Joint Entity and Relation Extraction for Legal Documents with Legal Feature Enhancement

Yanguang Chen[1], **Yuanyuan Sun**[1*], **Zhihao Yang**[1], **Hongfei Lin**[1]
[1]School of Computer Science and Technology, Dalian University of Technology
cygAriel@mail.dlut.edu.cn, {syuan, yangzh, hflin}@dlut.edu.cn

Abstract

In recent years, the plentiful information contained in Chinese legal documents has attracted a great deal of attention because of the large-scale release of the judgment documents on *China Judgments Online*. It is in great need of enabling machines to understand the semantic information stored in the documents which are transcribed in the form of natural language. The technique of information extraction provides a way of mining the valuable information implied in the unstructured judgment documents. We propose a Legal Triplet Extraction System for drug-related criminal judgment documents. The system extracts the entities and the semantic relations jointly and benefits from the proposed entity feature and multi-task learning framework. Furthermore, we manually annotate a dataset for Named Entity Recognition and Relation Extraction in Chinese legal domain, which contributes to training supervised triplet extraction models and evaluating the model performance. Our experimental results show that the entity feature introduction and multi-task learning framework are feasible and effective for the Legal Triplet Extraction System. The F1 score of triplet extraction finally reaches 0.836 on the legal dataset.

1 Introduction

Automatic extraction of information from legal documents is crucial for legal document analysis and related business processing. The techniques of information extraction are pivotal modules for down-stream judicial applications such as the assistance of reviewing case documents, identification of criminal case facts and auxiliary generation of legal documents. Besides, by implementing named entity recognition and relation extraction, the judgment documents can be transformed into several triplets, capturing entity pairs and their interrelations inside the fact description. The structured legal triplets conduce to legal knowledge graph construction, which benefits query capabilities and interpretability in judicial applications. There are great quantities of actual judgment documents released on *China Judgments Online*[1]. The abundant information contained in these judgment documents is worthy of in-depth study thanks to the authenticity and typicality of the documents. In this work, we conduct information extraction based on these public legal texts.

Recently, information extraction techniques have developed rapidly. Neural networks for named entity recognition (Lample et al., 2016; Zhu and Wang, 2019) and deep learning methods applying to relation extraction (Nguyen and Grishman, 2015; Zhou et al., 2016) are extensively investigated. Early methods accomplish the triplet extraction drawing on the idea of pipelining. They first recognize the entities in the texts and then classify each entity pair into predefined relation types. These methods suffer from the error propagation problem caused by the incorrect and redundant entities attained by the previous step. Besides, the interactions between the prediction of entities and relations also need to be emphasized. For these challenges, joint learning methods have been intensively carried out.

*Corresponding author.

[1]http://wenshu.court.gov.cn/

Proceedings of the 28th International Conference on Computational Linguistics, pages 1561–1571
Barcelona, Spain (Online), December 8-13, 2020

Initially, joint models based on neural networks (Miwa and Bansal, 2016; Zheng et al., 2017a) apply parameter sharing mechanism to extract entities and relations in a single model. However, these models essentially treat entity recognition and relation extraction as two separated steps and do the prediction in a pipeline. The novel tagging scheme (Zheng et al., 2017b) converts the joint extraction task into a sequence tagging problem and decodes the entities and relations all together. But it cannot solve the problem of overlapping triplets since it assigns each token only one tag. Zeng et al. (2018) proposes a joint extraction method based on the Sequence-to-Sequence (Seq2Seq) model with copy mechanism, which handles the overlapping problem. Recently some Seq2Seq-based methods (Nayak and Ng, 2019; Zeng et al., 2020) improve the performance of joint entity and relation extraction on the accuracy as well as the computational efficiency.

In this work, we concentrate on the triplet extraction in Chinese legal domain, especially on the drug-related criminal cases. There are twelve crime types of drug-related crimes, and three of them have the most cases, i.e. drug trafficking, illegal possession of drugs and providing venues for drug users. Taking the representation of the facts of drug-related cases into account, we define four relation types, i.e. *traffic_in*, *sell_drug_to*, *possess* and provide_shelter_for, which cover the crime of the most three drug-related types according to *Criminal Law of The People's Republic of CHINA*. Specifically, they differ in that *traffic_in* denotes the fact that the suspect deals in drugs and *sell_drug_to* denotes that the suspect conducts a drug trade with someone else. It is obvious that these two relations generally share the same head entity in one case, which appears as the overlapping triplets.

In consideration of the performance and the solution to overlapping problem, we propose a Legal Triplet Extraction System based on the Seq2Seq model to accomplish joint entity and relation extraction from the drug-related criminal judgment documents. The system consists of three main components, encoder, decoder and sequence tagging layer. Concretely, the encoder converts the source sentences into semantic vectors with legal feature enhancement. We explore the pre-trained language model BERT for the encoder in light of its remarkable performance on multiple NLP tasks. The decoder is an efficient network able to solve the overlapping problem, inspired by previous work (Nayak and Ng, 2019). The sequence tagging layer serves as an auxiliary task and assists the encoder in learning the entity boundary information. Our main contributions can be summarized as follows:

(i) We propose a Legal Triplet Extraction System based on the Seq2Seq framework, which can jointly extract the entities and the interrelationships from legal texts.

(ii) We focus on the triplet extraction on the drug-related criminal cases, and thus introduce the lexicon of drug names as the legal feature into the model. Furthermore, an auxiliary sequence tagging task is utilized to strengthen the entity recognition ability of the model.

(iii) We manually annotate a dataset for Named Entity Recognition and Relation Extraction in Chinese legal domain, based on the drug-related criminal judgment documents. The dataset is valuable for training supervised triplet extraction models and evaluating the model performance.

2 Related Work

2.1 Joint Entity and Relation Extraction

The techniques of information extraction are crucial for many downstream natural language tasks such as knowledge graph construction and question answering system. Early works regard the triplet extraction task as two separate subtasks, i.e. named entity recognition (Lample et al., 2016; Ma and Hovy, 2016; Chiu and Nichols, 2016; Zhu and Wang, 2019) and relation classification (Zeng et al., 2014; Nguyen and Grishman, 2015; Zhou et al., 2016; Wang et al., 2019; Wu and He, 2019). They use pipeline methods to extract the triplets, where all entities in the texts are recognized firstly and then combined into entity pairs for relation prediction. The pipeline methods suffer from the error propagation and ignore the interactions between the prediction of entities and relations.

To tackle the aforementioned challenges, joint models for triplet extraction are well-studied. Initial methods for joint extraction (Hoffmann et al., 2011; Li and Ji, 2014; Miwa and Sasaki, 2014; Ren et al.,

2017) are feature-based, which rely on heavy feature engineering and require quantities of manual efforts and domain knowledge. With the development of Deep Neural Networks, models that automatically learn to extract features have emerged. Some neural-network-based joint models achieve the interaction between entity recognition and relation classification by sharing the parameters (Miwa and Bansal, 2016; Zheng et al., 2017a). The tagging methods utilize novel tagging schemes to convert the triplet extraction task into a sequence tagging problem and jointly extract the entities and relations (Zheng et al., 2017b; Yu et al., 2019; Wei et al., 2019). The Seq2Seq model (Sutskever et al., 2014) is another method for joint extraction. Zeng et al. (2018) proposes CopyRE, a Seq2Seq-based model with copy mechanism, which can solve the overlapping problem. Other works based on the Seq2Seq model (Nayak and Ng, 2019; Zeng et al., 2020) optimize the decoder of CopyRE and improve the performance of the triplet extraction on the accuracy as well as the computational efficiency. In addition, Takanobu et al. (2019) apply a hierarchical reinforcement learning framework to deal with overlapping triplets in joint extraction. Fu et al. (2019) and Sun et al. (2019) take advantage of the graph structure and use graph neural networks to jointly learn the entities and relations.

2.2 Pre-trained Language Model

There is a long history and rich literature on pre-trained language models. It is able for pre-trained language models to capture the meaning of words dynamically considering their context. Conneau et al. (2017) shows the effectiveness of universal sentence representations trained with supervision on Stanford Natural Language Inference datasets. Some approaches regard learned representations as features in a model for the downstream task. According to different granularities, there are word embedding methods (Mikolov et al., 2013), as well as sentence embedding methods (Logeswaran and Lee, 2018). ELMo (Peters et al., 2018) and its predecessor generate context sensitive word representations through stacked bidirectional LSTM and residual structure. For unsupervised fine-tuning approaches on language models, Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018) improves the state-of-the-art results on many natural language processing tasks. On this basis, BERT with Whole Word Masking (BERT-wwm) (Cui et al., 2019) are proposed for Chinese NLP tasks. Yang et al. (2019) exploits XLNet, which is a generalized autoregressive pre-training method. RoBERTa (Liu et al., 2019) improves the training procedure of BERT and achieves state-of-the-art results on various tasks with substantial improvement.

The pre-trained language model BERT benefits many NLP tasks thanks to its abundant prior knowledge. However, it captures the general language information from the large-scale corpus during the training procedure, which leads to the lack of the task-specific and domain-specific knowledge. There are researches aiming at integrating knowledges into the BERT model. SG-Net (Zhang et al., 2020) incorporates explicit syntactic constraints into attention mechanism in order to guide the text modeling with syntax. Works on embedding the knowledge bases into pre-trained language models (Liu et al., 2020; Peters et al., 2019) contribute to introducing domain knowledge.

3 Legal Triplet Extraction System

In this section, we describe our method to extract the triplets occurring in the fact descriptions of judgment documents. We start with the introduction of the triplet extraction task in Chinese legal domain and propose a manually annotated dataset for legal triplet extraction. We introduce each component of the proposed Legal Triplet Extraction System and the training procedure in detail. The architecture of the triplet extraction system is shown as Figure 1.

3.1 Task Description and Dataset Construction

In this research, we concentrate on the triplet extraction in Chinese legal domain, especially on the drug-related criminal cases. Given a fact description sentence S from the judgment documents, the target of the legal triplet extraction system is to identify the triplets in the form of $\langle e_1, r, e_2 \rangle$, where e_1 and e_2 are the entities in S, and r is the relationship between them.

There are twelve crime types of drug-related crimes.We focus on three types among them with the most

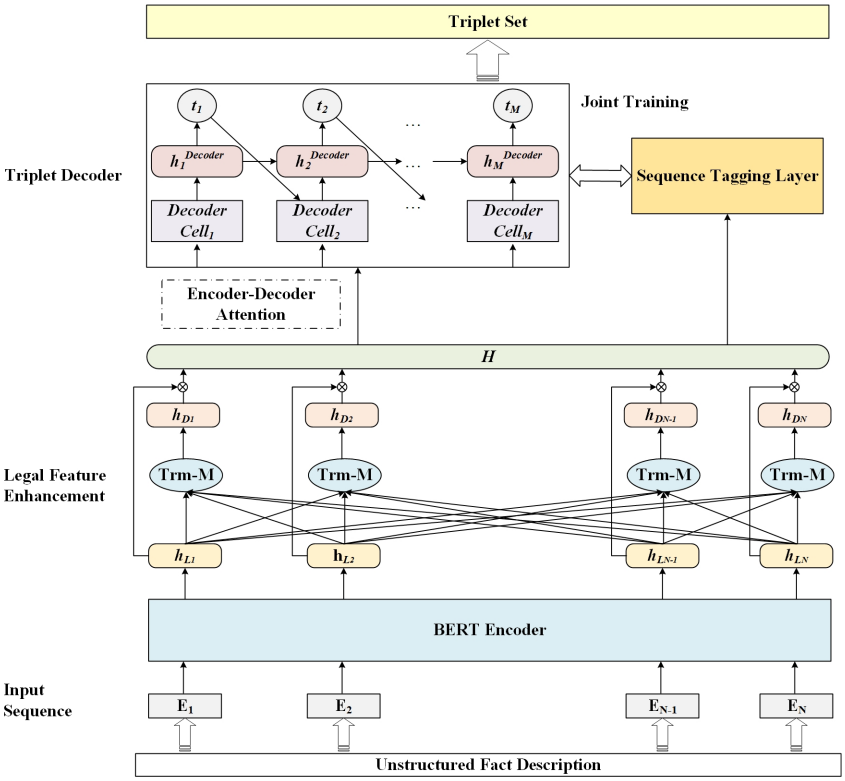

Figure 1: The architecture of the Legal Triplet Extraction System. The operator '$\otimes$' denotes the operation of weighted average.

cases, i.e. drug trafficking, illegal possession of drugs and providing venues for drug users. In practice, it happens that one case involves more than two crimes. The relevant crimes need to be distinguished and reorganized for measurement of penalty. In order to describe the key criminal behaviors covering the three drug-related crime types, we summarize four relation types, i.e. *traffic_in*, *sell_drug_to*, *possess* and *provide_shelter_for* according to the criminal law. Concretely, *traffic_in* and *sell_drug_to* represent the relationships in drug trafficking cases. The former denotes the fact that the suspect deals in drugs and the latter means that the suspect conducts a drug trade with someone else. The relation type *possess* denotes that the suspect holds a certain amount of narcotic drugs, and *provide_shelter_for* describes the fact that the suspect provides shelter for others to ingest or inject drugs. These two relations cover the crime types of illegal possession of drugs and providing venues for drug users respectively.

In order to realize legal information extraction, we manually annotate the entities and their relations of the drug-related criminal judgment documents, which are downloaded from *China Judgments Online*. The fact descriptions are firstly extracted from the raw documents through rules. There are 1750 fact descriptions selected as the instances to be annotated. We annotate the relations between all entity pairs in every instance. The annotated data is conducive to supervised training and performance evaluation.

3.2 Encoder with Legal Feature Enhancement

The encoder targets to convert the source sentences into semantic vectors. we explore RoBERTa (Liu et al., 2019) to encode the contextual information. The architecture of the pre-trained model is the same as BERT (Devlin et al., 2018), which is a L-layer bidirectional Transformer encoder (Vaswani et al., 2017). For RoBERTa$_{\mathrm{BASE}}$ the number of L is 12, and for RoBERTa$_{\mathrm{LARGE}}$ L is 24. In this work, we use the model of RoBERTa$_{\mathrm{BASE}}$ for the encoder. The hidden state vectors of the last layer from RoBERTa$_{\mathrm{BASE}}$ are utilized as the general representation of each token in the input sentence S, denoted as $\boldsymbol{H_L}$.

The encoding of the pre-trained model tends to capture the general text representation but is short of domain knowledge. In order to make up for the lack of legal domain information, we add legal feature enhancement into the encoder. The accuracy of the recognized entities is crucial for the performance

of triplet extraction, hence the lexicon feature for the entities is worthy of exploration. Towards the triplet extraction on the Chinese drug-related judgment documents, we first build up a drug name lexicon $Lexicon_{Drug}$ as the legal feature. We collect the scientific names of all kind of drugs and their common statements recording in the judgment documents. The drug name lexicon includes the possible expressions of all existing drugs, both in written and spoken language.

Given an input sequence $S = w_1, w_2, \ldots, w_N$, where w_i denotes the i-th token in the input sequence and N denotes the length of the sequence, we match it with the drug name lexicon and find all subsequences that may form the expressions of the drugs. We define $S[i:j]$ as the subsequence of S which begins with token w_i and ends with token w_j. We utilize a mask matrix M_D to represent the legal feature of drug names. M_D is organized as a $N \times N$ matrix, where the element m_{ij} at the i-th row and the j-th column denote whether the subsequence $S[i:j]$ is an expression of the drugs. The computational process of M_D can be mathematically described as:

$$m_{ij} = \begin{cases} 1 & \text{if } S[i:j] \in Lexicon_{Drug} \\ 0 & otherwise. \end{cases} \tag{1}$$

We calculate the legal domain specific representation of the input sentence with an extra Transformer encoder layer (Vaswani et al., 2017). The layer has two sub-layers, i.e. a multi-head self-attention mechanism and a position-wise feed-forward network. Each sub-layer is followed by a residual connection and layer normalization. The general representation of the input sentence $\boldsymbol{H_L}$ is first projected into distinct matrixes Q_h^D, K_h^D, V_h^D, where h denotes the h-th head of the multi-head attention. Q_h^D, K_h^D and V_h^D have the dimention of $N \times d_q$, $N \times d_k$ and $N \times d_v$, respectively. The output of the self-attention function, denoted as Att_h^D, is computed with the legal feature mask matrix M_D:

$$Att_h^D = softmax(M_D * \frac{(Q_h^D {K_h^D}^T)}{\sqrt{d_k}}) V_h^D \tag{2}$$

where the operator '*' denotes the mask operation in the attention computing.

We concatenate the outputs of all the attention heads and pass the result through the feed-forward sublayer. The final output from the feature-masked Transformer encoder is integrated with the drug lexicon feature, denoted as $\boldsymbol{H_D}$. Finally, we make a weighted average of the general representation $\boldsymbol{H_L}$ and the feature-fused representation $\boldsymbol{H_D}$ to obtain the legal feature enhanced representation $\boldsymbol{H^{Encoder}}$:

$$\boldsymbol{H^{Encoder}} = \gamma \boldsymbol{H_L} + (1 - \gamma) \boldsymbol{H_D} \tag{3}$$

where γ is the weighting parameter. In this work, we employ $\gamma = 0.5$.

3.3 Decoder

The decoder aims to predict the entity pairs and their interrelationships of the input sentence. A Long Short Term Memory (LSTM) network is utilized to decode the triplet sequence T. We regard the indexes of the first token and the last token of an entity as the representation of each entity. Thus we can extract the entities of a triplet from the original texts by locating them. Moreover, we obtain the relation type of an entity pair by a relation classifier.

Given the legal feature enhanced representation $\boldsymbol{H^{Encoder}}$ from the encoder, the triplet sequence $T = t_0, t_1, t_2, \ldots, t_M$ is decoded, where t_k denotes the k-th triplet in the sequence and M denotes the length of the triple sequence T. Since t_0 is the beginning of the decoded sequence, it has no practical meaning and is assigned to be a zero vector. $t_k (k > 0)$ represents a triplet $\langle e_1, r, e_2 \rangle$, constitutive of the starting index and the ending index of e_1 and e_2, and the relation type between them. The decoder keeps operating until the relation type of the current triplet turns to be 'NA' or the sequence length reaches the default maximum. For each time step k, we define the hidden state vector of decoder as $h_k^{Decoder}$ and the representation of decoded triplets before k as t_{pr}. t_{pr} is computed by the sum of the previous triplets.

$$t_{pr} = \sum_{i=0}^{k-1} t_i \tag{4}$$

We first calculate the encoder-decoder attention vector a_k with the representation $\boldsymbol{H^{Encoder}}$ from the encoder, the last hidden state vector $h_{k-1}^{Decoder}$ from the decoder and t_{pr} presented earlier. Then the hidden state vector at time step k of the decoder is computed with a LSTM cell. The input of the calculation is the concatenation of t_{pr} and a_k, which contains the information of the decoded triplets as well as the encoder representation.This process is denoted as:

$$a_k = Attention(\boldsymbol{H^{Encoder}}, h_{k-1}^{Decoder}, t_{pr}) \tag{5}$$

$$h_k^{Decoder} = LSTMcell([t_{pr}; a_k], h_{k-1}^{Decoder}) \tag{6}$$

Finally, we predict the indexes of the entity pair and the relation type which form a triplet based on $\boldsymbol{H^{Encoder}}$ and $h_k^{Decoder}$. The hidden state vector $h_k^{Decoder}$ is extended to the input sequence length N and obtain the matrix $\boldsymbol{H_k^{Decoder}}$. These two representations from the encoder and the decoder are concatenated and passed through a bi-directional LSTM layer.The probabilities of each token in the input sequence being the beginning and the end of the entity e_1 is computed:

$$\boldsymbol{H_k^1} = BiLSTM([\boldsymbol{H^{Encoder}}; \boldsymbol{H_k^{Decoder}}]) \tag{7}$$

$$p^{b1} = softmax(W^{b1}\boldsymbol{H_k^1}) \tag{8}$$

$$p^{e1} = softmax(W^{e1}\boldsymbol{H_k^1}) \tag{9}$$

where p^{b1} and p^{e1} denote the probabilities of each token being the beginning and the end of e_1. The indexes of tail entity e_2 are calculated in a similar way except for the input of the BiLSTM layer. p^{b2} and p^{e2} are the probabilities of each token being the beginning and the end of e_2.

$$\boldsymbol{H_k^2} = BiLSTM([\boldsymbol{H^{Encoder}}; \boldsymbol{H_k^{Decoder}}; \boldsymbol{H_k^1}]) \tag{10}$$

$$p^{b2} = softmax(W^{b2}\boldsymbol{H_k^2}) \tag{11}$$

$$p^{e2} = softmax(W^{e2}\boldsymbol{H_k^2}) \tag{12}$$

W^{b1}, W^{e1}, W^{b2} and W^{e2} mentioned below are trainable parameters. The embeddings of e_1 and e_2 in the current triplet are denoted as $\boldsymbol{e_k^1}$ and $\boldsymbol{e_k^2}$, which can be obtained by:

$$\boldsymbol{e_k^1} = [\sum_{i=1}^{N} p_i^{b1} h_i^1; \sum_{i=1}^{N} p_i^{e1} h_i^1] \tag{13}$$

$$\boldsymbol{e_k^2} = [\sum_{i=1}^{N} p_i^{b2} h_i^2; \sum_{i=1}^{N} p_i^{e2} h_i^2] \tag{14}$$

where h_i^1 and h_i^2 are the i-th hidden vector in $\boldsymbol{H_k^1}$ and $\boldsymbol{H_k^2}$. The conditional probability of the relation type between e_1 and e_2 is predicted by a relation classifier. Moreover, the representation t_k of the current triplet is computed, where $\boldsymbol{r_k}$ is the embedding of the relation type between e_1 and e_2 at time step k and W^r is a trainable parameter matrix.

$$p(r_k|S) = softmax(W^r[\boldsymbol{e_k^1}; \boldsymbol{e_k^2}; h_k^{Decoder}]) \tag{15}$$

$$t_k = [\boldsymbol{e_k^1}; \boldsymbol{e_k^2}; \boldsymbol{r_k}] \tag{16}$$

3.4 Sequence Tagging Layer

The recognition of entity span is crucial for the triplet extraction. It not only decides the accuracy of the entities in a triplet, but also partly influences the relation prediction of the triplet because of the decoding process. We use a sequence tagging layer to conduct entity span recognition. This auxiliary task conduces to introducing information of entity boundary to the model. The input of the sequence tagging layer is the legal feature enhanced representation $\boldsymbol{H^{Encoder}}$ from the encoder. We use a multi-label classifier to predict the entity span for each token in the input sequence. The probability of the tag sequence X is computed:

$$p(X|S) = softmax(\boldsymbol{H^{Encoder}}) \tag{17}$$

We adopt the BIO tagging scheme to distinguish entity boundary. Concretely, the tag 'B' denotes the beginning token of an entity, the tag 'I' denotes the token in a multi-token entity except the first token, and 'O' means that the token doesn't belong to any entities.

3.5 Training Details

In the training process, we use the ground-truth labels to obtain the relation embeddings. We mini-mize the negative log-likelihood loss for the prediction of the entity indexes and the relation types. Given all training examples $\{(S_i, T_i)\}|^H$, the loss function of the decoder is denoted as:

$$\mathcal{L}_{Dec} = \sum_{i=1}^{H}\sum_{j=1}^{M}\left(log\Big(p\big(r_{i,j}|S_i\big)\Big) + \alpha\sum_{z=1}^{2}\left(log\left(p_{i,j}^{bz}\right) + log\left(p_{i,j}^{ez}\right)\right)\right) \tag{18}$$

where H is the size of training examples, M is the length of decoded triplet sequence, and α is a weighting parameter.

The sequence tagging layer participates in calculating only in the training procedure. It plays a role in the assistance of learning the entity boundary information. Given all training examples $\{(S_i, X_i)\}|^H$, the loss function is computed with sentence-level log-likelihood loss:

$$\mathcal{L}_{Tag} = \sum_{i=1}^{H} log\Big(p\big(X_i|S_i\big)\Big) \tag{19}$$

The final loss of the Legal Triplet Extraction System is defined as the weighted summation of $\mathcal{L}_{Dec}$ and $\mathcal{L}_{Tag}$, where β denotes the weighting parameter:

$$\mathcal{L}_{final} = \mathcal{L}_{Dec} + \beta\mathcal{L}_{Tag} \tag{20}$$

4 Experiments and Results

4.1 Dataset and Experimental Settings

We use the legal dataset mentioned in section 3.1 to evaluate our proposed Triplet Extraction System. The dataset consists of 1750 fact descriptions of the drug-related criminal judgment documents downloaded from *China Judgments Online*. We split the dataset by a ratio of 4:1 to obtain the training set and the test set.

We utilize the pre-trained language model ***RoBERTa-wwm-ext, Chinese***[2] (Liu et al., 2019; Cui et al., 2019) for the encoder. The length of input sequence N is set to 512 and length of triplet sequence M is 10. The dimensions of the encoder representation and the hidden vector of the decoder are both 768. The weighting parameters γ, α and β are set to 0.5, 1 and 1 respectively.

4.2 Experimental Results and Analysis

The Precision, Recall and F_1-score of the extracted triplets are used as evaluation metrics. The equations of the evaluation metrics are as follows, where $correct\_num$, $predict\_num$ and $true\_num$ are the number of triplets extracted correctly, the number of triplets extracted by the system and the number of true triplets. We regard that a triplet is extracted correctly only if the beginning and end of the two entities and the relation of the triplet are all correct.

We experiment the performance of Legal Triplet Extraction System with the state-of-the-art method PNDec proposed by Nayak and Ng (2019) for joint entity and relation extraction. The main results on the legal dataset are shown in Table 1.

$$precision = \frac{correct\_num}{predict\_num} \tag{21}$$

$$recall = \frac{correct\_num}{true\_num} \tag{22}$$

$$F_1 = \frac{2 \times precision \times recall}{precision + recall} \tag{23}$$

[2]https://github.com/ymcui/Chinese-BERT-wwm

In Table 1, PNDec denotes the baseline method proposed by Nayak and Ng (2019), which implements the joint entity and relation extraction by a bi-directional LSTM encoder and a pointer network-based decoder. +BERT Enc denotes the model replacing the encoder by the pre-trained language model, which refers to *RoBERTa-wwm-ext, Chinese*. On this basis, +STL is the model adding Sequence Tagging Layer, which enables the encoder to learn entity boundary information better. Our model enhances encoder with the legal lexicon feature. The results illustrate the superiority in the abundant prior knowledge of BERT with the increase of 8.1% in F_1-score. Furthermore, they indicate that the auxiliary task of sequence tagging conduces to extracting the triplets better since there are improvements of 1.4%, 1.5% and 1.4% in Precision, Recall and F_1-score, respectively. The proposed enhancement of legal feature on encoder further advances the joint extraction in Chinese legal domain with an improvement of 1% in Precision.

Models	**Precision**	**Recall**	**F_1-score**
PNDec	80.6	67.8	73.6
+BERT Enc	84.4	79.1	81.7
+STL	85.8	80.6	83.1
Our Model	86.8	80.6	**83.6**

Table 1: The main results of on legal dataset. (%)

Effectiveness of the Auxiliary Sequence Tagging Task:To further investigate the importance of the auxiliary sequence tagging task, we evaluate the performance of adding a sequence tagging layer to the joint model. We carry out on both the models with BERT encoder and BiLSTM encoder. For the model with BERT encoder, we use a multi-label classifier to tag the sequence, whereas a Conditional Random Fields (CRF) decoder (Lafferty et al., 2001) is used on sequence tagging for the model with BiLSTM encoder. The results on the legal dataset are shown in Table 2. Two sequence tagging schemes are utilized in the experiment. +STL with type denotes the sequence tags consist of entity boundary and entity type, while +STL w/o type denotes that the tags consist of only entity boundary.

Models	**Precision**	**Recall**	**F_1-score**
BiLSTM Enc	80.6	67.8	73.6
BiLSTM Enc+STL with type	79.8	71.1	**75.2**
BiLSTM Enc+STL w/o type	80.5	69.5	74.6
BERT Enc	84.4	79.1	81.7
BERT Enc+STL with type	86.8	78.7	82.5
BERT Enc+STL w/o type	85.8	80.6	**83.1**

Table 2: The results of models trained with different sequence tagging tasks. (%)

The results in Table 2 suggest that the auxiliary sequence tagging task is effective for the model to learn the features of entities. There are improvements in both the models with different encoders by adding the sequence tagging task. For the model with BERT encoder, the F1-score of the tagging scheme with entity type has an increase of 0.8%, moreover the scheme without entity type improves by 1.4% compared with the single model. For the model with BERT encoder, the tagging scheme with entity type has more effective performance with an increase of 1.6%.

Effectiveness of the Legal Lexicon Feature: For the purpose of evaluating the effectiveness of the legal feature enhancement on the encoder, we compare the model performance on the triplet extraction models without and with legal lexicon feature enhancement. We experiment the models with a sequence tagging layer based on both BiLSTM encoder and BERT encoder. The results on the legal dataset are summarized in Table 3.

The experimental results show that the F_1-score of the models with legal feature has an improvement of 0.8% on BiLSTM-based model and 0.5% on BERT-based model. It illustrates that the proposed method of legal feature enhancement benefits the legal triplet extraction especially in Precision. The legal feature

is built up based on the specific of legal domain and information extraction task. In Chinese drug-related judgment documents, the name of drugs is a representative feature which assists in the recognition of drug entities. Consequently, the integration of the lexicon feature of drug names improve the extraction precision of the model.

Models	Precision	Recall	F_1-score
BiLSTM-based	79.8	71.1	75.2
BiLSTM-based+ Legal Feature	81.8	70.9	**76.0**
BERT-based	85.8	80.6	83.1
BERT-based+Legal Feature	86.8	80.6	**83.6**

Table 3: The results of the legal lexicon feature enhancement on the encoder. (%)

4.3 Comparison with Pipelining Method

We make a comparison of the performance of our joint learning-based method and the traditional pipelining method. We conduct two pipeline-based experiments on the legal dataset to prove the efficiency and the performance of our Legal Triplet Extraction System. The results are shown in Table 4.

Models	Precision	Recall	F_1-score
Pipelining	22.7	91.3	36.3
Pipelining-Rules	72.6	88.1	79.6
Our Model	85.8	80.6	**83.1**

Table 4: The comparison of using pipelining method and our model on triplet extraction. (%)

Pipelining in Table 4 denotes the method simply combining the two steps of triplet extraction, i.e. the entity recognition and relation classification together without any training constraints. On this basis, Pipelining-Rules denotes the method which conducts the triplet extraction considering the redundancy negative entity pairs and trains the relation extraction model with negative sampling. These two methods utilize *BERT-Base, Chinese*[3] for entity recognition and *RoBERTa-wwm-ext, Chinese* for relation extraction. A process of filtrating the entity pairs by pre-defined rules is carried out between the two steps so as to decrease the negative entity pairs. In contrast, our model is based on joint learning method and doesn't need any entity filter rules or training constraints. There is an absolute increase in Precision and F1-score has improved by 3.5% compared with the pipelining method filled with rules. We choose the joint model without legal lexicon feature for the sake of fairness.

5 Conclusion

In this paper, we introduce a triplet extraction system to extract the triplets from the unstructured crime judgment documents. We explore the pre-trained language model for the system. The system extracts the entities and the semantic relations jointly with the assistance of legal feature enhancement. In addition, we manually annotate a dataset for information extraction in Chinese legal domain, in order to train supervised models and evaluate the model performance. Experiments show that the adoptions of legal feature enhancement and multi-task learning framework promote the performance of legal triplet extraction. For future work, we will explore more effective legal features for the legal triplet extraction system. Information extraction on other crimes will be carried out as well.

Acknowledgments

This research is supported by the National Key Research and Development Program of China (No.2018YFC0830603). Finally, we would like to thank the anonymous reviewers for their valuable comments.

[3]https://github.com/google-research/bert

References

Jason P. C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. 2019. 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*, pages 1409–1418.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *ACL*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *ACL*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv: Computation and Language*.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-bert: Enabling language representation with knowledge graph. In *AAAI*, pages 2901–2908.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. *arXiv preprint arXiv:1803.02893*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. *arXiv preprint arXiv:1601.00770*.

Makoto Miwa and Yutaka Sasaki. 2014. Modeling joint entity and relation extraction with table representation. In *EMNLP*.

Tapas Nayak and Hwee Tou Ng. 2019. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. *arXiv preprint arXiv:1911.09886*.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Matthew E Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. 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)*, pages 43–54.

Xiang Ren, Zeqiu Wu, Wenqi He, Meng Qu, Clare R. Voss, Heng Ji, Tarek F. Abdelzaher, and Jiawei Han. 2017. Cotype: Joint extraction of typed entities and relations with knowledge bases. *Proceedings of the 26th International Conference on World Wide Web*.

Changzhi Sun, Yeyun Gong, Yuanbin Wu, Ming Gong, Daxin Jiang, Man Lan, Shiliang Sun, and Nan Duan. 2019. Joint type inference on entities and relations via graph convolutional networks. In *ACL*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *ArXiv*, abs/1409.3215.

Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2019. A hierarchical framework for relation extraction with reinforcement learning. In *AAAI*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Haoyu Wang, Ming Tan, Mo Yu, Shiyu Chang, Dakuo Wang, Kun Xu, Xiaoxiao Guo, and Saloni Potdar. 2019. Extracting multiple-relations in one-pass with pre-trained transformers. *arXiv preprint arXiv:1902.01030*.

Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. 2019. A novel cascade binary tagging framework for relational triple extraction. *arXiv: Computation and Language*.

Shanchan Wu and Yifan He. 2019. Enriching pre-trained language model with entity information for relation classification. *arXiv preprint arXiv:1905.08284*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Bowen Yu, Zhenyu Zhang, Jianlin Su, Yubin Wang, Tingwen Liu, Bin Wang, and Sujian Li. 2019. Joint extraction of entities and relations based on a novel decomposition strategy. *ArXiv*, abs/1909.04273.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation Classification via Convolutional Deep Neural Network. *Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014)*, (2011):2335–2344.

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. 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)*, pages 506–514.

Daojian Zeng, Haoran Zhang, and Qianying Liu. 2020. Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning. In *AAAI*, pages 9507–9514.

Zhuosheng Zhang, Yu-Wei Wu, Junru Zhou, Sufeng Duan, Hai Zhao, and Rui Wang. 2020. Sg-net: Syntax-guided machine reading comprehension. In *AAAI*.

Suncong Zheng, Yuexing Hao, Dongyuan Lu, Hongyun Bao, Jiaming Xu, Hongwei Hao, and Bo Xu. 2017a. Joint entity and relation extraction based on a hybrid neural network. *Neurocomputing*, 257:59–66.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017b. Joint extraction of entities and relations based on a novel tagging scheme. In *ACL*.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212.

Yuying Zhu and Guoxin Wang. 2019. Can-ner: Convolutional attention network for chinese named entity recognition. 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)*, pages 3384–3393.