on the obtained dependency tree, and the pruned dependency tree is transformed into a graph structure. Fourth, the sequence information and graph information are jointly transferred to the graph convolution layer for convolution operation. In addition, a multi-head attention layer is added after the graph convolution layer to learn the weights of different entities. Finally, the output information is transferred together to the relation classification layer for relation classification. The model consists of an input layer, a BiLSTM layer, a GCN layer, a multi-head attention layer, and a classification layer. The model diagram is shown in Figure 1.

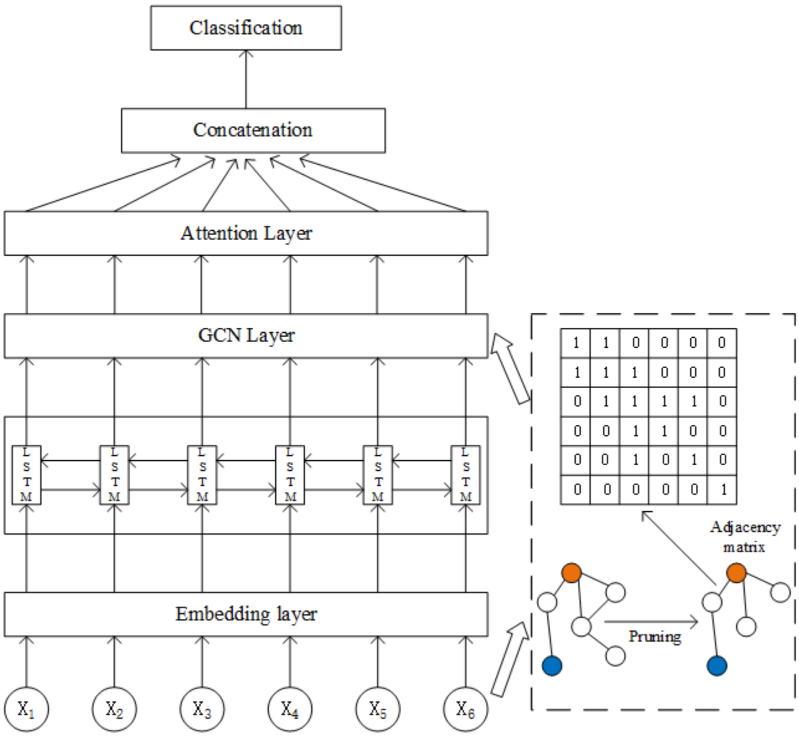


Figure 1. BAGCN model diagram.

3.1. Input Layer

3.1.1. Corpus Pre-Processing

How to transform unstructured medical texts into graph structures is the basis for learning using graph convolutional neural networks. The model requires pre-processing of the experimental corpus. First, the Chinese Language Technology Platform (LTP) [33] developed by the Harbin Institute of Technology was used for analysis and processing of the input sentences. The tool provides Chinese language processing modules such as Chinese word separation, lexical tagging, syntax, and semantics. The model input in this paper consists of a character vector, a lexical feature vector, and a type feature vector. The character vectors are obtained by training from Chinese dictionaries. The lexical feature vectors are obtained by word tagging of the input sentences by the LTP tool. Since there are different types of relations in the corpus, we added type features to improve classification. The type feature vector is set according to the entity type. If a character belongs to a certain entity after Chinese word segmentation, it is set to that entity label; otherwise, it is set to UNK.

Syntactic information contains important grammatical information in a sentence. Especially in the medical field, there are pairs of entities in a sentence that are closely connected, and some pairs of entities that are distant from each other. The dependency relation tree constructed by syntactic analysis can provide long-distance connections between words and can also build a graph structure of all entity relations in a sentence. When converting the input text into a graph structure, this paper constructs a dependency relation tree of the sentences by syntactically parsing the input sentences with the LTP tool. Dependency trees focus on the grammatical relations between words in a sentence. In addition, dependency trees can constrain grammatical relations into a tree structure. In a sentence, if a word modifies another word, the modifier is called dependent, the modified word is called head, and the grammatical relationship between the two is called dependency relation. In the dependency tree, the direction of the arrow is from the head to the dependent. The dependency tree is obtained by representing all word dependencies in a sentence in the form of directed edges. The graph convolutional neural network can learn the syntactic information of sentences from the dependency relations. Syntactic analysis can learn well the grammatical structure of sentences and the dependency between words according to the content of sentences. The dependency tree obtained by parsing a given sentence is shown in Figure 2:

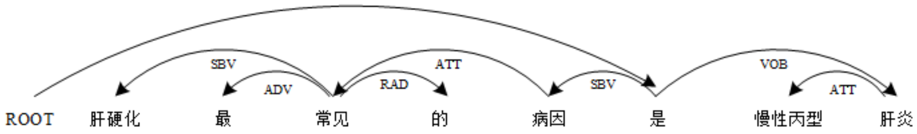
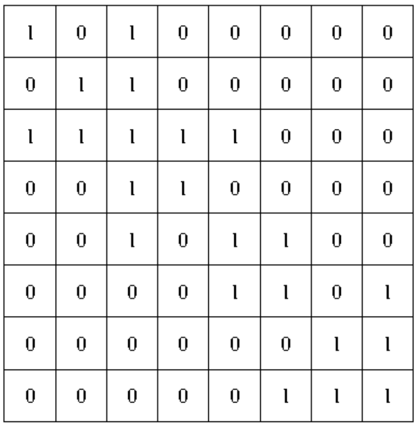


Figure 2. Dependency tree. The Chinese sentence in the figure means "The most common cause of liver cirrhosis is chronic hepatitis C". "肝硬化" represents "liver cirrhosis", "最" represents "the most", "常见" represents "common", "的" represents "of", "病因" represents "cause", "是" represents "is", "慢性丙型" represents "chronic C", "肝炎" represents "hepatitis". Root, root node; SBV, subject–predicate relation; ADV, dative–medial relation; ATT, definitive–medial relation; RAD, right-additive relation; VOB, verb–object relation.

After obtaining the dependency relation tree, the model needs to transform the tree structure into a form that can be computed by a graph convolutional neural network. Usually, standard graph convolutional neural networks are constructed based on word dependencies and are represented by adjacency matrices. The adjacency matrix can represent the relations between vertices in the graph and also store edge information. If a dependency diagram, $G = (V, E)$, where $V = v_1, v_2, ..., v_n$ denotes the vertices in the graph, and $E = e_{v_1,v_2}, e_{v_2,v_3}, ..., e_{v_m,v_n}$ denotes the set of edges in the graph. We use the adjacency matrix $A = (a_{ij})_{n \times n}$ to represent the dependency graph. As shown in Equation (1), when $i = j$ or when there is a connection between nodes V_i and V_j in the dependency tree, $a_{ij} = 1$, otherwise $a_{ij} = 0$.

$$a_{ij} = \begin{cases} 1 & i = j \quad or \quad e_{v_i,v_j} \in E \quad or \quad e_{v_j,v_i} \in E \\ 0 & otherwise \end{cases} \tag{1}$$

where e_{v_i,v_j} denotes the edge between node v_i and v_j in the dependency tree or 0 if no edge exists. According to Equation (1), we can transform the dependency tree of Figure 2 into an adjacency matrix, as shown in Figure 3.



1	0	1	0	0	0	0	0
0	1	1	0	0	0	0	0
1	1	1	1	1	0	0	0
0	0	1	1	0	0	0	0
0	0	1	0	1	1	0	0
0	0	0	0	1	1	0	1
0	0	0	0	0	0	1	1
0	0	0	0	0	1	1	1

Figure 3. Adjacency matrix diagram.

3.1.2. Pruning Operation

To construct the dependency graph, we construct a dependency tree for each input sentence. The model captures long-range word relations and hidden sentence features in sentences through dependency trees. However, most studies nowadays are affected by the noise in the dependency trees, and excessive use of dependency information may confuse the relation classification. In particular, there is a lot of noise in the automatically generated dependency trees, and ignoring this noise can impact the results and computational complexity. Therefore, it is necessary to prune the dependency trees.

Usually, pruning operations use distributed methods to resolve noise, such as lowest common ancestor subtree pruning (a subtree is formed by using the common node closest to two entities as the root) and shortest dependency path tree pruning (preserving the shortest path between two entities in the dependency tree). In the relation extraction task, the dependency structure of sentences contains rich information, but there is also redundant information in the complete dependency tree. Useless information in the dependency tree can interfere with the model, but pruning operations on the dependency tree may ignore some important information in the sentence. In order to remove the nodes with irrelevant information in the dependency tree while effectively using some important information in the sentences, a new pruning strategy is proposed in this paper.

Although the lowest common ancestor subtree and shortest dependency path methods can remove some useless nodes, there is a possibility that some dependency information will be lost during pruning, and that even critical information will be lost. As shown in Figure 4, we propose a combination of a local subtree and the shortest path tree to construct the input graph. The local subtree contains all the dependencies directly connected to the head entity and the tail entity. The shortest path tree contains all the dependencies on the shortest path between two entities. In a complete dependency tree, the path from the root node through the least number of nodes to the head and tail entity nodes is the shortest path. The shortest dependency path can effectively represent the structure of semantic relationships between entities, and the path contains the lexical information on the path between the root node and the head and tail entity nodes. In the pruning operation, we retain all node relations contained in the local subtree and all node relations contained in the shortest path tree. We consider the words removed from the sentence by the two pruning operations as noise, and the words retained by the two pruning operations as actual retention. The dependency tree is pruned into two subtrees by two pruning operations, and then, the final dependency tree is formed based on the node dependencies retained by the two subtrees. When transforming the pruned dependency tree into an adjacency matrix, the corresponding adjacency matrix value is set to 1 for the retained nodes and 0 for the deleted nodes. As shown in Figure 5, our final dependency relation graph is composed of two different dependency relation graphs, which allows the dependency relation graph to have reduced noise while retaining valid information.

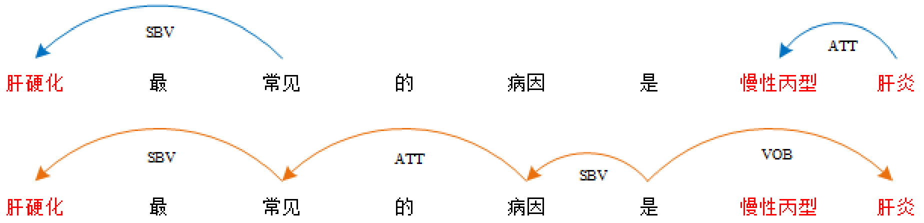


Figure 4. Two pruning operation diagrams: top is local subtree pruning; bottom is shortest-path tree pruning. SBV, subject–predicate relation; ATT, definitive–medial relation; VOB, verb-object relation. The Chinese sentence in the figure means “The most common cause of liver cirrhosis is chronic hepatitis C”. “肝硬化” represents “liver cirrhosis”, “最” represents “the most”, “常见” represents “common”, “的” represents “of”, “病因” represents “cause”, “是” represents “is”, “慢性丙型” represents “chronic C”, “肝炎” represents “hepatitis”.

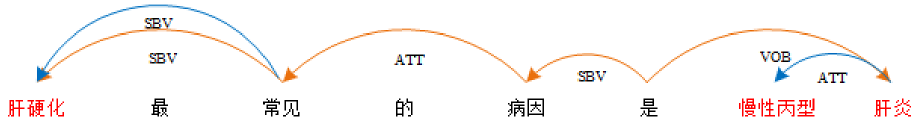


Figure 5. Diagram of the novel pruning strategy for fusion. SBV, subject–predicate relation; ATT, definitive–medial relation; VOB, verb–object relation. The Chinese sentence in the figure means “The most common cause of liver cirrhosis is chronic hepatitis C”. “肝硬化” represents “liver cirrhosis”, “最” represents “the most”, “常见” represents “common”, “的” represents “of”, “病因” represents “cause”, “是” represents “is”, “慢性丙型” represents “chronic C”, “肝炎” represents “hepatitis”.

3.2. BiLSTM Layer

BiLSTM can acquire the features of the context, and in order to fully learn the sentence information, this paper uses the BiLSTM layer to encode and model the sentences. The input of the BiLSTM layer is composed of word vector representation (c_n), lexical feature representation (p_n), and type feature representation (t_n) together. In the relation extraction task, a sentence may contain multiple entity types. The type feature representation can help the model identify the target entities more accurately. For the input sentence $S_n = w_1, w_2, \ldots, w_n$, the input of BiLSTM is represented by three feature vectors, as shown in Equation (2).

$$X_i = (c_i, p_i, t_i) \quad i \in (1, 2, \ldots, n) \tag{2}$$

For a joint embedding vector X_i at any position i in the input sequence, the LSTM will combine X_i and the state h_{i-1} from the previous moment to calculate the hidden state h_i at the current moment. BiLSTM can effectively memorize the context information by setting two independent hidden layers. Finally, calculate the forward representation $\overrightarrow{h_i}$ and backward representation $\overleftarrow{h_i}$ of any input X_i to obtain the final i moment hidden state h_i. The hidden state h_i contains both the sentence forward information, the backward information of the sentence, and the current input X_i. Thus, BiLSTM can learn sentence bidirectional semantic information better.

$$h_i = \overrightarrow{h_i} \oplus \overleftarrow{h_i} \tag{3}$$

3.3. GCN Layer

GCN is a simple and effective graph-based convolutional neural network that learns information for graph nodes containing all neighboring nodes and its own nodes, as shown in Figure 6. GCN acts directly on the graph, and its inputs are the graph structure and the feature representation of the nodes in the graph. The model proposed in this paper learns the dependency information in the input sentence dependency relation tree by a graph convolutional neural network.

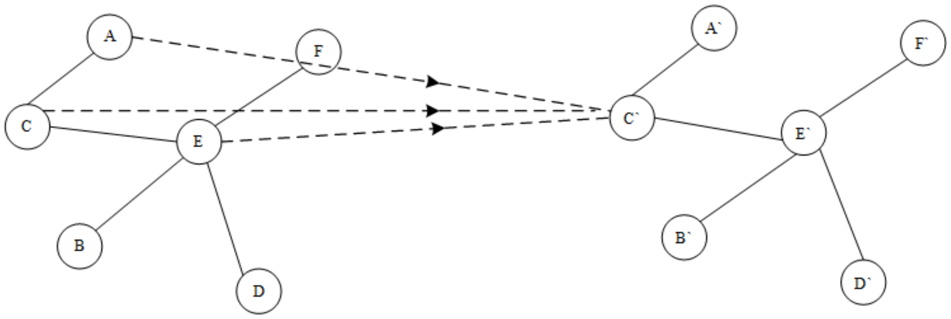


Figure 6. GCN local convolution graph.

Firstly, the sentences are preprocessed, and the feature vectors obtained by encoding the word segmentation through the BiLSTM layer are used as nodes in the graph. The relations between different nodes in the results of the dependency analysis are then used as edges that constitute the graph structure of the graph convolutional neural network. To reduce the effect of noise in the sentences, the model uses a pruning strategy on the dependency tree and transforms the pruned dependency graph into the adjacency matrix A.

Based on the adjacency matrix A, for each node $v_i \in V$, the GCN at layer l learns the node information on the dependency relation tree and calculates the output of node v_i at layer l as $h_{v_i}^{(l)}$. The specific calculation is shown in Equations (4) and (5):

$$h_{v_i}^{(0)} = x_{v_i} \quad v_i \in V \tag{4}$$

$$h_{v_i}^{(l)} = f(\sum_{j=1}^{n} a_{i,j} W^l h_{v_j}^{l-1} + b^{(l)}) \quad v_i, v_j \in V \tag{5}$$

where $h_{v_i}^{(0)}$ represents the initial embedding of node v_i, x_{v_i} represents the original feature of node v_i, $h_{v_j}^{l-1}$ represents the hidden state of node v_j after $l-1$ layer graph convolutional neural network, w^l represents the weight matrix, $b^{(l)}$ represents the bias, $a_{i,j}$ represents the corresponding elements of node i and node j in the adjacency matrix, $f(.)$ represents a nonlinear function and is the ReLU function in this model, and $h_{v_i}^{(l)}$ represents the hidden state of node v_i after the l layer convolutional neural network. For each layer of GCN, functions $f(.)$, matrix W^l, and matrix $b^{(l)}$ are shared on all nodes, which makes the number of parameters on the model irrelevant to the graph, enabling the GCN model to be well-extended.

3.4. Attention Layer

The self-attention mechanism can learn the internal structure of sentences and can learn the weight between every two nodes according to the correlation information between words. However, the weight vector usually obtained by the self-attention mechanism can only represent one-sided information of the sentence. Medical texts have a high entity density distribution, and for a sentence, there may be multiple aspects of semantic information that together constitute the overall information of the sentence. To be able to capture the dependency information between each node in the graph structure in a multi-dimensional way, this model uses a multi-head attention mechanism to learn the weight information between nodes from different semantic spaces.

The calculation of the attention mechanism involves a set of vectors Q, K, V. Firstly, the current input h is multiplied by three independent parameter vectors w^q, w^k, w^v. Then, the attention is calculated by zooming the dot product, as shown in Equation (6):

$$Attention(Q, K, V) = softmax(\frac{Q^T K}{\sqrt{d_k}})V \tag{6}$$

where Q, K, V represent the query matrix, key matrix, and value matrix, respectively, $\frac{1}{\sqrt{d_k}}$ is used as the scaling factor, and $softmax$ normalizes the result of $Q^T K$.

The multi-head attention module contains multiple heads for parallel computation, and N is the number of heads. Q, K, and V will be mapped N times independently using different parameter matrices, and then will be input to N parallel heads for attention computation. Finally, the attention results of the N heads are stitched together and then linearly transformed to obtain the final output. The specific calculation process is shown in Equations (7) and (8).

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad i \in [1, N] \tag{7}$$

$$at_t = Multihead(Q, K, V) = concat(head_1, head_2, ..., head_n)W^o \tag{8}$$

where $QW_i^Q, KW_i^K, VW_i^V, W^o$ are the parameter matrices used in the linear mapping, $head_i$ denotes the i-th head attention module, and $concat$ is the splicing multi-head operation.

3.5. Classification Layer

After the attention layer, the model has sufficiently learned the complete information of the sentence. The role of the relation classification layer is to classify the relations between entities based on the learned information. The model input contains entity type labels, and span mask vectors for the head and tail entities are also input to the model. In this paper, the vector representation of the sentence h_{all} is obtained by doing the maximum pooling function on the output vector representation h_{att} of the attention layer. The same maximum pooling function is also used to obtain the entity vector representations h_s and h_o from h_{att}. The maximum pooling function is calculated as follows:

$$h_{all} = f(h_{att}) \tag{9}$$

$$h_s = f(h_{att} \times S_{span}) \tag{10}$$

$$h_o = f(h_{att} \times O_{span}) \tag{11}$$

where $f(.)$ denotes the maximum pooling function, and S_{span} and O_{span} represent the span of the head and tail entities, respectively.

The model concatenates the vector representation of sentences and the vector representation of entities to transmit them to the fully connected layer to obtain the final representation. Finally, the relation probability distribution of entity pairs is predicted by the $softmax$ function. The calculation is shown in Equations (12) and (14).

$$h_{final} = FNN(h_{all}, h_s, h_o) \tag{12}$$

$$p(\frac{r}{h_{final}}) = softmax(w_r h + b_r) \tag{13}$$

$$T = argmax(P) \tag{14}$$

where r denotes the total relation type, w_r is the parameter matrix, b_r is the bias term, and T denotes the final prediction label.

4. Experiment

4.1. Dataset and Experimental Setup

4.1.1. Dataset

The dataset of this paper is from the public Chinese medical entity extraction evaluation task published by the competition. The content of this dataset consists of clinical practice texts and medical textbooks. In this paper, the dataset was processed and filtered before the experiment. In the dataset, each medical text contains multiple entity relation categories. The dataset used for the experiments includes 53 predefined relations, with a total of 14,289 sentences for training and 3567 sentences for testing.

4.1.2. Evaluation Indicators

In this paper, the models are evaluated using the standard precision, recall and F1 (F-measure) values in natural language processing. Let r_i be a predefined relation in the relation set, TP_i (True Positives) denotes the correct prediction relation r_i, FP_i (False Positives) denotes the predictions labeled true but actually false, and FN_i (False Negatives) denotes the predictions labeled false but actually true. The specific calculation process is as follows:

$$P_i = \frac{TP_i}{TP_i + FP_i} \tag{15}$$

$$R_i = \frac{TP_i}{TP_i + FN_i} \tag{16}$$

$$\mathrm{F1}_i = \frac{2 \times P_i \times R_i}{P_i + R_i} \tag{17}$$

The dataset in this paper contains a variety of predefined relations, so Marco-F1 is used to evaluate the model performance. The problem of an unbalanced amount of relation data exists in the dataset. The use of Marco-F1 ignores the effect brought by the training of sample size of different types of relations, considers the effect of rare categories relatively, and is not susceptible to the effect of large sample types when the samples are unbalanced. The calculation process is as follows:

$$\text{Marco-P} = \frac{1}{n}\sum_{i=1}^{n} P_i \tag{18}$$

$$\text{Marco-R} = \frac{1}{n}\sum_{i=1}^{n} R_i \tag{19}$$

$$\text{Marco-F1} = \frac{1}{n}\sum_{i=1}^{n} F1_i \tag{20}$$

where n is the predefined relation categories in the dataset, and Marco-F1 is the average of all relation categories F1.

4.1.3. Parameter Setting

The hyperparameter configuration of the model in this paper is shown in Table 1.

Table 1. Model parameter configuration.

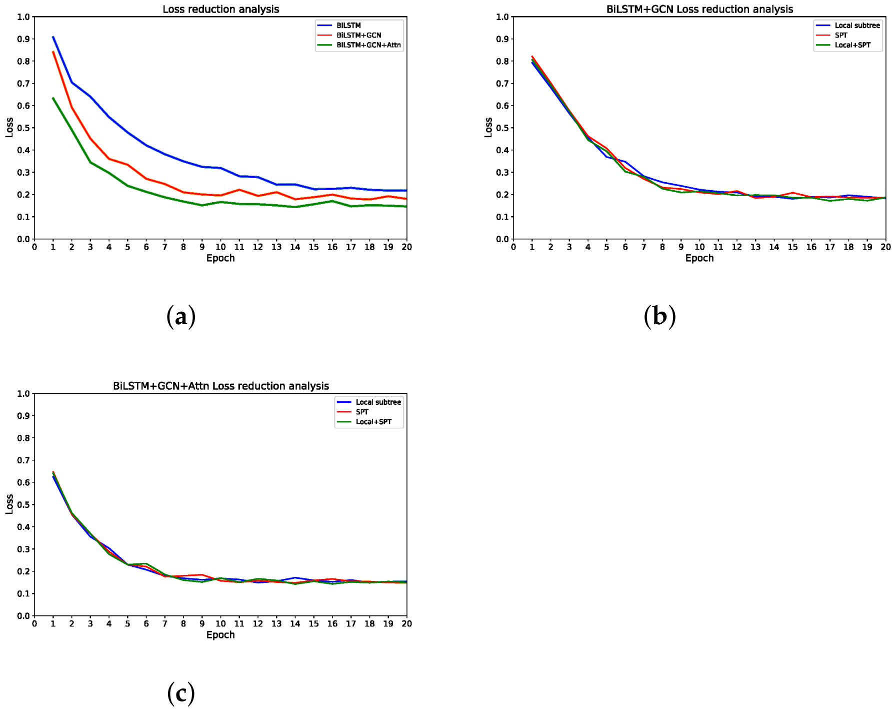
Parameter Name	Value	Parameter Name	Value
Batch size	64	GCN layers	2
Embedding size	64	GCN dropout	0.5
LSTM hidden size	128	Learning rate	8×10^{-5}
LSTM layers	2	Optimizer	Adam
LSTM dropout	0.5	Attention heads	8
Epoch	20	Layer normalization	1×10^{-12}

Batch size denotes the size of the single input data of the model. Embedding size denotes the size of the embedded model vector. LSTM hidden size denotes the size of the hidden layers of the LSTM. GCN layers denote the number of layers of the GCN. Learning rate denotes the learning rate set by the model. Optimizer denotes the model optimizer. Attention heads denotes the number of multi-head attention heads. Layer normalization can make the obtained model more stable and play the role of regularization.

4.2. Experimental Results and Analysis

4.2.1. Experimental Results with Different Parameters

Different datasets require different epochs for training, and a suitable epoch allows the loss to converge to a stable value. Choosing the best epoch allows the model to achieve the best results while reducing the training time. As shown in Figure 7, we explored the loss descent plots for different models with the same learning rate. The model loss converges faster until 10 epochs, and then the loss decreases smoothly with increasing epochs. At close to 20 epochs, all model losses have only a weak float. Therefore, we set the training epoch to 20, thus reducing meaningless training time.



(**a**)

(**b**)

(**c**)

Figure 7. Comparison of training loss reduction for different models on the same dataset. (**a**) Loss reduction plots for different models with the same parameters. (**b**) Comparison chart of loss reduction for fusing three pruning operations on the BILSTM+GCN model. (**c**) Comparison chart of loss reduction for fusing three pruning operations on the BILSTM+GCN+Attn model.

As shown in Figure 7, the experiment compares the loss reduction states of the three models. Trained with the same parameter settings, the lowest loss in the experiment is 0.2173 for the BiLSTM model, 0.1773 for the BiLSTM+GCN model, and 0.1430 for the BiLSTM+Attn+GCN model. compared with the BiLSTM model and the BiLSTM+GCN model, our proposed BiLSTM+Attn+GCN model converges faster and has the smallest loss value, indicating that the model predicts results more accurately and classifies better. Comparing (b) and (c) in Figure 7 shows the change in loss decline of the two models after incorporating the three pruning operations. The experimental results show that the loss of the model converges faster after adding the pruning operation. It shows that by adding pruning, the redundant weights in the model are removed, and computational efficiency is improved at the same time. The experiments compare the BiLSTM+GCN (add pruning) model and BiLSTM+Attn+GCN (add pruning) model. The latter has the fastest convergence speed and the lowest loss value, which indicates that the model has the best classification.

The experiments also compare the effects of different learning rates on model performance. The learning rate comparison experiments are validated on a BAGCN model that includes a novel pruning operation. The learning rate directly affects the convergence state of the model, and choosing a suitable initial learning rate can lead to better model training.

If the initial learning rate is too large, the model will not converge. If the initial learning rate is too small, the model will converge too slowly. As shown in Figure 8, the degree of convergence of model loss is different at different learning rates. When the learning rate is small, the model converges slowly. When the learning rate is larger, the loss of the model first decreases and then increases. Too large a learning rate leads to overfitting the data in the training set, which makes the model less generalizable. When the learning rate is 8×10^{-5}, the loss of the model is minimized, and the loss tends to smooth out as the number of training iterations increases, indicating that an appropriate learning rate can lead to better convergence of the model loss.

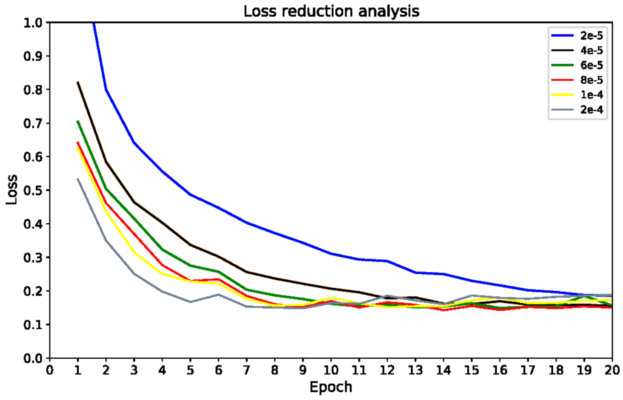


Figure 8. Loss convergence diagram for different learning rates of the model.

As shown in Figure 9, this paper also verifies the comparison of model results at different learning rates. The experiments show that the F1 value of the model gradually increases to the maximum and then decreases as the learning rate increases. In the experimental results, Marco-P is 0.721, and Marco-R is 0.642 for a learning rate of 4×10^{-5}. Marco-P is 0.695, and Marco-R is 0.662 for a learning rate of 6×10^{-5}. The Marco-P decreased by 0.026 and the Marco-R increased by 0.020 at a learning rate of 6×10^{-5} compared to the results for 4×10^{-5}. For the changes in the experimental results occurring at a learning rate of 6×10^{-5}, we examine the sample labeling results of the model predictions. In some relation categories with a large number of samples, the model predicts a large number of negative samples as positive samples (i.e., FP becomes larger; the denominator becomes larger in the Marco-P formula), which leads to a decrease in accuracy in these categories. In several relation categories with smaller sample sizes, the model predicts fewer positive samples as negative (i.e., FN becomes smaller; the denominator becomes smaller in the Marco-R formula), which leads to higher recall in these categories. In the categories with smaller sample sizes, the labeling of each sample has a significant impact on the identification results for that category. In addition, Macro-P and Marco-R are used in this paper, and the precision and recall of all categories affect the final results equally. At a learning rate of 6×10^{-5}, the reason for the change in Marco-P and Marco-R is the large change in precision and recall for some relation categories. We used Marco-F1 to evaluate the model performance more comprehensively, thus balancing the effects of precision and recall. The Marco-F1 of the model reaches the maximum value when the learning rate is 8×10^{-5}, which indicates that the model works best at the current initial learning rate.

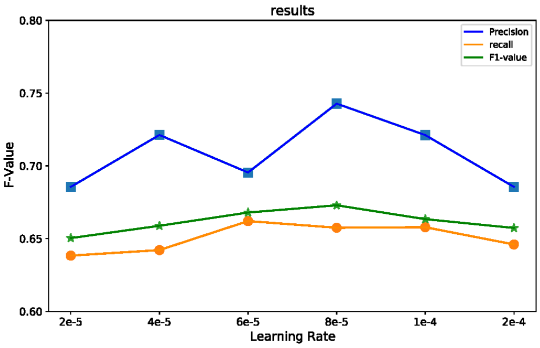


Figure 9. Comparison of experimental results of the model under different learning rates.

4.2.2. Comparison Experiment

To validate the performance of the BAGCN model, we checked the BiLSTM model and BiLSTM+GCN model against each other on a unified dataset. The experimental results are shown in Table 2. The F1 value of the BiLSTM model is 55.23%, and the F1 value of the BiLSTM+GCN model reaches 61.11%. This shows that the graph convolutional neural network can learn the dependencies in the sentences and the node information in the graph, thus improving the effectiveness of the model. The BiLSTM+Attn+GCN model has a higher *p*-value, R-value, and F1-value than the other models. Compared with the BiLSTM+GCN model, introducing a multi-head attention mechanism to learn sentence information from different semantic spaces and giving high attention scores to related entities can significantly improve the classification results. According to the experimental results, adding GCN and multi-head attention mechanism to BiLSTM can effectively improve the performance of relation classification model.

Table 2. Results of ablation experiments with different models (unit: %).

Model	*p*	R	F1
BiLSTM	61.05	54.30	55.23
BiLSTM+GCN	64.00	61.19	61.11
BiLSTM+Attn+GCN	70.98	64.52	66.15

Information about the different dependencies of the sentences can impact model training. Different pruning operations cause the model to learn different information about the sentences as well, which affects the results. Table 3 compares the results of different pruning operations of the two models, where B+G represents the BiLSTM+GCN model, and B+A+G represents the BiLSTM+Attn+GCN model proposed in this paper. Local denotes local pruning operation, SPT denotes shortest-path tree pruning operation, and Local+SPT denotes the fusion of two pruning operations. The F1 value for the B+G(Local) model is 60.44%, and the F1 value for the B+G(SPT) model is 61.18%. However, in Table 2, the F value of the B+G model is 61.11%. The F1 value of the B+G(Local) model is lower than that of the B+G model, which indicates that there may be information loss after adding local pruning to the B+G model. Because local pruning retains only the entities and their related nodes in the sentence, this leads to the neglect of important information in the sentence during pruning and reduces the relation classification results. The F1 of the B+G (Local+SPT) model was 62.17%, indicating that the B+G model incorporating the two pruning operations had the best performance. Fusion of the two pruning operations can retain the important information in the sentences and remove the useless information from the sentences. In addition, compared with the B+A+G model without pruning in Table 2, the F1 value of the B+A+G (Local+SPT) model increased by 1.14% When the B+A+G model incorporates only one pruning operation, the model does not work well. When the B+A+G model incorporates both pruning operations, the model achieves the highest *p*-value, R-value, and F1-value. This shows that the pruning strategy proposed in this paper can improve the performance of the model's relation classification.

Table 3. Results of ablation experiments with different models (unit: %).

Model	***p***	**R**	**F1**
B+G (Local)	63.04	60.28	60.44
B+G (SPT)	65.27	60.7	61.18
B+G (Local+SPT)	64.39	61.88	62.17
B+A+G (Local)	66.95	64.30	64.91
B+A+G (SPT)	67.93	62.57	63.83
B+A+G (Local+SPT)	74.28	65.75	67.29

4.2.3. Comparison of Models and Other Baseline Models

Finally, the BiLSTM+Attn+GCN (BAGCN) model proposed in this paper that incorporates pruning operations is compared with other baseline models. The other baseline models are a relation extraction model based on an SVM classifier and syntactic semantic features [34], a location-aware neural network model PA-LSTM [35], a GCN model based on contextual information [36], and an attention-guided graph convolutional network model (AGGCN) with a fully dependent tree as input [31]. Table 4 shows the results. Our BAGCN model substantially outperforms the baselines and achieves 5.80%, 2.61%, and 1.94% improvements over the state-of-art baseline method (AGGCN) in precision, recall, and F1-score, respectively. The experimental results demonstrate the effectiveness of our BAGCN model.

Table 4. Experimental results with baseline models (unit: %).

Model	***p***	**R**	**F1**
SVM	59.37	54.18	55.83
PA-LSTM	61.28	58.57	59.79
GCN	63.15	58.23	60.24
AGGCN	68.48	63.14	65.35
BAGCN (Pruning)	74.28	65.75	67.29

4.2.4. Further Analysis Based on Experimental Results

The SVM model is integrated according to the grammatical features of Chinese to obtain rich relational features between entity pairs and then trained and tested by the SVM classifier. However, as the number of entity relation types increases and the objects to be processed are complex entity relations, the optimal boundaries of various relation classes are often difficult to determine, resulting in low performance of SVM. The PA-LSTM model adds a location-aware attention mechanism to the LSTM network. The results of this paper's model are considerably better than those of the PA-LSTM model. This is because although the PA-LSTM model learns the global location of entities in the sequence, it does not pay attention to the rich dependency features in the sentences. The GCN model learns the dependency information and adds the shortest-path pruning strategy. However, it is experimentally demonstrated that this pruning method may ignore important information. The new pruning strategy proposed in this model can well avoid the over-pruning problem. The AGGCN model takes the dependency tree as input directly and uses a multi-head attention mechanism to focus on relevant substructures in the dependency tree. However, in the field of Chinese medicine, sentences are usually long, and the relations between entities are complex. Taking the dependency tree as input directly brings useless information. In addition, constructing multiple attention-weighted graphs through dependency relations also increases the computational complexity. According to Table 4, our proposed BAGCN model has better results on the Chinese medical relation extraction task. Compared with previous work, we combine sequence information and dependency relations to learn sentence information. A new pruning strategy is also adopted to effectively utilize relevant information and ignore irrelevant information in the dependency tree. In addition, a multi-head attention mechanism is added to capture the global information of the sentences. In this way, the model improves the performance of relation classification of Chinese medical entities.

4.2.5. Discussion

In all demonstrated experiments, our proposed BAGCN model significantly outperformed other methods. BiLSTM-based approaches are likely to be limited because they only learn sentence sequence information. On the other hand, most dependency-based methods do not make full use of the dependency information because they only consider the connectivity of the dependency tree and ignore the important information in the dependency tree. Our proposed model can overcome these two limitations by stacking BiLSTM sentence encoders and GCN dependency tree encoders to automatically extract the hidden features of sentences. In addition, the model incorporates a novel pruning operation and a multi-head attention mechanism to improve performance. At the same time, we propose new pruning operations that also contribute to model performance. Compared with some single-pruning operations, our pruning operation well preserves the important words in the dependency tree. Additionally, it avoids the possibility of pruning removing useful information. Compared with other baseline models, our model combines their advantages and has better prediction performance.

5. Conclusions

In this paper, we propose a Chinese medical relation extraction method based on syntactic dependency structure information. Compared with previous approaches, our model learns sequence information through BiLSTM. The model also captures syntactic dependency information in sentences through GCN, thus learning sentence information more comprehensively. In addition, we propose a new pruning operation for pruning dependencies. Finally, the model also incorporates a multi-head attention mechanism to learn sentence information from different semantic spaces. The experimental results show that our BAGCN model outperforms the baseline model on the Chinese medical entity relation extraction dataset. In addition, it is illustrated through experiments that syntactic dependency information is important in the relation extraction task.

However, there are still many difficulties in the task of Chinese medical entity relation extraction. Our model is still deficient in predicting complex medical entity relations. Our future work will focus more on relation extraction in more complex medical texts, such as document-level relation extraction and medical event relation extraction.

Author Contributions: Conceptualization, Q.Z.; methodology, M.W.; software, M.W.; validation, Q.Z., M.W. and P.L.; formal analysis, L.L. and M.Z.; investigation, M.Z.; resources, L.L.; data curation, M.W.; writing—original draft preparation, M.W. and P.L.; writing—review and editing, Q.Z., P.L. and M.Z.; visualization, M.W.; supervision, L.L.; project administration, Q.Z.; funding acquisition, Q.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (62073123) and Major Public Welfare Project of Henan Province (201300311200).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: All authors declare no conflict of interest about the representation or interpretation of reported research results.

References

1. Zhang, Y.; Lin, H.; Yang, Z.; Wang, J.; Zhang, S.; Sun, Y.; Yang, L. A hybrid model based on neural networks for biomedical relation extraction. *J. Biomed. Inform.* **2018**, *81*, 83–92.
2. Zhang, Z.; Zhou, T.; Zhang, Y.; Pang, Y. Attention-based deep residual learning network for entity relation extraction in Chinese EMRs. *BMC Med. Inform. Decis. Mak.* **2019**, *19* (Suppl. 2), 171–177.
3. Zhang, T.; Lin, H.; Tadesse, M.M.; Ren, Y.; Xu, B. Chinese medical relation extraction based on multi-hop self-attention mechanism. *Int. J. Mach. Learn. Cybern.* **2020**, *2*, 355–363.
4. Chen, T.; Wu, M.; Li, H. A general approach for improving deep learning-based medical relation extraction using a pre-trained model and fine-tuning. *Database Biol. Databases Curation* **2019**, *2019*, baz116. https://doi.org/10.1093/database/baz116
5. E, H.; Zhang, W.; Xiao, S.; Cheng, R.; Hu, Y.; Zhou, X.; Niu, P. Survey of entity relationship extraction based on deep learning. *J. Softw.* **2019**, *30*, 1793–1818.
6. Socher, R.; Huval, B.; Christopher, D.; Andrew, Y.N. Semantic compositionality through recursive matrix-vector spaces. In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing(EMNLP) and Computational Natural Language Learning(CONLL), Jeju, Korea, 12–14 July 2012; pp. 1201–1211.
7. Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; Zhao, J. Relation classification via convolutional deep neural network. In Proceedings of the 25th International Conference on Computational Linguistics(COLING), Dublin, Ireland, 23–29 August 2014; pp. 2335–2344.
8. Tang, L.; Guo, C.; Chen, J. Review of Chinese word Segmentation Studies. *Data Anal. Knowl. Discov.* **2020**, *4*, 1–17.
9. Hong, C.-M.; Chen, C.-M.; Chiu, C.-Y. Automatic extraction of new words based on Google News corpora for supporting lexicon-based Chinese word segmentation systems. *Expert Syst. Appl.* **2009**, *36*, 3641–3651.
10. Zhang, M.; Yue, Z.; Fu, G. Transition-Based Neural Word Segmentation. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; Volume 1.
11. Lai, Y.; Liu, Y.; Feng, Y.; Huang, S.; Zhao, D. Lattice-BERT: Leveraging Multi-Granularity Representations in Chinese Pre-trained Language Models. *arXiv* **2021**, arXiv:2104.07204.
12. Li, Z.; Sun, Y.; Zhu, J.; Tang, S.; Zhang, C.; Ma, H. Improve relation extraction with dual attention-guided graph convolutional networks. *Neural Comput. Appl.* **2021**, *33*, 1773–1784.
13. Du, C.; Wang, J.; Sun, H.; Qi, Q.; Liao, J. Syntax-type-aware graph convolutional networks for natural language understanding. *Appl. Soft Comput.* **2021**, *102*, 107080.
14. Leaman, R.; Lu, Z. NCBI Disease Corpus: A Resource for Disease Name Recognition and Concept Normalization. *J. Biomed. Inform.* **2014**, *47*, 1–10.
15. Rink, B.; Harabagiu, S.; Roberts, K. Automatic extraction of relations between medical concepts in clinical texts. *J. Am. Med. Inform. Assoc.* **2011**, *18*, 594–600.
16. Alimova, I.; Tutubalina, E. Multiple features for clinical relation extraction: A machine learning approach. *J. Biomed. Inform.* **2020**, *103*, 103382.
17. Zhang, Q.; Jie, Y.; Li, K. Text classifier based on fuzzy support vector machine and decision tree. *J. Comput. Appl.* **2008**, *28*, 3227–3230.
18. Abu-halaweh, N.M.; Harrison, R.W. Rule set reduction in fuzzy decision trees. In Proceedings of the NAFIPS 2009—2009 Annual Meeting of the North American Fuzzy Information Processing Society, Cincinnati, OH, USA, 14–17 June 2009; pp. 1–4.
19. Levashenko, V.; Zaitseva, E.; Puuronen, S. Fuzzy Classifier Based on Fuzzy Decision Tree. In Proceedings of the EUROCON 2007—The International Conference on "Computer as a Tool", Warsaw, Poland, 9–12 September 2007; pp. 823–827.
20. He, B.; Guan, Y.; Dai, R. Classifying medical relations in clinical text via convolutional neural networks. *Artif. Intell. Med.* **2018**, *93*, 43–49.
21. Bai, T.; Guan, H.; Wang, S.; Wang, Y.; Huang, L. Traditional Chinese medicine entity relation extraction based on CNN with segment attention. *Neural Comput. Appl.* **2021**, *34*, 1–10.
22. Eberts, M.; Ulges, A. Span-based joint entity and relation extraction with transformer pre-training. *arXiv* **2019**, arXiv:1909.07755.
23. Yuan, L. Recurrent neural networks for classifying relations in clinical notes. *J. Biomed. Inform.* **2017**, *72*, 85–95.
24. Sangrak, L.; Kyubum, L.; Jaewoo, K. Drug drug interaction extraction from the literature using a recursive neural network. *PLoS ONE* **2018**, *13*, e0190926.
25. Lin, Y.; Shen, S.; Liu, Z.; Luan, H.; Sun, M. Neural Relation Extraction with Selective Attention over Instances. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; Volume 1.
26. Lee, J.; Yoon, W.; Kim, S.; Kim, D.; Kim, S.; So, C.H.; Kang, J. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* **2020**, *36*, 1234–1240.
27. Sun, C.; Yang, Z.; Su, L.; Wang, L.; Wang, J. Chemical–protein interaction extraction via Gaussian probability distribution and external biomedical knowledge. *Bioinformatics* **2020**, *36*, 4323–4330.
28. Wu, S.; He, Y. Enriching Pre-trained Language Model with Entity Information for Relation Classification. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019.
29. El-Allaly, E.D.; Sarrouti, M.; En-Nahnahi, N.; El Alaoui, S.O. An attentive joint model with transformer-based weighted graph convolutional network for extracting adverse drug event relation. *J. Biomed. Inform.* **2022**, *125*, 4323–4330.

30. Song, L.; Zhang, Y.; Gildea, D.; Yu, M.; Wang, Z.; Su, J. Leveraging Dependency Forest for Neural Medical Relation Extraction. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 208–218
31. Zhang, Y.; Guo, Z.; Lu, W. Attention Guided Graph Convolutional Networks for Relation Extraction. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019.
32. Geng, Z.; Chen, G.; Han, Y.; Lu, G.; Li, F. Semantic relation extraction using sequential and tree-structured LSTM with attention. *Inf. Sci.* **2020**, *509*, 183–192.
33. Che, W.; Li, Z.; Liu, T. LTP: A Chinese Language Technology Platform. In *Proceedings of the Coling 2010: Demonstrations*; COLING: Beijing, China, 2010; pp. 13–16.
34. Gan, L.; Wan, C.; Liu, D.; Zhong, Q.; Jiang, T. Chinese Named Entity Relation Extraction Based on Syntactic and Semantic Features. *J. Chin. Inf. Process.* **2014**, *28*, 183–189.
35. Zhang, Y.; Zhong, V.; Chen, D.; Angeli, G.; Manning, C.D. Position-aware Attention and Supervised Data Improve Slot Filling. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 9–11 September 2017.
36. Zhang, Y.; Qi, P.; Manning, C.D. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018.