Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 19(10): 10656–10672.
DOI: 10.3934/mbe.2022498
Received: 31 May 2022
Revised: 04 July 2022
Accepted: 07 July 2022
Published: 27 July 2022

Research article

Application of cascade binary pointer tagging in joint entity and relation extraction of Chinese medical text**

Hongyang Chang[1,2], **Hongying Zan**[1,2,*], **Tongfeng Guan**[1,2], **Kunli Zhang**[1,2] **and Zhifang Sui**[2,3]

[1] School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou, China
[2] Pengcheng Laboratory, Shenzhen, China
[3] Key Laboratory of Computational Linguistics, Peking University, Beijing, China

* **Correspondence:** Email: iehyzan@zzu.edu.cn.

Abstract: Extracting relational triples from unstructured medical texts can provide a basis for the construction of large-scale medical knowledge graphs. The cascade binary pointer tagging network (CBPTN) shows excellent performance in the joint entity and relation extraction, so we try to explore its effectiveness in the joint entity and relation extraction of Chinese medical texts. In this paper, we propose two models based on the CBPTN: CBPTN with conditional layer normalization (Cas-CLN) and biaffine transformation-based CBPTN with multi-head selection (BTCAMS). Cas-CLN uses the CBPTN to decode the head entity and relation-tail entity successively and utilizes conditional layer normalization to enhance the connection between the two steps. BTCAMS detects all possible entities in a sentence by using the CBPTN and then determines the relation between each entity pair through biaffine transformation. We test the performance of the two models on two Chinese medical datasets: CMeIE and CEMRDS. The experimental results prove the effectiveness of the two models. Compared with the baseline CasREL, the F1 value of Cas-CLN and BTCAMS on the test data of CMeIE improved by 1.01 and 2.13%; on the test data of CEMRDS, the F1 value improved by 1.99 and 0.68%.

Keywords: Chinese medical; joint entity and relation extraction; cascade binary pointer tagging

1. Introduction

Medical texts, including medical textbooks, medical literature, clinical practice guidelines, medical records and others, contain a large amount of medical and health knowledge. With the rapid and vigorous development of the medical and health sectors in China, a large amount of Chinese medical text data have been generated. The proper utilization of the information in these texts can facilitate

**Our code is available at https://github.com/Chang-Hongyang/Cas_CLN-BTCAMS intelligent development in the medical field, such as the construction of medical knowledge graphs and other research. However, unstructured text cannot be directly used by deep learning algorithms, and it is time-consuming and laborious to extract information manually. In this case, an important branch of natural language processing, joint entity and relation extraction, can be applied to complete the extraction of structured medical information in a low-cost and rapid way.

Overlap type	Input sentence (Chinese)	Relational triples (Chinese)	Input sentence (English)	Relational triples (English)
NOR	甲状腺彩超示：甲状腺左侧叶结节	<甲状腺彩超，检查证实了疾病，甲状腺左侧叶结节>	Ultrasonography of thyroid showed nodules in the left lobe of thyroid	<ultrasonography of thyroid, ECD, nodules in the left lobe of thyroid>
SEO	心脏彩超：左室舒张功能下降;双侧股总动脉斑块形成;	<心脏彩超，检查证实了症状，左室舒张功能下降> <心脏彩超，检查证实了症状，双侧股总动脉斑块形成>	Color doppler echocardiography: left ventricular diastolic function decreased and bilateral common femoral artery plaque formed;	<color doppler ECDio, ECD, left ventricular diastolic function decreased> <color doppler ECDio, ECD, bilateral common femoral artery plaque formed>
EPO	急性咽喉炎@麻疹:伴有结膜炎、鼻炎、咳嗽和特征性皮疹*科氏斑（颊黏膜红斑基底上出现蓝白色的突起病变）是麻疹的特征性表现。	<麻疹，并发症，咳嗽> <麻疹，并发症，鼻炎> <麻疹，并发症，结膜炎> <麻疹，临床表现，咳嗽> <麻疹，临床表现，鼻炎> <麻疹，临床表现，结膜炎> ...	Acute pharyngitis @ measles: With conjunctivitis, rhinitis, cough, and characteristic rash * Koenie's spot (a bluish white protruding lesion on the base of the erythema of the buccal mucosa) is characteristic for measles.	<measles, Comp, rhinitis> <measles, Comp, cough> <measles, Comp, conjunctivitis> <measles, CMa, rhinitis> <measles, CMa, cough> <measles, CMa, conjunctivitis> ...

Figure 1. Examples of triples of different entity overlap types. ECDio stands for "Echocardiography"; ECD stands for "The examination confirmed the disease"; Comp stands for "Complications"; and CMa stands for "Clinical Manifestation".

In these medical texts, there are a large number of relationships that cluster in sentences, such as one disease entity corresponding to multiple symptom entities, one examination entity corresponding to multiple symptom entities, etc., which are obviously characterized by a high density of triples, complex types, and diverse reference meanings of sentence elements. The entity overlap of relational triples is common in sentences. According to the degrees of entity overlap, sentences can be divided into three types: the normal type (NOR), single entity overlap type (SEO), and entity pair overlap type (EPO). If the triples do not share the same entity in a sentence, that is, there is no entity overlap, it is called the normal type (NOR); if there are two or more triples that share the same entity in a sentence, it is called the single entity overlap type (SEO); if a sentence contains two or more relations between one entity pair, it is called the entity pair overlap type (EPO). Figure 1 provides a more intuitive and detailed explanation of triple overlap. Complex overlap problems and the large number of triples bring major challenges to research on the joint entity and relation extraction of Chinese medical texts.

To solve the problems mentioned above, we propose a subject-based cascade tagging framework with conditional layer normalization (Cas-CLN) and a biaffine transformation-based cascade tagging framework with multi-head selection (BTCAMS) model. The Cas-CLN model divides the task into two parts: head entity decoding and relation-tail entity joint decoding. First, the head entity classifier detects all possible head entities in the multi-layer fusion of sentence representation. The model then deeply fuses the sentence representation with the head entity information and relation embedding information through the conditional layer normalization. The tail entity-relation joint decoder, which is composed of a multi-layer CBPTN network, decodes the tail entities in the fusion representation on the network layer corresponding to each relation. The advantages of using the encoder for the multi-layer fusion of sentence features and conditional layer normalization are (a) a multi-layer fusion of the encoders learns more comprehensive sentence encoding representations than using the last layer only; (b) conditional layer normalization is used to fuse the sentence representation with the head entity information to make the head entity information more accessible to the tail entity tagger. This improves the validity of the tail entity decoding. Although Cas-CLN is effective, when the number of predefined relations in the dataset is too large, the number of CBPTN layers will increase accordingly. In this case, there will be only a very small amount of entity pointer signals for most relations in the target output. The sparse and weakened supervision signals increase the difficulty of training. Due to the characteristics of medical texts, most datasets in the medical field are constructed manually, and the scale will be limited to a certain extent, which will affect the performance of Cas-CLN. To address this problem, we propose the BTCAMS model to divide the task into two parts: named entity recognition and relation extraction. BTCAMS uses a CBPTN to extract entities and entity types from sentence encoding representations, and then calculates possible relations between each entity pair by biaffine. We verified the effectiveness of the models in extracting relation triples on the two Chinese medical text datasets CMeIE [1] of CHIP2020 and CEMRDS.

Our main contributions can be summarized as follows:

- For Chinese medical text data, we proposed the Cas-CLN model based on CBPTN, which uses the multi-layer fusion representation mechanism and conditional layer normalization to improve the performance of the model.
- For datasets with a small scale or a large number of relation types, we proposed the BTCAMS model based on CBPTN, which enhances the relation determination between entity pairs through biaffine transformation.
- Experiments on two Chinese medical text datasets demonstrate the effectiveness of the two models that we proposed. In addition, experiments on the CMeIE dataset show that our models outperform the base models in all scenarios for the entity overlap and the multiple triples in sentence.

2. Related works

2.1. Entity relation extraction

Early research on relational triple extraction was based on the pipeline method [2–4], which divides the task into two subtasks: named entity recognition and relation classification. Such methods ignored the internal connections between the elements of the relational triples, resulting in the cumulative propagation of errors. In response to this problem, subsequent research has made progress using the joint extraction method based on feature engineering [5, 6] and the early application of neural networks [7, 8]. However, the methods based on feature engineering rely heavily on the manual construction of features and require a lot of manual labor. The early joint extraction methods simply shared the weights in the neural network, but they still decoded entities and relations independently.

In 2017, Zheng et al. [9] realized the joint extraction of triples by converting the relation extraction task into a sequence label. Since then, joint entity and relation extraction has developed rapidly, and a large number of joint extraction models have sprung up, such as the end-to-end model, which uses the copy strategy [10], the end-to-end model for fusion graph convolution neural networks [11], the Seq2Seq model, which introduces the reinforcement learning strategy [12], and the end-to-end model for multi-task learning with the copy mechanism [13]. Recently, Wei et al. [14] regarded the relation as the mapping function from the head entity to the tail entity and completed the task of the joint extraction of triples through a cascading binary tagging framework named CasREL. Additionally, some other excellent research, used a special handshake marker to reduce exposure bias [15] or employed a heterogeneous graph to fuse token and relation information [16].

2.2. Entity relation extraction in the medical field

The relation extraction task in the medical field is usually to identify the predefined types of relations between two medical entities. As early as 2011, Uzuner et al. [17] added the relation extraction of disease entities in electronic medical records. In the subsequent evaluations, the following tasks were added: the relation between the extraction of disease entities and time [18], the risk factors that may cause heart disease in the electronic medical record [19], and the relation between diseases and chemical drugs in the biomedical text (chemically induced disease relation extraction, CID) [20]. The following studies have been conducted in the relation extraction of medical texts. Yang et al. [21] used special rules and conditional random field (CRF) models to extract temporal relations from condition records; Sahu et al. [22] achieved the best result on the i2b2/VA relation extraction dataset [17] using the CNN model. Zhou et al. [23] proposed a framework based on the feature model and RNN neural network model to extract the relation between chemistry and diseases; Nguyen et al. [24] spliced the character encoding representation of CNN and LSTM as a CNN input to complete the medical relation extraction task; Chikka [25] aimed at the relation between diseases and treatments in the i2b2-2010 dataset and proposed a strategy based on the fusion of rules and Bi-LSTM; Ramamoorthy et al. [26] realized the extraction of adverse drug reaction relations through the question and answer format of reading comprehension tasks; Li et al. [27] integrated domain knowledge and an attention mechanism into the CNN model, which was improved in the CID task; and Zhou et al. [28] used the TransE model [29] to learn the knowledge representation of the dataset to guide the training of the CNN model and achieved the best results on the CID task.

3. Methods

3.1. Subject-based Cas-CLN

To solve these problems, we carried out a joint extraction model at the triple level, as shown in the following formulas. In the given sentence x_j of the training set D and the set of triples $T_j = (h, r, t)$ in the sentence, there may be entity overlap between the elements in T_j. During training, the task of the model is to maximize all x_j maximum likelihood estimates.

$$\begin{aligned}
&\sum_{j=1}^{|D|} \sum_{(h,r,t)\in T_j} \log p((h,r,t)|x_j) \\
&= \sum_{j=1}^{|D|} \Big[\sum_{h\in T_j} \log p(h|x_j) + \sum_{h\in T_j} \log p((r,t)|h,x_j) \Big] \\
&= \sum_{j=1}^{|D|} \Big[\sum_{h\in T_j} \log p(h|x_j) + \sum_{r\in T_j|h} \log p_r(t|h,x_j) + \sum_{r\in R\setminus T_j|h} \log p_r(t_\oslash|h,x_j) \Big]
\end{aligned} \tag{3.1}$$

where R is the set of predefined relations in the dataset, $r \in R$. $h \in T_j$ represents the head entity in the relation triple set; $p(h|x_j)$ represents the conditional probability that the head entity is h when the training sentence x_j is given; $p_r(t|h, x_j)$ represents the conditional probability of the tail entity t specific to the relationship r when the training sentence x_j and the head entity h are given; $r \in R \setminus T_j|h$ represents all the relations in the triple set T_j that have no semantic relationship with the head entity h. Since h has no semantic relationship with r, the tail entity is defined as $t_{\oslash}$.

In this formula, the relation is modeled as the function $t = r(h)$, where the head entity h is mapped to the tail entity t through the corresponding relation r, thereby avoiding the operation of relation classification. The task is disassembled into two independent modeling parts: head entity recognition $p(h|x_j)$ and relation-tail entity joint recognition $p_r(t|h, x_j)$. This modeling method alleviates the problem of multiple triples in sentences, especially with entity overlap.

According to the above method, we propose the CAS-CLN model. The processing of the model was described earlier in the Introduction. The overall structure of the Cas-CLN is shown in Figure 2, in which the input sample is "Heart failure can show signs of edema and pulmonary congestion." We can see that the head entity decoder decodes the encoding feature of the sentence to obtain three possible head entities: heart failure, pulmonary congestion, and edema. The first head entity, heart failure, is then selected as the condition for the relation-tail entity decoding. Two tail entities are decoded in the clinical manifestation relation to obtain the corresponding triples: <heart failure, clinical manifestation, edema > and <heart failure, clinical manifestation, pulmonary congestion>. It then continues to traverse other candidate head entities and repeat these operations.

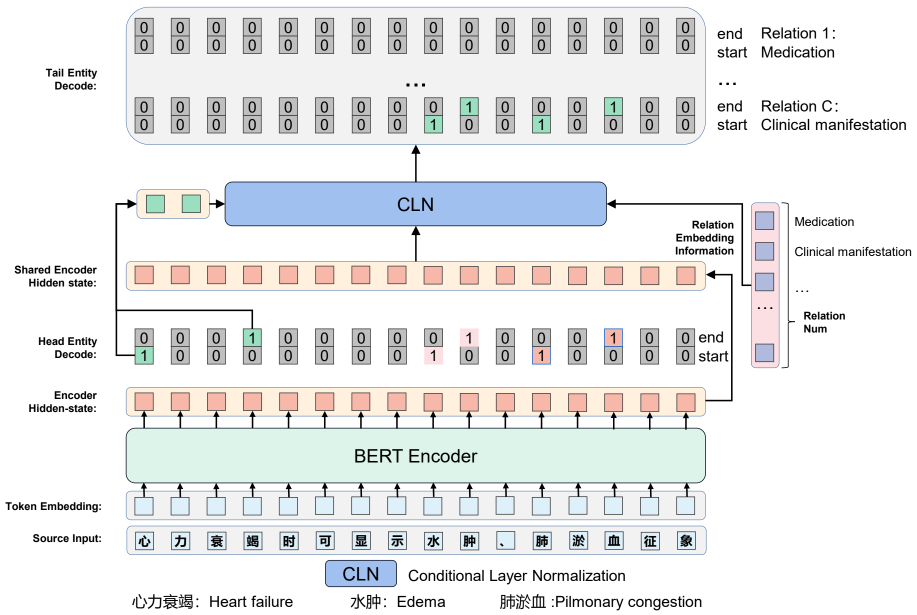

Figure 2. The cascade binary pointer framework fusing subject knowledge by conditional layer normalization.

3.1.1. Encoder

The encoder is used to extract the feature representation of the input sentence for the decoding of downstream modules. When choosing the encoder, we tried several mainstream pre-trained models, such as BERT [30], RoBERTa [31], and ERNIE [32], all of which are open-source Chinese versions.

BERT is a binary deep pre-trained language model. The pre-trained BERT model has been widely used in many NLP tasks owing to its rich prior knowledge learned from a large unlabeled corpus and its excellent in-depth bidirectional structure. Therefore, we do not elaborate on the pre-trained BERT model here.

RoBERTa optimizes BERT by 1) increasing the amount of pre-trained corpora, using a larger input batch size and more sufficient training for a longer time; 2) canceling the next sentence prediction pre-trained task in BERT; 3) dynamically selecting masked words in the training data. ERNIE has made three improvements in the training method: 1) changing the mask strategy during the pre-trained period to allow the model to learn the information of tokens, words, and entities through masking the token level, word level, and named entity level step by step; 2) using a large amount of multiple heterogeneous data for training; 3) introducing a dialogue language model to learn the semantic information of multiple rounds of dialogue in Baidu Tieba data.

3.1.2. Cascade binary classification decoder

1) Head entity decoding

Here, we adopt a binary pointer tagging network and use the sentence representation to calculate all head entities with probabilities greater than the set threshold. Binary pointer labeling refers to assigning a 0 or 1 mark to each token in the sentence and using two binary classifiers to detect the start and end positions of the head entity. This process can be explained in the following formula:

$$p_i^{start_h} = \sigma(W_{start}x_i + bias_{start}) \tag{3.2}$$

$$p_i^{end_h} = \sigma(W_{end}x_i + bias_{end}) \tag{3.3}$$

where x_i represents the feature code of the input $i-$th word; $W_{(.)}$and $bias_{(.)}$ represent the weight parameter matrix and bias of the model, respectively; σ represents the activation function; $p_i^{start_h}$and $p_i^{end_h}$ represent the probability that the $i-$th token can be used as the start position and the end position of the head entity, respectively. When the value is greater than the set threshold, the corresponding position is marked as 1; otherwise, it is marked as 0.

The process of identifying the span of the head entity in a sentence is optimized by the following function:

$$p_\theta(h|x) = \prod_{t\in(start_h,end_h)} \prod_{i=1}^{L} (p_i^t)^{I\{y_i^t=1\}}(1-p_i^t)^{I\{y_i^t=0\}} \tag{3.4}$$

where L represents the text length; $I\{u\}$ follows the rule: $I\{u\}$ is 1 when u is true, and 0 when u is false; $y_i^{start_h}$ and $y_i^{end_h}$ are, respectively, the start position and end position marks of the head entity containing the $i-$th word calculated by the formula, and their values are 0 or 1. $\theta = \{W_{start}, W_{end}, bias_{start}, bias_{end}\}$ are the parameters of the model.

2) Relation-tail entity joint decoding

This step is to decode the tail entities on the specific relation layer from the sentence representation. It can be seen from Figure 2 that the joint decoder is stacked by the CBPTN with the same number of layers as the number of predefined relations. To better fuse the relations embedding information, we add a weight parameter that can be learned and let the model independently choose the collection amount of relation information. In the information fusion, we choose the conditional layer normalization strategy that can make the information deeply interactive and fused.

Conditional layer normalization is mainly used to solve the problem that batch normalization leads to a decline in results when there is a small amount of training data in a single batch. Su et al. [33] generated the corresponding category text by adjusting positive and negative emotions as input conditions. Its application in this task is to find the corresponding relations and tail entities according to the known head entity.The specific LN formula is shown as follows:

$$u^l = \frac{1}{H}\sum_{i=1}^{H} a_i^l \tag{3.5}$$

$$\sigma^l = \sqrt{\frac{1}{H}\sum_{i=1}^{H}(a_i^l - u^l)^2} \tag{3.6}$$

$$x = Relu\left(\frac{g^l}{\sigma^l}\cdot(a_i^l - u^l) + b\right) \tag{3.7}$$

where l represents the l−th hidden layer; H represents the number of nodes in this hidden layer; and a represents the value of the a node before the activation function. g represents the trainable gain parameter matrix; b represents the trainable bias parameter; and x is the output of the hidden layer after being processed by the activation function, that is, the result after normalization.

To convert the LN strategy into our framework, we use the extracted head entities as conditions to assist the model in completing binary classification of the starting and ending positions of the tail entities. The detailed process is described in the following formula:

$$c_{rel} = w_{rel} * R \tag{3.8}$$

$$g^{'} = w_g * c + g \tag{3.9}$$

$$b^{'} = w_b * c + b \tag{3.10}$$

$$x^{'} = Relu\left(\frac{g^{'l}}{\sigma^l}\cdot(a_i^l - u^l) + b^{'}\right) \tag{3.11}$$

$$p_i^{start_t} = \sigma(W_{start}^r x_i^{'} + bias_{start}^r) \tag{3.12}$$

$$p_i^{end_t} = \sigma(W_{end}^r x_i^{'} + bias_{end}^r) \tag{3.13}$$

where w_g and w_b are linear transformation matrices. To obtain the encoding information of the candidate head entity and the fusing representation of relations, condition c has the same dimension as g and b, respectively, while c represents the fusion of relations and is expressed as Eq (3.8), R stands for the set of relations, w_{rel} is a trainable matrix, and the dimension is equal to the number of relations; $x^{'}$ represents the sentence feature representation after the fusion of head entity encoding information and relations embedding information; $p_i^{start_o}$ and $p_i^{end_o}$ are similar to those in the head entity decoding in Eqs (3.2) and (3.3).

The model performs relation-tail entity decoding for each candidate head entity and extracts the span of the tail entities that matches the head entity and the relations from the sentence. This process is optimized by the following formula:

$$p_r(t|x,h) = \prod_{t\in(start_t,end_t)}\prod_{i=1}^{L}(p_i^t)^{I\{y_i^t=1\}}(1-p_i^t)^{I\{y_i^t=0\}} \tag{3.14}$$

The parameter expressions in Eq (3.14) are the same as those in Eq (3.4).

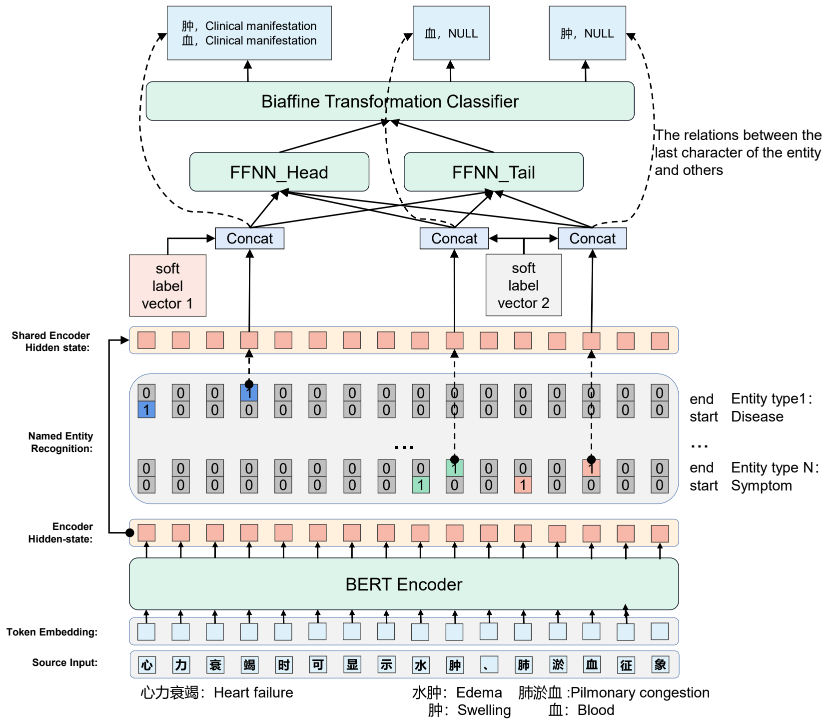

Figure 3. The biaffine transformation-based cascade tagging framework with multi-head selection.

3.2. *BTCAMS*

Cas-CLN uses the CBPTN with the same number of layers as the number of relations to solve the joint decoding of relations and tail entities, which reduces the average number of triples assigned to each layer. This weakens the supervised signal, increasing the difficulty of training. Therefore, Cas-CLN is more suitable for datasets with large scales or fewer predefined relation types. For this reason, we propose a biaffine transformation-based cascade tagging framework with multi-head selection (BTCAMS) model and disassemble the task into named entity recognition and relation classification. The BTCAMS model first extracts all entities and entity types from the sentence representation using CBPTN and then uses biaffine to calculate the relations between entity pairs after concatenating the entity and entity soft label. Here, we continue to choose the BERT pre-trained model for text feature extraction: 1) use the pointer labeling strategy to replace the conditional random field (CRF) model in the sequence labeling to achieve the extraction of nested entities; 2) add entity soft label vectors to strengthen the connection between named entity recognition and relation classification; 3) when the multi-head selection module judges the semantic information between two entities, the entity-encoded information is calculated by biaffine transformation to obtain the final relation matrix. The intuitive structure of the model is shown in Figure 3.

3.2.1. Pointer labeling framework

Nested entity recognition is a complex problem in named entity recognition tasks. In the traditional sequence labeling strategy, a multi-label classification task is used to replace multi-classification tasks, as shown in Figure 4(a). However, such processing will make each label of the entity isolated. Another approach is to merge the label layers, as shown in Figure 4(b). Merging the label layers and re-encoding the labels will result in a sharp increase in the number of labels and make some labels sparse. Thus, we continue to choose the cascading pointer annotation strategy introduced above to deal with these problems. Specifically, as shown in Figure 4(c), an R-layer pointer network is built, where R is the number of entity types in the dataset, and 1 is used to mark the start and end positions of the entities.

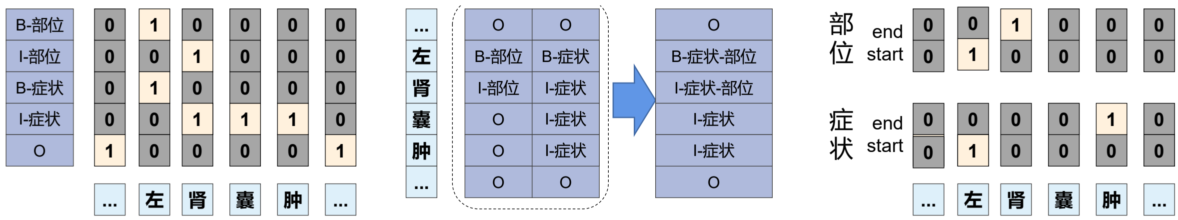

Figure 4. Different labeling schemes for nested entities.

3.2.2. Multiple selection module

Here, "head" refers to the last character of an entity, and multi-head selection means that each entity in the text may have predefined relations with other entities. The multi-head selection module and the named entity recognition module share weights with the encoder layer. The new entity feature is used as the entity head by concatenating the entity encoding feature with the entity type soft label, participating in the subsequent relationship classification task to calculate the relation types between other entities. The entity overlap problem of triples can be alleviated by this joint extraction method. The function of the module is to identify the predefined relation $\bar{r}_l \subseteq R$ between each entity ending in the character w_i, $i \in [0, n]$ and other entities $\bar{y}_l \subseteq w$ in the given sentence w and relation set R. The calculation of the relation score between any two entities w_i and w_j is shown in the following formula:

$$g_i = \frac{\sum softmax(s_i) \cdot M}{N} \tag{3.15}$$

$$z_i = [h_i; g_i], i = 0, ..., n \tag{3.16}$$

$$s(z_j, z_i, r_k) = VRelu(Uz_j + Wz_i + b) \tag{3.17}$$

where s_i is the state vector of the i-th character in the sequence; M is the vector matrix of the entity labels; N is the number of entity labels; g_i is the label representation vector learned by the model; h_i is the feature encoding representation of the character; z_i represents the spliced state vector of the label and character features; $V, b \in \mathbb{R}^l$, $U, W \in \mathbb{R}^{(l \cdot (2d+b))}$, d is the number of encoder hidden layer units; b is the label vector dimension; and l is the number of single-layer hidden layer units. The r_k represents the k-th relation type.

The definition of the probability formula for calculating the triple $< z_j, r_k, z_i >$ and the loss function in the relational calculation are shown in the following formula:

$$p_r(head = w_j, label = r_k | w_i) = sigmoid\Big(s(z_j, z_i, r_k)\Big) \tag{3.18}$$

$$L_{rel} = \sum_{i=0}^{n} \sum_{j=0}^{m} - \log p_r(head = y_i, relation = r_{i,j}|w_i) \tag{3.19}$$

where n is the length of the input text, m represents the number of triples composed of the last character w_i of the entity, $y_i \subseteq w$ is the last character of the other entities, and $r_i \subseteq R$ is the relation between the two entities.

3.2.3. Biaffine transformation attention

The improvements here are attributed to the methods proposed by Dozat et al. [34] and Yu et al. [35]. Dozat et al. introduced the biaffine attention mechanism into the dependency syntax analysis task to enhance the syntactic dependency between the dependent word and the head word. Yu et al. introduced this mechanism to the task of named entity recognition and judged the entity type by calculating the biaffine attention value of the character vector at the start and end of the entity. The biaffine attention mechanism uses the feedforward neural network (FFNN) to process the output of the feature-encoding layer to express h_i and adds the original linear deviation as the output result. The specific calculation process is shown in the following formula:

$$z_i^{'} = FFNN_{Head}(z_i) \tag{3.20}$$

$$z_j^{'} = FFNN_{Tail}(z_j) \tag{3.21}$$

$$s(z_i^{'}, z_j^{'}) = z_i^{'} U_m z_j + W_m(z_i^{'} \oplus z_j^{'}) + b_m \tag{3.22}$$

where $FFNN_{Head}$ and $FFNN_{Tail}$ represent two independent FFNNs; z_i and z_j are the stitching vectors of the feature layer encoding output and label representation vector in Eq (3.17); $z_i^{'}$ and $z_j^{'}$ represent the results of z_i and z_j after the dimension reduction processing of the FFNN to increase the proportion of the main features in the data. $U_m \in \mathbb{R}^{(d \times c \times d)}$, $W_m \in \mathbb{R}^{(2d \times c)}$; b_m is the bias parameter; d represents the number of hidden units in the FFNN; c represents the number of relations in the dataset.

Equation (3.22) is used to calculate the relation between the two entities in all their respective scores; the probability distribution of entity's last character w_i and another entity's last character w_j in all the semantic relations is calculated as shown in Eq (3.23); the relationship to pump loss function (cross entropy loss function) is defined as shown in Eq (3.24).

$$p_r(head = w_j|w_i) = SoftMax\Big(s_m(z_i^{'}, z_j^{'})\Big) \tag{3.23}$$

$$L_{rel} = \sum_{i=0}^{n} \sum_{j=0}^{m} - \log p_r(head = y_{i,j}, relation = r_{i,j}|w_i) \tag{3.24}$$

where n represents the length of the input text, m represents the number of the entity's last character w_i forming triples, y_i is the last character of the other entity, and r_i is the relation between the two entities.

4. Experiment

4.1. Datasets and evaluation metrics

We evaluate our model on two datasets, CMeIE [1] and CEMRDS. The CMeIE dataset is generated by manual construction. The corpus contains multi-source medical text data, including medical textbooks and clinical practice guidelines, totaling 28,008 sentences, 11 entity labels, and 44 relation labels. The CEMRDS dataset was constructed manually by ourselves. The data source includes electronic medical records of stroke and diabetes, which contains a total of 6,192 sentences, 7 entity labels and 14 relation labels. The two datasets consist of many sentences containing multiple triples. More importantly, the data sources are relatively broad, covering a representative sample of Chinese medical texts. Therefore, CMeIE and CEMRDS are suitable for evaluating the ability of the model to extract entity overlap triples from Chinese medical text data. We divide the sentence into normal (NOR), single entity overlap (SEO), and entity pair overlap (EPO) according to the different types of triple overlap in the sentences. In addition, we count the dataset according to the number of triples in the sentences. The detailed statistical results are shown in Table 1.

Table 1. Results of the CMeIE and CEMRDS dataset statistics. N is the number of relational triples in the sentence. The sentences are categorized into three types: NOR, SEO, and EPO. Note that a sentence can be divided into SEO and EPO at the same time. #Relations is the pre-defined number of relations of these two datasets, respectively; the other numbers are those of instances.

	CMeIE			CEMRDS		
Class	Train	Dev	Test	Train	Dev	Test
$N = 1$	6713	1663	2036	2322	301	283
$N = 2$	3711	962	1147	1037	128	121
$N = 3$	2304	583	699	499	49	63
$N = 4$	1635	396	494	294	45	42
$N \geq 5$	3561	878	1223	801	96	111
NOR	6931	1718	2116	2966	380	362
SEO	10,993	2764	3486	1987	239	258
EPO	1572	197	268	692	16	28
Total	17,924	4482	5602	4953	619	620
#Relations	44			14		

We report the precision rate (Prec.), recall rate (Rec.), and F1 value as the evaluation indexes of the model extraction effect. Only when the triple elements extracted by the model are completely consistent with the answer can the extracted triple be considered correct, that is, the exact matching method.

4.2. Compared methods

The baseline models we chose are 1) the state transition network Lattice LSTM-Trans model based on Lattice LSTM coding [36]; 2) CasREL. In CasREL, we used pre-trained models such as ERNIE, BERT, BERT-wwm, and RoBERTa-wwm to enhance performance.

4.3. Implementation details

Since the tasks are the same and the pre-trained models' structures are relatively similar, we used the RoBERTa-wwm pre-trained model that performs best in CasREL. To explore the impact of adding biaffine transformation, we conducted ablation experiments in BTCAMS, tested the model without a BT strategy, and denoted it as CAMS. We took into account the semantic relation of synonyms between similar entities to address the unique phenomenon in the CMeIE dataset. This phenomenon is more common in disease type entities. When a synonymous semantic relation appears in a sentence, we will no longer distinguish the order of the head and tail entities; when a entity has a synonymous semantic relation, we consider its dominant triple to be equivalent to the dominant triple of its synonym. Other experimental parameters are set as follows: the maximum text length of the model input is 300, the single input batch size during training is 12, the training epoch is 100, the Adam optimizer is used for model optimization, the learning rate is set to 5e-5, the output dimension of the encoding layer is 768, the word embedding and position embedding dimensions are set to 300, the position embedding window is 30, the hidden layer dimension is 150, and the drop rate is set to 0.5.

4.4. Results

4.4.1. Main result

Table 2 shows the results of different models for extracting semantic triples on two Chinese medical datasets. It can be seen from the table that the best results of CasREL were obtained when RoBERTa-wwm was uesd in CasREL's experiments. We speculate that this may be because RoBERTa-wwm can be more adequately trained by changing the mask strategy and increasing the training time and corpus size during pre-training. Therefore, we choose to use RoBERTa-wwm in the following experiments as the encoder layer. It can be seen from the results that our model exceeds the baseline model, except for the results of CAMS on CEMRDS. Among these, the result of Cas-CLN on the CMeIE dataset is lower than that of BTCAMS, while the result on the CEMRDS dataset is the best. By comparing the datasets, we can see that the CMeIE dataset has 44 predefined relations, while the CEMRDS dataset has only 14 predefined relations. The problem of sparse supervised signals in Cas-CLN training is improved, which confirms our speculation that Cas-CLN is more suitable for datasets with fewer predefined relation types. In addition, the comparison between the results of CAMS and BTCAMS shows that it is useful to integrate the calculation of deep biaffine transformation into the model.

Table 2. Results of the main experiment.

Setting	CMeIE			CEMRDS		
	Prec.(%)	Rec.(%)	F1(%)	Prec.(%)	Rec.(%)	F1(%)
Lattice LSTM-Trans	87.54	15.86	26.86	49.34	47.24	48.27
$CasREL_{ERNIE}$	56.78	50.76	53.60	67.79	64.95	66.34
$CasREL_{BERT}$	60.61	55.09	57.72	71.51	66.06	68.68
$CasREL_{BERT-wwm}$	60.80	55.02	57.76	70.05	67.58	68.79
$CasREL_{RoBERTa-wwm}$	60.45	56.57	58.44	74.82	63.94	68.95
Cas-CLN	65.40	53.90	59.09	73.73	68.89	71.23
Cas-CLN-Syn	61.09	58.18	59.60	-	-	-
CAMS	59.92	58.39	59.14	71.31	64.14	67.54
CAMS-Syn	60.43	58.63	59.52	-	-	-
BTCAMS	63.96	56.78	60.16	71.25	68.08	69.63
BTCAMS-Syn	64.51	57.08	60.57	-	-	-

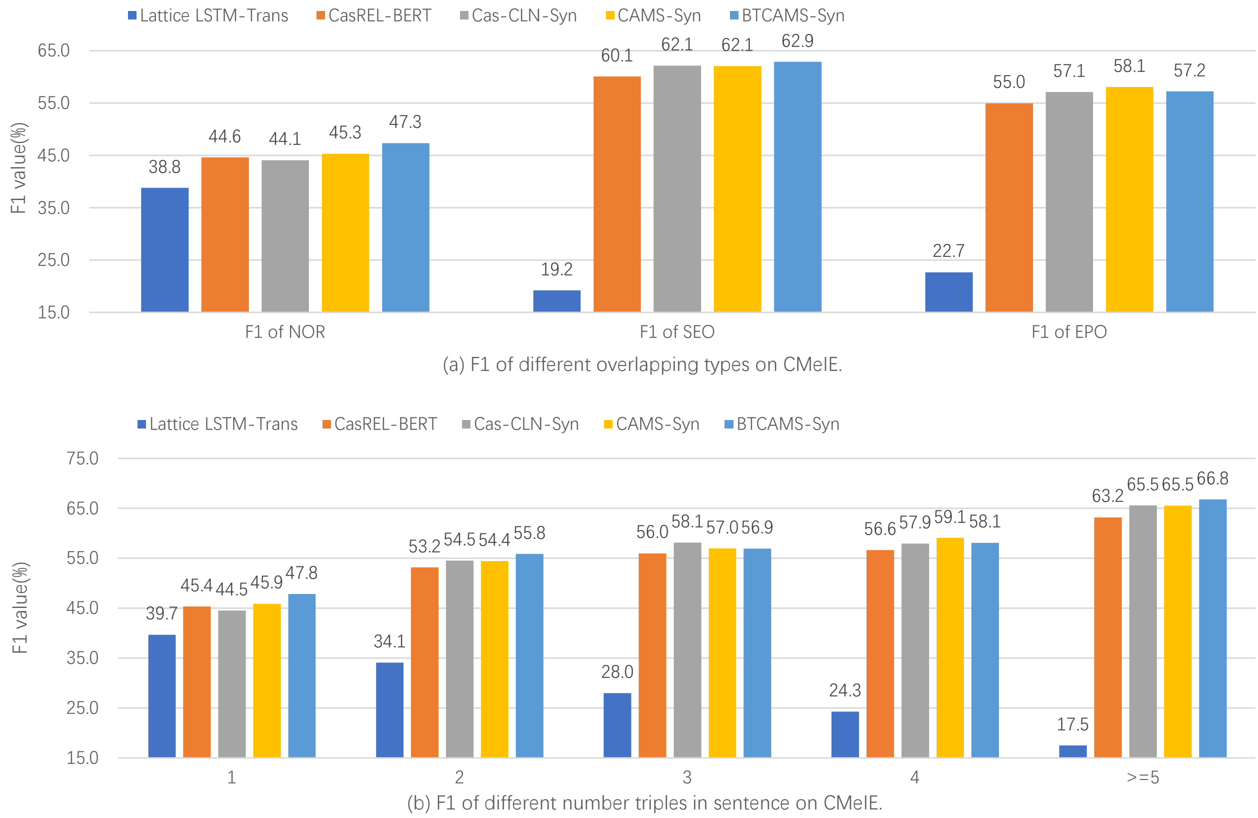

Figure 5. Detailed results of sentences with different overlap types and different numbers of triples on the CMeIE.

4.4.2. Results of different overlap types and different numbers of triple in sentence

We further test sentences containing different overlap types and different numbers of triples on the CMeIE dataset to verify the triple extraction ability of our model in dealing with complex cases. The results are shown in Figure 5. When dealing with different entity overlap types, the baseline model Lattice LSTM-Trans, in dealing with the two overlap problems of SEO and EPO, has much lower performance than the NOR type, which shows that the extraction of overlap relational triples is more difficult. However, the performance of our proposed models caught up with the baseline CasREL when dealing with various overlap problems, with a considerable number of models exceeding the baseline CasREL. Among them, CAMS and BTCAMS exceeded the baseline in all types, and even Cas-CLN was only 0.53% lower than CasREL in common types. This shows that our method is effective in dealing with complex overlap problems, whether single entity overlap or entity pair overlap.

When dealing with the problem of different numbers of triples in the sentence, it is similar to the situation of entity overlap. The more triples in the sentence, the more difficulties the model faces to extract them. Our proposed BTCAMS and CAMS completely exceed CasREL's performance. Cas-CLN is slightly lower than CasREL in sentences containing only one triplet type, and it also exceeds CasREL in other cases. This proves that when faced with sentences containing different numbers of triples, our models all show excellent and consistent performance.

5. Conclusions

In this paper, aiming at the extraction task of semantic relation triples in Chinese medical texts, we propose a subject-based cascade tagging framework with conditional layer normalization (Cas-CLN) and a biaffine transformation-based cascade tagging framework with multi-head selection (BTCAMS) model for the datasets with a wide variety of relations, and extensive experiments were conducted to verify the validity of the two models. In Cas-CLN, we used the head entity information auxiliary model to extract the tail entities and the corresponding semantic relations through the conditional layer normalization strategy. In BTCAMS, we improved the BIO entity labeling strategy through the cascade pointer label network and enhanced the extraction of semantic relations between two entities through biaffine attention. In conclusion, our methods have achieved better results than baseline CasREL on the two Chinese medical text datasets CMeIE and CEMRDS and experimental results on different sentence types show that our model can perform well in complex and difficult scenarios.

Acknowledgments

We thank the anonymous reviewers for their constructive comments and gratefully acknowledge the support of Zhengzhou collaborative innovation major special project (20XTZX11020).

Conflict of interest

The authors declare that there is no conflict of interest.

References

1. T. Guan, H. Zan, X. Zhou, H. Xu, K. Zhang, CMeIE: Construction and evaluation of Chinese medical information extraction dataset, in *Natural Language Processing and Chinese Computing* (eds. X. Zhu, M. Zhang, Y. Hong, R. He), Springer International Publishing, Cham, (2020), 270–282. https://doi.org/10.1007/978-3-030-60450-9_22
2. D. Zelenko, C. Aone, A. Richardella, Kernel methods for relation extraction, *J. Mach. Learn. Res.*, **3** (2003), 1083–1106. Available from: http://www.jmlr.org/papers/volume3/zelenko03a/zelenko03a.pdf.
3. G. Zhou, J. Su, J. Zhang, M. Zhang, Exploring various knowledge in relation extraction, in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, Association for Computational Linguistics, Ann Arbor, Michigan, (2005), 427–434. https://doi.org/10.3115/1219840.1219893
4. M. Mintz, S. Bills, R. Snow, D. Jurafsky, Distant supervision for relation extraction without labeled data, in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, Association for Computational Linguistics, Suntec, Singapore, (2009), 1003–1011. https://doi.org/10.1016/B978-0-12-374144-8.00264-2

5. X. Yu, W. Lam, Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach, in *Coling 2010: Posters*, Coling 2010 Organizing Committee, Beijing, China, (2010), 1399–1407. Available from: https://aclanthology.org/C10-2160.

6. M. Miwa, Y. Sasaki, Modeling joint entity and relation extraction with table representation, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Doha, Qatar, (2014), 1858–1869. https://doi.org/10.3115/v1/D14-1200

7. D. Zeng, K. Liu, S. Lai, G. Zhou, J. Zhao, Relation classification via convolutional deep neural network, in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, Dublin City University and Association for Computational Linguistics, Dublin, Ireland, (2014), 2335–2344. Available from: https://aclanthology.org/C14-1220.

8. M. Miwa, M. Bansal, End-to-end relation extraction using LSTMs on sequences and tree structures, in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Berlin, Germany, (2016), 1105–1116. https://doi.org/10.18653/v1/P16-1105

9. S. Zheng, F. Wang, H. Bao, Y. Hao, P. Zhou, B. Xu, Joint extraction of entities and relations based on a novel tagging scheme, in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Vancouver, Canada, (2017), 1227–1236. https://doi.org/10.18653/v1/P17-1113

10. X. Zeng, D. Zeng, S. He, K. Liu, J. Zhao, Extracting relational facts by an end-to-end neural model with copy mechanism, in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Melbourne, Australia, (2018), 506–514. https://doi.org/10.18653/v1/P18-1047

11. T. J. Fu, P. H. Li, W. Y. Ma, GraphRel: Modeling text as relational graphs for joint entity and relation extraction, in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Florence, Italy, (2019), 1409–1418. https://doi.org/10.18653/v1/P19-1136

12. X. Zeng, S. He, D. Zeng, K. Liu, S. Liu, J. Zhao, Learning the extraction order of multiple relational facts in a sentence with reinforcement learning, 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)*, Association for Computational Linguistics, Hong Kong, China, (2019), 367–377. https://doi.org/10.4337/9781786433787.00038

13. D. Zeng, H. Zhang, Q. Liu, CopyMTL: Copy mechanism for joint extraction of entities and relations with multi-task learning, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **34** (2020), 9507–9514. https://doi.org/10.1609/aaai.v34i05.6495

14. Z. Wei, J. Su, Y. Wang, Y. Tian, Y. Chang, A novel cascade binary tagging framework for relational triple extraction, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Online, (2020), 1476–1488. https://doi.org/10.18653/v1/2020.acl-main.136

15. Y. Wang, B. Yu, Y. Zhang, T. Liu, H. Zhu, L. Sun, TPLinker: Single-stage joint extraction of entities and relations through token pair linking, in *Proceedings of the 28th International Conference on Computational Linguistics*, International Committee on Computational Linguistics, Barcelona, Spain (Online), (2020), 1572–1582. https://doi.org/10.18653/v1/2020.coling-main.138
16. K. Zhao, H. Xu, Y. Cheng, X. Li, K. Gao, Representation iterative fusion based on heterogeneous graph neural network for joint entity and relation extraction, *Knowl. Based Syst.*, **219** (2021), 106888. https://doi.org/10.1016/j.knosys.2021.106888
17. Ö. Uzuner, B. R. South, S. Shen, S. L. DuVall, 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text, *J. Am. Med. Inf. Assoc.*, **18** (2011), 552–556. https://doi.org/10.1136/amiajnl-2011-000203
18. W. Sun, A. Rumshisky, Ö. Uzuner, Evaluating temporal relations in clinical text: 2012 i2b2 Challenge, *J. Am. Med. Inf. Assoc.*, **20** (2013), 806–813. https://doi.org/10.1136/amiajnl-2013-001628
19. A. Stubbs, Ö. Uzuner, Annotating risk factors for heart disease in clinical narratives for diabetic patients, *J. Biomed. Inf.*, **58** (2015), S78–S91. https://doi.org/10.1016/j.jbi.2015.05.009
20. C. H. Wei, Y. Peng, R. Leaman, A. P. Davis, C. J. Mattingly, J. Li, et al., Assessing the state of the art in biomedical relation extraction: Overview of the BioCreative V chemical-disease relation (CDR) task, *Database*, **2016** (2016), 1–8. https://doi.org/10.1093/database/baw032
21. Y. L. Yang, P. T. Lai, R. T. H. Tsai, A hybrid system for temporal relation extraction from discharge summaries, in *Technologies and Applications of Artificial Intelligence*, Springer International Publishing, Cham, (2014), 379–386. https://doi.org/10.1007/978-3-319-13987-6_35
22. S. Sahu, A. Anand, K. Oruganty, M. Gattu, Relation extraction from clinical texts using domain invariant convolutional neural network, in *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, Association for Computational Linguistics, Berlin, Germany, (2016), 206–215. https://doi.org/10.18653/v1/W16-2928
23. H. Zhou, H. Deng, L. Chen, Y. Yang, C. Jia, D. Huang, Exploiting syntactic and semantics information for chemicaldisease relation extraction, *Database*, **2016** (2016), https://doi.org/10.1093/database/baw048
24. D. Q. Nguyen, K. Verspoor, Convolutional neural networks for chemical-disease relation extraction are improved with character-based word embeddings, in *Proceedings of the BioNLP 2018 workshop*, Association for Computational Linguistics, Melbourne, Australia, (2018), 129–136. https://doi.org/10.18653/v1/W18-2314
25. V. R. Chikka, K. Karlapalem, A hybrid deep learning approach for medical relation extraction, preprint, arXiv:1806.11189.
26. S. Ramamoorthy, S. Murugan, An attentive sequence model for adverse drug event extraction from biomedical text, preprint, arXiv:1801.00625.
27. H. Li, Q. Chen, B. Tang, X. Wang, Chemical-induced disease extraction via convolutional neural networks with attention, in *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, IEEE Computer Society, Los Alamitos, CA, USA, (2017), 1276–1279. https://doi.org/10.1109/BIBM.2017.8217843

28. H. Zhou, C. Lang, Z. Liu, S. Ning, Y. Lin, L. Du, Knowledge-guided convolutional networks for chemical-disease relation extraction, *BMC Bioinf.*, **20** (2019). https://doi.org/10.1186/s12859-019-2873-7

29. A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., **26** (2013), 1–9. Available from: https://proceedings.neurips.cc/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf.

30. J. Devlin, M. W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, Minneapolis, Minnesota, (2019), 4171–4186. https://doi.org/10.18653/v1/N19-1423

31. Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, et al., RoBERTa: A robustly optimized BERT pretraining approach, preprint, arXiv:1907.11692.

32. Y. Sun, S. Wang, Y. Li, S. Feng, X. Chen, H. Zhang, et al., Ernie: Enhanced representation through knowledge integration, preprint, arXiv:1904.09223.

33. J. Su, Conditional text generation based on conditional layer normalization, Available from: https://kexue.fm/archives/7124.

34. T. Dozat, C. D. Manning, Deep biaffine attention for neural dependency parsing, preprint, arXiv:1611.01734.

35. J. Yu, B. Bohnet, M. Poesio, Named entity recognition as dependency parsing, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Online, (2020), 6470–6476. https://doi.org/10.18653/v1/2020.acl-main.577

36. Y. Liu, *Research on Automatic Extraction of Chinese Named Entities and Entity relations*, MSE thesis, Zhengzhou University, 2019.

AIMS Press

© 2022 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)