Journal of Network Intelligence ©2025 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **10**, Number **3**, August **2025**

Joint Entity Relation Extraction for Traditional Chinese Medicine Formula Data

Lin-Ying Chen, Zhong-Mei Zhou*, Rong-Guo Wang, Wei Chen

School of Technology
Fuzhou Technology and Business University, Fuzhou 350715, China
1191949284@qq.com, 464523040@qq.com, 274600845@qq.com, 470041595@qq.com

Jian-Hua Liu

School of Computer Science and Mathematics
Fujian University of Technology, Fuzhou 350118, China
jhliu@fjnu.edu.cn

*Corresponding author: Zhong-Mei Zhou

Received February 12, 2025, revised April 7, 2025, accepted June 21, 2025.

Abstract. *Entity Relation Extraction aims to extract named entities and their relationship from unstructured text, but often ignores the intrinsic connection between the entities and relationship, leading to error propagation. Meanwhile, the textual description structure of traditional chinese medicine formula data is special, which needs to consider the complexity of entity relation nesting and overlapping, etc. This paper proposes a Joint Entity Relation Extraction for Traditional Chinese Medicine Formula Data (JERE-TCMFD) model. Firstly, to be able to learn tag semantic information at a deeper level, the model integrates syntactic annotation tag features into an attention-based relational graph convolutional network, which strengthens the feature representation of grammatical relations. Secondly, this paper explicitly integrates annotation labeling features into the textual representation to further enhance the joint modeling of entities and relationships. Thirdly, entity relation extraction is transformed into a unified word-pair tagging task using a grid tagging scheme, and entity relation triples are extracted by a unique decoding approach. Finally, the model is applied to extract entity relations from the Chinese medicine formula corpus. Its experimental results show that the F1-score of JERE-TCMFD model can reach 84.57%, which is 8.75% higher than the traditional pipeline model BiLSTM-CRF+LSTM.*

Keywords: Traditional chinese medicine formula, Joint extraction, Entity recognition, Relation extraction

1. **Introduction.** The entity relation extraction task aims to extract structured information from natural language text and it is an important sub-task of information extraction. [1]. Approaches to solving the task of entity-relation extraction can be categorized into two main types: pipeline methods and entity-relation joint extraction methods. Early on, researchers focused on pipelined approaches by dividing the entity relation extraction task into Named Entity Recognition (NER) [2] and Relation Extraction (RE) [3] subtasks, i.e., given a piece of semi-structured or unstructured text, the entities in the text are first extracted by NER, and then the relationships are classified for each candidate entity pair. Zhong et al. [4] implemented entity recognition and relation extraction using a simple pipelined model using two independent encoders, the input for relation extraction is simply the result of entity recognition, and experimental results show that introducing entity category information leads to an improvement in relation modeling. Zhao et al. [5] proposed a knowledge-guided distance supervised (KGDS) model to deal with biomedical relation extraction in Chinese electronic medical records. Sousa et al. [6] integrated a knowledge graph-based recommendation model into biomedical RE to improve its usefulness further.

However, the pipeline approach ignores additional joint features between entities and relations, which leads to three problems: (1) errors in early steps cannot be corrected in subsequent steps [7], leading to error propagation; (2) this approach leaves insufficient interactions between entity pairs and relations within the ternary structure [8]; (3) this approach generates a large amount of redundant information.

Hence, some researchers have optimized the overall performance of the model by building a unified model that interactively uses information from the two tasks of entity recognition and relation extraction, fully utilizing the correlation between them. Miwa et al. [9] used an end-to-end neural network model to obtain extract entities and their relations for textual data. Hang et al. [10] first labeled entities and their relations based on a multi-label labeling scheme, and then used a joint entity and relation extraction model with a multi-layered attention mechanism to extract triples from sentences.Finally, the relationship alignment model is used to align the predicted relationship classifications. Zheng et al. [11] were the first to transform joint entity and relation extraction into a sequence labeling task, proposed a novel joint extraction method based on sequence labeling, and designed an end-to-end model with a biased loss function to achieve true joint entity and relation extraction. Xu et al. [12] proposed a joint extraction framework that utilizes Graph Convolutional Networks (GCN) to jointly extract entity relations from document-level Chinese medicine texts. Wu et al. [13] proposed the Grid Tagging Scheme (GTS), which transforms triple extraction into a unified word-pair tagging task and extracts triples through a unique decoding scheme. Xu et al. [14] proposed the Span-ASTE model, which employs a spanning approach to consider the interaction between text segments of entity 1 and entity 2 and reduces training time with a two-channel span pruning strategy. Lv et al. [15] utilized a global pointer network and multiple decoders to extract entity relations jointly to address the issue of overlapping relations in Traditional Chinese Medicine (TCM) texts. However, these unified extraction methods focus on different integration algorithms, ignoring the fact that they are essentially based on distinct tag spaces. The tag name is derived from the model's category generalization and represents the type of word that occurs in the sentence and is semantically related to other words in the text, making this association exploitable. However, existing studies have generally adopted oversimplified tagging representations of word information, leading to incomplete or missing tag semantics.

Text	The Chinese goldthread antidote soup is composed of multiple traditional Chinese medicines such as Baikal Skullcap and Chinese Goldthread	
Entity1	Chinese goldthread antidote soup	
Entity2	Baikal Skullcap	Chinese Goldthread
Relation	(Chinese goldthread antidote soup,Baikal Skullcap,formula/Chinese medicines) (Chinese goldthread antidote soup,Chinese Goldthread, formula/Chinese medicines)	

Figure 1. Example of TCM formulary entity relation extraction

Entity relation extraction in TCM formulary texts aims to identify entities such as formula names, Chinese medicine names, syndrome types, symptoms, and causes of disease from formulary texts and extract relationship categories between entities, such as formula/Chinese medicine, formula/symptom type, symptom type/symptom, and cause of disease/symptom type. As shown in Figure 1, in the given sentence "The Chinese Goldthread Antidote Soup is composed of multiple traditional Chinese medicines, such as Baikal Skullcap and Chinese Goldthread," it contains two formula entity relation triples: (Chinese goldthread antidote soup, Baikal Skullcap, formula/Chinese medicine) and (Chinese goldthread antidote soup, Chinese Goldthread, formula/Chinese medicine). This knowledge not only helps gain a deeper understanding of the therapeutic principles of formulae, but also supports computer-aided Chinese medicine diagnostic and treatment systems, to guide the accurate diagnosis, effective prevention, and treatment of diseases. been fewer studies on entity relation extraction in the TCM domain [16]. Building on the approach of Zheng et al. [11], Cao et al. [17] improved the labeling strategy and employed the Bi-directional Long Short-Term Memory Conditional Random Fields (BiLSTM-CRF) model to address the issue of overlapping relationships where the same entity participates in multiple relationships, which achieved better results on the drug entity relationship dataset in the biomedical field. Yu et al. [18] introduced a Category-BERT BiLSTM Sigmoid (CBBS), which frames the relationship extraction problem in the medical domain as a sequence annotation problem and enhances extraction performance when the same entity appears in multiple relationships within a single corpus. Additionally, due to the unique text structure in TCM formula entity relation extraction tasks, issues such as nested entities and overlapping relationships arise. Yang et al. [19] proposed a joint extraction of TCM entity relation based on word vector splicing. Firstly, word vector splicing is used as input. Secondly, an improved sequence annotation strategy is used to label the Chinese medicine text on BiLSTM-CRF model. Finally, the relationship triples are extracted by customized extraction rules. However, the limited availability of publicly available datasets in the field of TCM formulas restricts the progress of related research.

Contribution. Joint Entity Relation Extraction for Traditional Chinese Medicine Formula Data (JERE-TCMFD) model. The JERE-TCMFD model independently encodes grid-annotated and syntactic-annotated tag semantic features using BERT, so that the encoding captures deeper and richer semantic features, and explicitly integrates these features into the textual representations to fully explore the tag's semantic properties. The contribution of this paper can be summed up as the following three points:

(1) For the task of joint entity relation extraction of TCM formula texts, the JERE-TCMFD model is proposed to leverage tag knowledge for facilitating triple extraction, which independently encodes text and tag annotations and explicitly integrates the semantic information of the labels into the textual representation.

(2) This paper proposes to integrate annotation tag syntactic features into an attention-based relational graph convolutional network to further enhance the feature representation of syntactic relationships.

(3) In view of the small number of publicly available entity-relationship extraction datasets in the field of TCM formulas, this paper creates a formulas dataset called TCMF, and adopts manual annotation to process the formulas data.

2. **Related work.** In this paper, the model uses the Attention-based Relational Graph Convolutional Network (ARGCN) [20] to extract syntactic dependency features, while the TCM formula entity relation triples are extracted by a Grid Tagging Scheme (GTS) [13]. Therefore, this section focuses on the fundamentals of ARGCN and GTS.

2.1. **ARGCN. (1)Syntactic Dependency Tree.** ARGCN primarily utilizes syntactic dependency trees to learn the contextual representation of grammatical structures. The structural information in dependency trees is a method for abstracting complex relationships and data structures into a graph structure, and ARGCN defines the dependency tree structure as a directed graph. As shown in Figure 2, for the sentence "Rhizoma Coptidis is one of the components of Rhizoma Coptidis Antidote Soup", the original dependency tree is generated by the spaCy parser. Jiang et al. [20] set a threshold D; if the relative distance between words in a sentence is less than this threshold, some additional edges are added, with their type set to "other," and relative distance features are added to all edges. For example, in Figure 3, with "components" as the center and the threshold set to 2, all the words within the relative distance of "components" will be connected to it through edges.

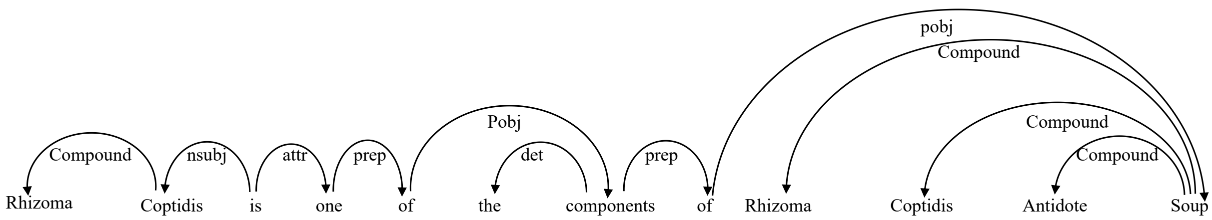

Figure 2. Example of a syntactic dependency tree for ARGCN.

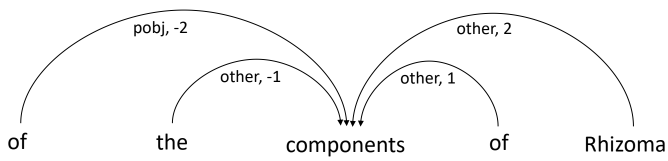

Figure 3. Example of a partial dependency graph with a threshold D = 2

The syntactic dependency tree of ARGCN can be simply expressed as $G = (V, E, R, P)$, where $V = \{V_i\}_i^n$ is the set of nodes, $E = \{e_{ij}\}_{i,j=1}^n$ is the set of edge relations, e_{ij} indicates whether node i to node

j is related or not, $R = \{r_{ij}\}_{i,j=1}^{n}$ is the set of edge relation types, r_{ij} indicates the type of relation from node i to node j, $P = \{p_{ij}\}_{i,j=1}^{n}$ denotes the set of distances from node i to the relative position of node j. In addition, ARGCN inverts the edge types nsubj or dobj in order to link to the target word, and removes root relations that have a self-loop structure.

(2) ARGCN model principle. ARGCN is an extension of Relational Graph Convolutional Neural Network (R-GCN) [21], which is specialized for processing graph data with multiple relationships, dependency tree graph structure of R-GCN can be represented as $G = (V, E, R)$. By incorporating the relation matrix and the relation-aware convolution operation, R-GCN can flexibly manage the heterogeneity among different relations. The node update process can be expressed by Equations (1) and (2):

$$h_i^{'} = \sigma(\sum_{r\in R}\sum_{j\in N_i^r} x_{i,j,r} + W_0 h_i) \tag{1}$$

$$x_{i,j,r} = \sigma(\sum_{r\in R}\sum_{j\in N_i^r} \frac{1}{c_{i,r}} W_r h_j) \tag{2}$$

Where h_i is the input representation of node i, $h_i^{'}$ is the output representation of node i, R is the set of edge relation types, N_i^r is the set of neighboring nodes of node i under relation $r \in R$, W_r and W_0 are trainable matrices and $c_{i,r}$ is a problem-specific normalization constant, which is usually the number of neighbors of node i under relation r. In addition, R-GCN reduces the computational cost by base decomposition. The mathematical representation is shown in Equation (3):

$$W_r = b_r W_1 \tag{3}$$

where W_1 is the basis for W_r and b_r is used to adjust the weights for different relationship types.

To incorporate additional distance information, ARGCN, an extension of RGCN, introduces a distance-aware attention mechanism to acquire relative distance features between words. Firstly, the two node features h_i and h_j are multiplied by the same matrix W_2 to obtain the query and key values respectively, while the relative position encoding p is obtained by the sinusoidal encoding matrix [22]. Secondly, the shared attention mechanism computes attention based on the query and key values, as well as the relative position encoding, and the results are normalized to yield the attention coefficient β_{ij} between the two nodes. Finally, the effect of the attention coefficient and the relationship type is modulated by the matrix W_c. The node update mathematical representations for ARGCN are shown in the following Equations (4) to (7):

$$o_{ij} = \sigma(a^T[W_2 h_j, W_2 h_i, p]) \tag{4}$$

$$\beta_{ij} = \frac{exp(o_{ij})}{\sum\limits_{k\in N_i} exp(o_{ik})} \tag{5}$$

$$\alpha_{ij,r} = \sigma(W_c^T[b_r, \beta_{ij}]) \tag{6}$$

$$x_{i,j,r} = \alpha_{ij,r} W_1 h_j \tag{7}$$

where both W_c and a are trainable matrices and σ is an activation function ReLU.

This paper uses the multi-head attention mechanism of ARGCN to capture diverse relational features among nodes by computing multiple attention heads in parallel. Each attention head independently computes attention based on query values, key values, and relative position encodings, and generates corresponding attention coefficients. Subsequently, the outputs of multiple attention heads are concatenated or aggregated to integrate information from different heads, enhancing the model's capacity to represent complex graph structures. The computation process of ARGCN's multi-head attention mechanism is given in Equations (8) and (9):

$$x_{i,j,r}^k = \alpha_{ij,r}^k W_1^k h_j \tag{8}$$

$$x_{i,j,r} = W_d[x_{i,j,r}^1, x_{i,j,r}^2, ..., x_{i,j,r}^k] \tag{9}$$

where K denotes the number of head attention and $x_{i,j,r}^k$ denotes the output of the kth head attention. Both W_1^k and W_d are trainable matrices. Additionally, to mitigate issues like gradient vanishing and explosion, residual connections are incorporated into each ARGCN layer, as shown in Equation (10):

$$h_i^{l+1} = h_i^l + h_i^{'l} \tag{10}$$

where h_i^l denotes the input representation of node i in layer l of ARGCN, $h_i^{'l}$ denotes the output representation of node i in layer l.

2.2. **Grid tagging scheme.** GTS is a commonly used method for addressing pipelined problems in the Aspect Sentiment Triple Extraction (ASTE) task. The grid tagging scheme incorporates all sentiment factors of a sentence into a unified grid tagging task by labeling the relationships between all word pairs, and then generates aspect sentiment triples using a specific decoding algorithm.

(1)Word Pair Marker. In the ASTE task, GTS uses the tag set $\{A, O, POS, NEG, NEU, N\}$ to label all word pairs (w_i, w_j) in a sentence. The label A indicates that the word pairs consists of the same aspect words, the label O indicates that the word pairs consists of the same opinion words, and the labels POS, NEG, and NEU indicate the sentiment polarity of the word pairs (w_i, w_j), corresponding to positive, negative, and neutral, respectively, the label N denotes other word pairs. Since word pairs are unordered, an upper triangular grid is used to label the relationships between word pairs.

However, the tagging method considers each word in an aspect or opinion item to be of equal importance, which hinders the model's ability to find all aspect and opinion items consisting of multiple words. Therefore, Xia et al. [23] proposed an eight-tag labeling $\{A, AR, O, OR, POS, NEG, NEU, N\}$ based on GTS, where AR denotes the relationship between non-terminal words in aspect words and their terminators, and OR denotes the relationship between non-terminal words in opinion words and their terminators. Xia et al. considered that terminating word can denote the entire aspect or opinion words. When tagging aspects or opinions, focus only on the relationship between the word terminal and the non-terminal word. When tagging sentiment polarity, focus only on the relationship between the terminal word of the opinion word and all the words in the aspect word.

(2)Specific decoding strategie. After obtaining the sentence prediction labeling results, which are decoded by a specific decoding strategie. First, the tags predicted for all word pairs (w_i, w_j), on the main diagonal are used to find labels A or O for aspect and opinion words, followed by an upward search for consecutive tags AR or OR that form the complete aspect a or opinion word o. When $w_i \in a$ and $w_j \in o$, the sentiment tags predicted for the word pair are computed, and the sentiment polarity $s \in \{POS, NEU, NEG\}$ with the highest number of occurrences is selected to form the triple $\{a, o, s\}$. If there is a tie in the number of tags, s is determined in the order of $POS > NEU > NEG$. a and o are considered to fail to form a triple if the tags are all predicted to be labeled N.

This paper refers to the method of Xia et al. [23] and uses an nine-tag word-pair labeling scheme to uniformly extract TCM formula entity relation triples. The grid tag set is defined as $\{E1, ER1, E2, ER2, CON, TRE, PER, CAU, N\}$, where $E1$ represents the word terminal at the head entity, $E2$ represents the word terminal at the tail entity, and $ER1$ denotes the relationship between the non-terminal words in the head entity and the terminal word of the head entity. $ER2$ represents the relationship between the non-terminal words in the tail entity and the terminal word of the tail entity. The four labels CON, TRE, PER, and CAU represent the relationships between entity pairs (w_i, w_j), which correspond to formula/Chinese medicine, formula/symptom type, symptom type/symptom, and cause of disease/symptom type, respectively. The nine-tag word-pair labeling scheme for TCM formula data is illustrated in Figure 4.

Rhizoma	Coptidis	is	one	of	the	components	of	Rhizoma	Coptidis	Antidote	Soup	
N	ER1	N	N	N	N	N	N	CON	CON	CON	CON	Rhizoma
	E1	N	N	N	N	N	N	CON	CON	CON	CON	Coptidis
		N	N	N	N	N	N	N	N	N	N	is
			N	N	N	N	N	N	N	N	N	one
				N	N	N	N	N	N	N	N	of
					N	N	N	N	N	N	N	the
						N	N	N	N	N	N	components
							N	N	N	N	N	of
								N	N	N	ER2	Rhizoma
									N	N	ER2	Coptidis
										N	ER2	Antidote
											E2	Soup

FIGURE 4. Example of nine-tagged word-pair labeling scheme for TCM formula data

3. **Joint Entity Relation Extraction for Traditional Chinese Medicine Formula Data.** The JERE-TCMFD model is innovatively proposed to address the absence of semantic labeling information.

On the one hand, the model enhances the grammatical relation features between words by acquiring the semantic information features of grammatical relation type labels when learning grammatical information features. On the other hand, the model fuses the semantic features of the grid tagged labels with the grammatical features by dot product in order to improve the accuracy rate of grid tagging. The JERE-TCMFD model consists of five main parts which are the input layer, encoding layer, syntactic feature learning layer, grid tagging scheme layer and output layer. The model structure is shown in Figure 5.

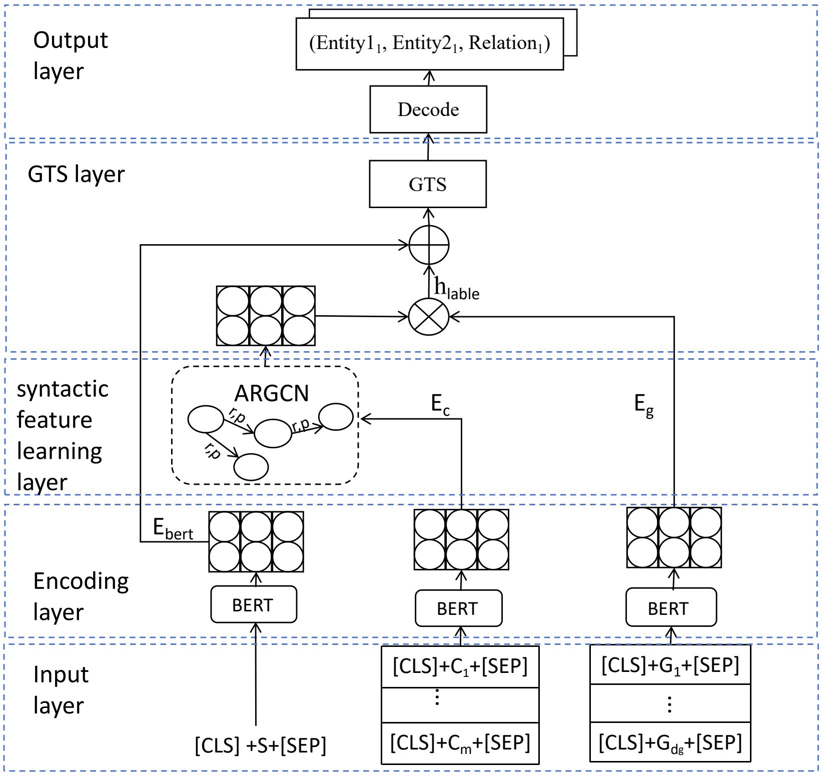

FIGURE 5. JERE-TCMFD model architecture

3.1. **Task Definition.** The JERE-TCMFD model aims to extract all entity relation triples $T = \{t_1, t_2, ..., t_k\}$ in a sentence $S = \{w_1, w_2, ..., w_n\}$ consisting of n words, where $t_i = \{f_i, l_i, r_i\}$, with f_i and l_i representing the head and last entities of the i-th triple, respectively, and r_i denotes the corresponding relation types. There are k formula entity relation triples in the sentence.

3.2. **Input Layer.** The input to the model consists of a sentence, a set of syntactic annotation tags, and a set of grid annotation tags. The set of syntactic annotation tags $C = \{C_1, C_2, ..., C_m\}$ is a specific phrase annotation designed for the set of relation type tags of the syntactic dependency tree, which can adequately represent the specific meaning of the corresponding syntactic relation type tags. The set of grid annotation tags $G = \{G_1, G_2, ..., G_{dg}\}$ is a specific phrase annotation designed for the set of labeled tags $\{E1, ER1, E2, ER2, CON, TRE, PER, CAU, N\}$ of a grid, which can adequately represent the specific meaning of the corresponding grid tag.

3.3. **Encoding Layer.** Word embedding is a crucial foundation for transforming textual symbols into computable values, and the Bidirectional Encoder Representation from Transformers (BERT) pre-trained language model is a word embedding model based on the Transformer architecture [24]. BERT learns a large amount of textual data through pre-training and captures the semantic relationships between words in the process. Therefore, this paper focuses on extracting semantic features using BERT to more accurately capture word contextual relevance, thereby obtaining a richer representation of semantic features. Specifically, this paper primarily employs three BERT encoders to extract semantic features from sentences, syntactic annotation tags, and grid annotation tags, respectively.

(1)Semantic Feature Extraction for Sentences. When extracting semantic features of a sentence, the use of a BERT encoder integrates the contextual features of the sentence into the word embedding representation. First, the sentence is reconstructed as "$[CLS] + s + [SEP]$" to conform to BERT's input format. It is then fed into the BERT model to obtain the contextual semantic feature representation $\{e_0, e_1, ..., e_n, e_{n+1}\}$ of the sentence, where e_0 and e_{n+1} denote the feature vectors of the special tokens $[CLS]$ and $[SEP]$ respectively. Then vectors $\{e_1, ..., e_n\}$ are collectively denoted as $E_{bert} \in R^{n \times d}$, which represents the semantic feature matrix of the sentence S. Here, n denotes the length of the sentence, and d denotes the word embedding feature dimension. This paper uses E_{bert} as the word embedding output of BERT.

Table 1. Syntactic annotation tags conversion table

No	Simple tag	syntactic annotation tag
1	adj	have adjective
2	advmod	have adverbial modifier
3	csubj	have clausal subject
4	csubjpass	have clausal passive subject
5	pobj	have object of preposition
6	nummod	have numeric modifier
7	pcomp	have prepositional complement
8	prt	have particle
9	nsubj	have nominal subject
10	nmod	have nominal modifier
11	acomp	have adjectival complement
12	predet	have pre-determiner
13	npadvmod	have noun phrase as adverbial modifier
14	agent	have agent
15	aux	have auxiliary
16	ccomp	have clausal complement
17	preconj	have preconjunct
18	dative	have have dative
19	auxpass	have auxiliary in passive construction
20	amod	have adjectival modifier
21	advcl	have adverbial clause modifier
22	prep	have prepositional modifier
23	compound	have compound
24	poss	have possession modifier
25	intj	have interjection
26	conj	have conjunct
27	attr	have attribute
28	neg	have negation modifier
29	nsubjpass	have nominal passive subject
30	punct	have punctuation
32	acl	have clausal modifier of noun
33	quantmod	have quantifier modifier
34	relcl	have relative clause modifier
35	mark	have marker
36	dobj	have direct object
37	appos	have appositional modifier
38	case	have case marking
39	expl	have expletive
40	xcomp	have open clausal complement
41	oprd	have object predicate
42	dep	have dependency
43	det	have determiner
44	parataxis	have parataxis

(2)Semantic Features Extraction for Syntactic Annotation Tags. In the syntactic dependency tree, the types of syntactic relationships between nodes are labeled by syntactic relationship labels. Previous studies usually use simple syntactic tagging feature representations, limiting the model's ability to capture complex syntactic relationships between nodes. However, due to the special structure of

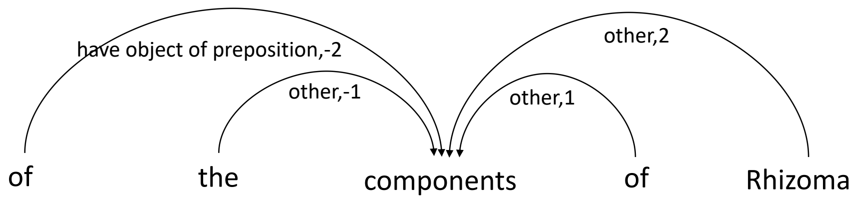

Figure 6. Syntax Dependency Tree Fragment for Replacing Simple Tags with Syntactic Annotation Tags

ARGCN, it can take any form of textual representation as input for syntactic relation types. Therefore, leveraging this property of ARGCN, this paper incorporates syntactic annotation tag features into the model to enhance its capability in jointly modeling syntactic structures and semantic information. Firstly, the sentence is parsed into a syntactic dependency tree by the spaCy parser, and all simple syntactic tags in the tree are converted into syntactic annotation tags. For example, as shown in Figure 6, the tag "pobj" is annotated as "have object of preposition", thus obtaining the set of syntax annotation tags C for the input sentence. The conversion of simple syntactic tags to syntactic annotation tags is shown as follows in Table 1. Secondly, each syntactic annotation tag is formatted as BERT input and fed into the model as "$[CLS]+C_i+[SEP]$", so as to obtain the semantic feature representation $\{c_1^i,...,c_h^i\}$ of the i-th syntactic annotation tag. Thirdly, the compressed syntactic annotation tag representation is obtained by weighted summation of its features. Finally, all syntax annotation tag features are concatenated, forming the syntactic relation type feature input for the ARGCN model. The calculation process is as follows in Equations (11) and (12):

$$syn_i = \frac{\sum_{j=1}^{h} c_j^i}{h} \tag{11}$$

$$E_{syn} = [syn_1, syn_2, ..., syn_m] \tag{12}$$

where h is the length of the i-th syntactic annotation tag, $syn_i \in R^{1\times d}$ denotes the i-th syntactic annotation tag representation obtained by weighted summation compression and m denotes the number of syntactic tags of the input sentence.

Table 2. Grid annotation tags conversion table

No	Simple entity relation tag	entity relation annotation tag
1	$E1$	Compose head entity
2	$ER1$	Compose head entity
3	$E2$	Compose tail entity
4	$ER2$	Compose tail entity
5	CON	judge contain
6	TRE	judge treatment
7	PER	judge performance
8	CAU	judge cause of disease
9	N	None

(3)Semantic Features Extraction for Grid Annotation Tags. Grid annotation tags aim to enhance sentence feature representation through tag semantics. For this reason, in this paper, annotation tags containing sufficient semantics are designed for each relation type and its corresponding entity tag, and the specific grid annotation tags are shown in Table 2.

In extracting the semantic features of the tags, the feature representation of each annotated tag $\{g_1^i,...,g_l^i\}$ is encoded by the BERT model. Next, the i-th grid annotation tag representation $gte_i \in R^{1\times d}$ is weighted and compressed, and then all grid annotation tag representations are concatenated to obtain $E_{gte} \in R^{dg\times d}$, dg is the number of grid labeling tags. The calculation process is as follows in Equations (13) and (14):

$$gte_i = \frac{\sum_{j=1}^{L} g_j^i}{L} \tag{13}$$

$$E_{gte} = [gte_1, gte_2, ..., gte_{dg}] \tag{14}$$

where L is the length of the i-th grid annotation tag.

3.4. **Syntactic Feature Learning Layer.** Once the sentence feature representation E_{bert} and the syntactic annotation tag feature representation E_{syn} have been obtained, the syntactic semantic information is incorporated into the sentence representation via ARGCN to enhance syntactic feature interactions. ARGCN considers each word in the text as a node, with annotation tags of inter-word dependencies serving as edge relations. Additionally, the distance feature is introduced to enrich local syntactic relation features. The specific procedure is as in Equations (4) to (7) in the related work. It is worth noting that to incorporate the syntactic annotation tag feature information into the model, E_{syn} is used to replace b_r in Equation (6), resulting in the following Equation (15):

$$\alpha_{ij,r} = \sigma(W_c^T[E_{syn}, \beta_{ij}]) \tag{15}$$

Where $\alpha_{ij,r}$ denotes the attention coefficient incorporated into the annotated tag features.

3.5. **Grid Tagging Scheme Layer.** The grid tagging layer consists of feature fusion and grid tagging. Its primary function is to fuse multiple feature information and label word pair relationships using the grid tagging scheme. First, the ARGCN output features $h_a = \{h_1, h_2, ..., h_n\}$, $h_a \in R^{n\times d}$, are dot-producted with $E_{gts}^T \in R^{d\times dg}$ to obtain the fused feature $h_{label} \in R^{n\times dg}$ of syntactic information and grid annotation tags. Then, concatenation is applied to merge h_{label} with the semantic feature E_{bert}, resulting in a final feature vector $h_f \in R^{n\times(d+dg)}$. The specific operating equations are shown in the following Equations (16) and (17):

$$h_{label} = h_a \otimes E_{gts}^T \tag{16}$$

$$h_f = [h_{label}; E_{bert}] \tag{17}$$

Then, the features h_f are labeled with word pairs, and the grid tagging is essentially a linear classification, which directly linearly classifies the word pairs (w_i, w_j) to get the word pair relationship score vector, and then normalized to get the tag distribution probability of the word pairs as shown Equations (18) and (19):

$$z_{i,j} = Linear(h_{f_{i,j}}) \tag{18}$$

$$r_{i,j} = softmax(z_{i,j}) \tag{19}$$

where $h_{f_{i,j}} \in R^{1\times1\times(d+dg)}$ denotes the feature representation corresponding to the i-th row and j-th column in the grid.

3.6. **Output Layer.** The output layer focuses on generating triples using a grid tagging specific decoding strategy. First, entity tags $E1$ or $E2$ are identified from the predicted labels of all word pairs (w_i, w_j) along the main diagonal. Then, an upward search is performed to find consecutive labels $ER1$ or $ER2$ to form a complete head entity word $e1$ or tail entity word $e2$. If $w_i \in e1$ and $w_j \in e2$, the relationship types $r \in \{CON, TRE, PER, CAU\}$ with the most frequently occurring label is selected to form the entity relation triple $(e1, e2, r)$. When the number of tags is the same, the order of r is determined as $CON > TRE > PER > CAU$. If all predicted labels are N, $e1$ and $e2$ are considered unable to form a valid triple.

3.7. **Training Loss.** In this paper, the cross-entropy loss function is applied to minimize the true value $y_{i,j}$ and the predicted value $r_{i,j}$ for all word pairs as shown in the following Equation (20):

$$l = -\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k\in G}^{n} I(y_{i,j} = k)log(r_{i,j}) \tag{20}$$

where G denotes the set of tags $\{E1, ER1, E2, ER2, CON, TRE, PER, CAU, N\}$ and n denotes the length of the sentence.

4. **Experiments and results for TCM Formula Data.**

4.1. **Dataset and Metric.** The Chinese dataset TCMF used in this study consists of 1,435 data items, adapted from the "Traditional-Chinese-Medicine-Dataset-SFT" available on Hugging Face. Each data item was manually labeled according to the labeling strategy outlined in this paper. The TCM formula dataset contains a total of five types of entities: formula, Chinese medicine, symptom type, symptom, and cause of disease, and four types of relationships: formula/Chinese medicine, formula/symptom type, symptom type/symptom, and cause of disease/symptom type.

The TCMF dataset is divided into training, validation, and test sets in an 8:1:1 ratio. Each sample in the dataset is labeled with all entity relation triples related to TCM formulae. The detailed dataset statistics are shown in Table 3, where #S denotes the number of sentences, #T denotes the total number of triples, and #FH, #FS, #SS, and #ES denote the number of triples for formula/Chinese medicine, formula/symptom type, symptom type/symptom, and cause of disease/symptom type, respectively.

Table 3. TCMF dataset statistics

Dataset	#S	#T	#FH	#FS	#SS	#ES
Train	1140	3908	2430	684	433	361
Dev	142	487	316	46	83	42
Test	153	845	531	52	227	35

In order to evaluate the performance of JERE-TCMFD model, this paper uses Recall (R%), Precision (P%), and F1-score (F1%) as evaluation metrics. A prediction is considered correct if the head entity, tail entity, and relation category in the extracted triple match the ground truth.

4.2. **Experimental Settings.** The experimental hyperparameters are listed in Table 4, and BERT is used to generate word vectors. The model is trained with a learning rate of 5e-5, a batch size of 16, a maximum sequence length of 100, and 100 training epochs. The Adam optimizer is used to optimize the network, and a dropout rate of 0.1 is applied to prevent overfitting. The experimental environment is described in Table 5, and the model is implemented and trained using the PyTorch framework. The best values for each metric are highlighted in bold.

Table 4. Hyper Parameter Setting

Hyper Parameter	Value
BERT-dim	768
dropout	0.1
learning rate	5e-5
epoch	100
batch-size	16
ARGCN Layers	8

Table 5. Environments of the experiment

Name	Parameter
operating system	Windows 10
Video card	Nvidia RTX 3070
video memory	8.0 GB
development tool	PyCharm
Deep Learning Framework	Pytorch 1.10.0+cu11.3

4.3. **Baseline.** To demonstrate the validity of the JERE-TCMFD model, which will be compared to the following pipeline and joint extraction baseline:

Pipeline. BiLSTM-CRF+SVM [25]: BiLSTM-CRF sequence annotation is used for entity recognition and relationship extraction is performed using SVM based on the entity recognition results.

BiLSTM-CRF+LSTM: BiLSTM-CRF sequence annotation is used for entity recognition and relationship extraction is performed using LSTM based on the entity recognition results.

Joint extraction. GTS: The GTS model transformed the entity relation extraction task into a grid tagging task and uniformly extracted the triples using a specific decoding strategy.

EMC-GCN [26]: The EMC-GCN model convert syntactic dependency type features, word features, dependency tree distance features, and relative distance features into a multi-channel graph, and extract these relational features through GCN to strengthen the connection between words.

Span-ASTE [14]: The model generates all possible head-entity segments or tail-entity segments by detecting the semantics of the whole span and predicting their relationship types.

WPRN [23]: On the basis of GTS, the WPRN model proposed Word-Pair Relation Tagging Scheme (WPRTS), which incorporates the triple information in the sentence into the word-pair relation tagging, and learned from the idea of multi-head self-attention to construct Word-Pair Relation Network (WPRN) to learn tagging.

Table 6. Experimental results on TCMF

Model	P% (Precision)	R%(Recall)	F1%(F1-score)
BiLSTM-CRF+SVM	70.16	72.79	71.48
BiLSTM-CRF+LSTM	76.43	75.20	75.82
EMC-GCN	79.71	81.87	80.80
Span-ASTE	82.62	83.89	83.25
WPRN	83.60	**84.34**	83.97
JERE-TCMFD	**85.16**	83.99	**84.57**

4.4. **Main Results.** Table 6 presents the comparison results of the JERE-TCMFD model on the TCMF dataset. The following conclusions can be drawn from these results:

Firstly, all end-to-end models in Table 6 outperform the pipeline model significantly. This is because the pipeline model suffers from the error propagation problem, where errors from earlier tasks affect subsequent tasks. In contrast, end-to-end models establish correlations between entity extraction and relationship extraction subtasks, mitigating error propagation through joint training. The traditional pipeline models, such as BiLSTM-CRF+SVM and BiLSTM-CRF+LSTM, achieve F1-scores of 71.48% and 75.82%, respectively, which are considerably lower than those of end-to-end models. This confirms that a joint approach improves information extraction by reducing cascading errors.

Secondly, compared with the WPRN model, the JERE-TCMFD model improves the F1-score by 0.6% on the TCMF dataset. This improvement is mainly due to the incorporation of labeled semantic information, which strengthens contextual linkages between entities and sentences, and the use of ARGCN to enhance syntactic dependencies, leading to more accurate grid tag classification. Additionally, the precision of JERE-TCMFD (85.16%) is the highest among all models, indicating its superior ability to extract relevant entities and relationships with minimal false positives.

Finally, the F1-score of the JERE-TCMFD model increases by 8.75% over the traditional pipeline approach and is higher than all the end-to-end models in the table, which illustrates the effectiveness of the model proposed in this paper. Notably, JERE-TCMFD achieves a balance between precision and recall, whereas other models tend to have one metric significantly higher than the other. This indicates that our approach does not sacrifice recall for higher precision or vice versa, but instead optimally enhances both aspects. Furthermore, the relative improvements in recall (83.99%) compared to other end-to-end models such as EMC-GCN (81.87%) and Span-ASTE(83.89%) demonstrate that JERE-TCMFD effectively captures more relevant entities and relationships while maintaining overall robustness.

In summary, these results validate the effectiveness of the JERE-TCMFD model in overcoming the limitations of traditional pipeline methods and improving the performance of entity-relation extraction tasks. The model's superior performance stems from its joint learning framework, enhanced syntactic dependency modeling, and incorporation of semantic cues, making it a promising approach for structured information extraction in the TCMF domain.

4.5. **Ablation Study.** To investigate the effectiveness of the three modules of ARGCN, syntactic annotation tag semantic features, and grid annotation tag semantic features in JERE-TCMFD model, this paper conduct ablation study on the TCMF dataset. The experimental results are shown in Table 8. #RGCN indicates the replacement of ARGCN with RGCN. In this case, the F1-score of JERE-TCMFD decreased by 1.34% on the TCMF dataset. Since ARGCN contains distance feature information, it can help the model to improve its performance to some extent. w/o Syntactic annotation tag means that the removal of syntactic annotation tag semantic features from JERE-TCMFD. Removing the syntactic annotation information resulted in a 0.25% decrease in the models F1-score. Although this decline is minor, it suggests that syntactic annotation features contribute to model performance by providing additional linguistic cues that help capture sentence semantics and structure. w/o Grid annotation tag refers to the removal of grid annotation tag semantic features from the JERE-TCMFD model. The absence of grid annotation information led to a 1.4% decrease in the models F1-score, demonstrating that these semantic features enhance model performance.

In summary, each module in the JERE-TCMFD contributes to the overall performance of the entity relation extraction task.

Table 7. Experimental results of ablation studies

Model	P% (Precision)	R%(Recall)	F1%(F1-score)
JERE-TCMFD	**85.16**	83.99	**84.57**
#RGCN	85.02	81.53	83.23
w/o Syntactic annotation tag	84.58	**84.06**	84.32
w/o Grid annotation tag	84.58	81.81	83.17

4.6. **Impact Analysis of ARGCN Layers.** The number of stacked ARGCN layers plays a crucial role in the performance of the JERE-TCMFD model. To explore this effect, we conduct experiments by varying the number of stacked ARGCN layers from 1 to 6 on the TCMF dataset. The experimental results, as illustrated in Figure 7, show a general trend where model performance first improves and then declines as the number of layers increases. Specifically, when the number of layers is set to 2, the model achieves the highest F1 score, indicating that this configuration effectively captures grammatical features while maintaining an appropriate level of complexity. In contrast, when only a single layer is used, the model struggles to learn sufficient grammatical features, leading to suboptimal performance. On the other hand, as the number of layers exceeds 2, the performance gradually deteriorates due to the increased depth of the model. This decline can be attributed to issues such as vanishing gradients or gradient explosion, which hinder effective weight updates during training.

From these results, we conclude that an optimal balance must be struck between model depth and representational capacity. While increasing the number of layers initially enhances feature extraction, excessive depth can introduce instability, ultimately degrading model performance. Thus, setting the number of ARGCN layers to 2 provides the best trade-off between learning capacity and computational efficiency for the JERE-TCMFD model on the TCMF dataset.

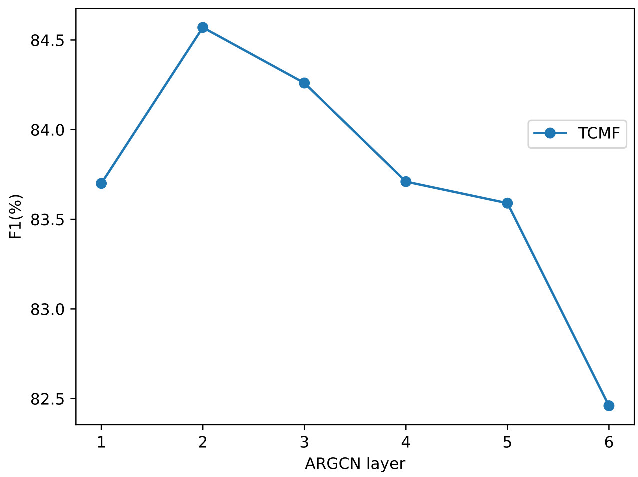

Figure 7. ARGCN Layer Analysis Results

4.7. **Case Study.** To further assess the impact of the JERE-TCMFD model on entity relation extraction performance, two representative cases were selected from the test set and analyzed using both the BiLSTM-CRF+LSTM model and the JERE-TCMFD model, as shown in Table 8. In Example 1, JERE-TCMFD successfully extracts all four correct entity-relation triples, whereas BiLSTM-CRF-LSTM fails to extract two triples associated with the entity "Enlightened Master Viscera-Nourishing Decoction". This discrepancy may stem from BiLSTM-CRF-LSTM‘s limitation as a pipelined model, which prevents it from accurately identifying Enlightened Master Viscera-Nourishing Decoction during entity recognition. In contrast, JERE-TCMFD enhances both entity recognition and relationship classification through joint modeling of entities and relationships. In Example 2, JERE-TCMFD also extracts all three correct triples, whereas BiLSTM-CRF-LSTM fails to extract the triple (Ephedra Soup, Cinnamon, Contain). This may be due to BiLSTM-CRF-LSTMs failure to accurately recognize Cinnamon as an entity during entity identification or its inability to correctly establish the relationship between Cinnamon and Ephedra Soup during relationship classification.

In summary, the end-to-end JERE-TCMFD model significantly outperforms the pipelined BiLSTM-CRF-LSTM model in relationship extraction for TCM formulas by jointly modeling entities and relationships, while leveraging annotation tags to strengthen inter-sentence dependencies and fully exploit contextual information.

Table 8. Different models outputs for a given sentence

Sentence	Correct triples	BiLSTM-CRF	JERE-TCMFD
Both TCM formulas, Xiaoyao San and Enlightened Master Viscera-Nourishing Decoction, contain Atractylodes macrocephala and Paeonia lactiflora.	(Xiaoyao San, Atractylodes macrocephala, contain) (Xiaoyao San, Paeonia lactiflora, contain) (Enlightened Master Viscera-Nourishing Decoction, Atractylodes macrocephala, contain) (Enlightened Master Viscera-Nourishing Decoction, Paeonia lactiflora, contain)	(Xiaoyao San, Atractylodes macrocephala, contain) (Xiaoyao San, Paeonia lactiflora, contain)	(Xiaoyao San, Atractylodes macrocephala, contain) (Xiaoyao San, Paeonia lactiflora, contain) (Enlightened Master Viscera-Nourishing Decoction, Atractylodes macrocephala, contain) (Enlightened Master Viscera-Nourishing Decoction, Paeonia lactiflora, contain)
Ephedra and Cinnamon must be used to increase enhance their sweat-inducing and exterior-releasing effects, such as ephedra soup, used for wind-cold exterior syndrome.	(ephedra soup, Ephedra, contain) (ephedra soup, Cinnamon, contain) (ephedra soup, wind-cold exterior syndrome, treatment)	(ephedra soup, Ephedra, contain)) (ephedra soup, wind-cold exterior syndrome, treatment)	(ephedra soup, Ephedra, contain) (ephedra soup, Cinnamon, contain) (ephedra soup, wind-cold exterior syndrome, treatment)

5. **Conclusion.** TThis paper presents the Joint Entity-Relation Extraction for Traditional Chinese Medicine Formula Data (JERE-TCMFD) model. It enhances model performance by incorporating semantic annotation tag features. The proposed model enhances performance by incorporating semantic features from annotated tags and transforms the entity relation extraction task into triple joint extraction using an nine-tag grid tagging approach, thereby enhancing both generalization and extraction accuracy. The results of the experiments on the TCMF dataset show the effectiveness of the model proposed in the entity relation extraction task. Although current research has addressed the joint extraction problem using a grid tagging scheme, entity terms in text are often composed of multiple words, leading to redundant word pair labeling and making it challenging for the model to extract complete phrases. Future research will explore methods to reduce the workload of word pair tagging by constructing grid phrase tagging.

Acknowledgment. This work is supported by the Natural Science Foundation of Fujian Province of China under Grant(No.2022J01398), National Natural Science Foundation of China (No.62172095).

REFERENCES

[1] W. Jia, R.-Z. Ma, L. Yan, W.-N. Niu, and Z.-M. Ma, "Joint entity and relation extraction with table filling based on graph convolutional networks," *Expert Systems with Applications*, vol. 266, p. 126130, 2025.

[2] R. Grishman and B.-M. Sundheim, "Message understanding conference-6: A brief history," in *Proceedings of the 16th Conference on Computational Linguistics - Volume 1.* Association for Computational Linguistics, 1996, pp. 466–471.

[3] K. Detroja, C. Bhensdadia, and B.-S. Bhatt, "A survey on relation extraction," *Intelligent Systems with Applications*, vol. 19, p. 200244, 2023.

[4] Z.-X. Zhong and D.-Q. Chen, "A frustratingly easy approach for entity and relation extraction," in *2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021.* Association for Computational Linguistics, 2021, pp. 50–61.

[5] Q. Zhao, D.-Z. Xu, J.-Q. Li, L.-N. Zhao, and F.-A. Rajput, "Knowledge guided distance supervision for biomedical relation extraction in chinese electronic medical records," *Expert Systems with Applications*, vol. 204, p. 117606, 2022.

[6] D. Sousa and F. M. Couto, "Biomedical relation extraction with knowledge graph-based recommendations," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 8, pp. 4207–4217, 2022.

[7] W.-J. Li, P. Zhang, F.-R. Wei, Y.-X. Hou, and Q. Lu, "A novel feature-based approach to chinese entity relation extraction," in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers.* Association for Computational Linguistics, 2008, pp. 89–92.

[8] Y.-M. Shang, H.-Y. Huang, and X.-L. Mao, "Onerel: Joint entity and relation extraction with one module in one step," in *Proceedings of the AAAI Conference on Artificial Intelligence.* Association for the Advancement of Artificial Intelligence, 2022, pp. 11 285–11 293.

[9] M. Miwa and M. Bansal, "End-to-end relation extraction using lstms on sequences and tree structures," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* Association for Computational Linguistics, 2016, pp. 1105–1116.

[10] T.-T. Hang, J. Feng, L. Yan, Y.-F. Wang, and J.-M. Lu, "Joint extraction of entities and relations using multi-label tagging and relational alignment," *Neural Computing and Applications*, vol. 34, no. 8, pp. 6397–6412, 2022.

[11] S.-C. Zheng, F. Wang, H.-Y. Bao, Y.-X. Hao, P. Zhou, and B. Xu, "Joint extraction of entities and relations based on a novel tagging scheme," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* Association for Computational Linguistics, 2017, pp. 1227–1236.

[12] W.-X. Xu, L. Wang, M.-C. Zhang, J.-L. Zhu, J.-Q. Yan, and Q.-T. Wu, "A joint entity relation extraction method for document level traditional chinese medicine texts," *Artificial Intelligence in Medicine*, vol. 154, p. 102915, 2024.

[13] Z. Wu, C.-C. Ying, F. Zhao, Z.-F. Fan, X.-Y. Dai, and R. Xia, "Grid tagging scheme for aspect-oriented fine-grained opinion extraction," in *Findings of the Association for Computational Linguistics: EMNLP 2020.* Association for Computational Linguistics, 2020, pp. 2576–2585.

[14] L. Xu, Y.-K. Chia, and L.-D. Bing, "Learning span-level interactions for aspect sentiment triplet extraction," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers).* Association for Computational Linguistics, 2021, pp. 4755–4766.

[15] P.-T. Lv, Q.-H. Zhang, M. Wu, M. Deveci, L. Shi, Y.-Y. Sun, K.-Y. Zhong, and J.-P. Wang, "Intelligent extraction of medical entity relationship based on graph neural network and optimization strategy," *Knowledge-Based Systems*, vol. 294, p. 111735, 2024.

[16] Y.-H. Lu, H.-Y. Chen, Y.-Y. Zhang, J.-H. Peng, D.-C. Xiang, and J.-X. Zhang, "Research on entity relation extraction for chinese medical text," *Health Informatics Journal*, vol. 30, no. 3, p. 14604582241274762, 2024.

[17] M.-Y. Cao, Z.-H. Vang, L. Luo, H.-F. Lin, and J. Wang, "Joint drug entities and relations extraction based on neural networks," *Journal of Computer Research and Development*, vol. 56, no. 07, pp. 1432–1440, 2019.

[18] X.-D. You, M.-Z. Zhao, X.-Y. Wang, G. Xu, and X.-Q. Lv, "A relation extraction method in the medical field integrating the entity category features," *Journal of Beijing Institute of Machinery — J Beijing Inst Mach*, vol. 37, no. 06, pp. 19–25, 2022.

[19] Y.-Y. Yang, J.-Q. Du, B. Nie, J.-G. Luo, and J. He, "A joint extraction method of entities and relations for traditional chinese medicine based on deep learning," *Computer Applications and Software*, vol. 40, no. 03, pp. 217–222+234, 2023.

[20] J.-F. Jiang, A. Wang, and A. Aizawa, "Attention-based relational graph convolutional network for target-oriented opinion words extraction," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics, 2021, pp. 1986–1997.

[21] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*. Springer, 2018, pp. 593–607.

[22] Z.-H. Dai, Z.-L. Yang, Y.-M. Yang, J.-G. Carbonell, Q. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019, pp. 2978–2988.

[23] H.-B. Xia, Q. Li, and Y.-F. Xiao, "Word-pair relation learning method for aspect sentiment triplet extraction," *Pattern Recognition and Artificial Intelligence*, vol. 35, no. 03, pp. 262–270, 2022.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, 2017.

[25] G. Bekoulis, J. Deleu, T. Demeester, and C. Develder, "Joint entity recognition and relation extraction as a multi-head selection problem," *Expert Systems with Applications*, vol. 114, no. 30, pp. 34–45, 2018.

[26] H. Chen, Z.-P. Zhai, F.-X. Feng, R.-F. Li, and X.-J. Wang, "Enhanced multi-channel graph convolutional network for aspect sentiment triplet extraction," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (volume 1: long papers)*. Association for Computational Linguistics, 2022, pp. 2974–2985.