Soft Computing (2021) 25:11817–11836
https://doi.org/10.1007/s00500-021-05971-3

APPLICATION OF SOFT COMPUTING

Automatic extraction of associated fact elements from civil cases based on a deep contextualized embeddings approach: KGCEE

Hongsong Dong[1,2] · Fengbao Yang[1] · Xiaoxia Wang[1] · Yufeng Sun[1]

Accepted: 10 June 2021 / Published online: 29 June 2021

Abstract
Automatic factor extraction is to extract the relevant facts from the case to assist the judge in the intelligent decision-making of civil disputes. Previously, the existing methods mainly focus on context-free word embeddings to deal with extraction tasks in the field of law, which cannot get a better semantic understanding of the text and in turn leads to an adverse extraction performance. Therefore, in this paper, a deep contextualized embeddings-based method called the knowledge-guided civil case fact elements extraction (KGCEE) model to automatically extract civil fact elements in the civil case domain is proposed. This approach is mainly based on the RoBERTa, but a few techniques make a more powerful model. Firstly, the model is retrained with civil domain data to provide more sensitive weight to initialize the model parameters in the downstream task. Secondly, the extraction is transformed into a sentence pairs task and we have incorporated data by leveraging label information to improve the generalization ability of the model. Thirdly, at the beginning of the KGCEE, we propose to inject part-of-speech information to the word embeddings to enhance the ability to capture the semantic and syntactic information, which aims to obtain better text representations. Finally, the KGCEE method is evaluated under civil domain data such as marriage and family, labor disputes and loan contracts originally from Chinese AI and Law (CAIL). The experimental results demonstrate that our KGCEE method outperforms other context-free word embeddings methods and other traditional transformer-based methods.

Keywords Civil domain · Fact elements · Fine-tuned RoBERTa model · Knowledge-guided civil case elements extraction · Deep contextualized embeddings · Sentence pairs

1 Introduction

With the rapid development of the Chinese economy, people in China are getting involved in different civil disputes, such as marriage family disputes, contract disputes, and labor disputes. Traditionally, civil disputes are generally solved by judges, which requires judges to consult cases and summarize key information to make decisions on disputes. However, due to a large number of cases and few judges, the cases will be piled up, resulting in a waste of resources. Automatic elements extraction in civil cases refers to extract the important facts or contents automatically in the case description with artificial intelligence methods. The results of the case fact elements extraction can be used in the actual business needs of the judicial field such as the case summary, the push of interpretable cases and the recommendation of relevant knowledge, which provides enough supports and assists judges or lawyers in resolving civil disputes.

A large portion of civil cases is written as unstructured text (Zablith and Osman 2019), therefore, natural language processing (NLP) methods (Chen and Luo 2019; Zhang et al. 2019a, b) such as text classification-based method (Burdisso et al. 2019) have been investigated to acquire a better understanding of fact elements extraction. Previous

✉ Fengbao Yang
yfengb@163.com

Hongsong Dong
dong_hs@126.com

Xiaoxia Wang
wangxiaoxia@nuc.edu.cn

[1] School of Information and Communication Engineering, North University of China, Taiyuan, China

[2] Department of Computer Science, Luliang University, Luliang, China

studies have explored the use of machine learning technologies to extract associated law factors (Liu et al. 2015) or charge elements to predict judicial decisions. Some researchers (Lin et al. 2012) exploit a conditional random fields (CRF) model to extract 21 kinds of law features and two kinds of crimes.

More recently, because of the development of language models, e.g., skip-gram and continuous bag-of-word (CBOW) (Mikolov et al. 2013), words can be trained as semantic word vectors on a large corpus and can be used as input of the neural network, which promotes the application of deep learning in text classification. Researchers focus on deep learning methods (Hu et al. 2018), e.g., recurrent neural network (RNN) (Luo et al. 2017) or convolutional neural network (CNN) (Zhao et al. 2019) to model the relation between law elements and the final decision, in which related articles are extracted for the final charge prediction task. In our earlier work (Dong et al. 2020), we extract associated law factors and charge elements on multi-label category combining CNN with bidirectional long short-time memory (BiLSTM) (Hochreiter and Schmidhuber 1997) and achieves competitive performance.

However, current deep neural network-based works for legal factor extraction of criminal cases are mainly focused on employing context-free word embeddings (Mikolov et al. 2013) to produce vectors, which is a single representation of each word without encoding word according to the context in which the word occurs. It was not until the context-aware word embeddings came along that this changed. BERT (Bidirectional Encoder Representations from Transformers) model (Devlin et al. 2019) utilizes contextualized word embeddings pre-trained on large corpora in an unsupervised manner (Moradi et al.2020; Gonzalez et al. 2020). The embeddings learned at the pre-training stage take into the context information and are used thereafter for different downstream tasks. This mechanism enables BERT to achieve outstanding results on eleven NLP tasks, and even outperforms human beings for some tasks. RoBERTa (Liu and Guo 2019) is an optimized method of BERT and produces state-of-the-art results on the widely used NLP benchmark.

Although promising results have been obtained by deep learning systems for criminal cases, only a few efforts have focused on Chinese civil cases. One potential reason is that civil cases are more complex than criminal cases. When dealing with criminal cases, there are mainly two kinds of elements: article and charge needed to consider. While making a judicial decision on a civil case (e.g., divorce case), there are many factors to consider, such as whether the couple has children, whether they have joint property, whether they have separated for two years, whether they have children out of wedlock, whether they have joint debts, etc.

Our research aims to investigate the usefulness of RoBERTa, a state-of-the-art deep contextualized language model for fact elements extraction of civil text. However, there are some challenges to use RoBERTa directly in civil fact elements extraction. First, the judicial domain is special. If the original language model of Chinese wiki data is fine-tuned directly and transferred to the legal domain data for civil fact elements extraction, some legal terms are insufficient to be understood, and unable to complete the transfer of knowledge well, which affects the final extraction effect. Second, for one civil case, there is more than one element (label), maybe a dozen or more, if the extraction is made by direct multi-label classification, the generalization ability is relatively poor since no additional information is used and the training data is limited. Last, the deep contextualized language representation models (e.g., BERT, RoBERTa) can accurately capture semantic properties of words, but the ability to capture the semantic and syntactic information of words needs to be improved.

To address these issues, we introduce a knowledge-guided civil case fact elements extraction (KGCEE) model, which is shown in Fig. 1. The overall goal can be broken down into three subsections. First, unsupervised sentence knowledge is collected from CAIL as domain data for retraining. It produces sensitive parameters that facilitate the extraction task. Second, the extraction task is transformed into a sentence pair task and we generate supervised sentence pairs knowledge by leveraging label information to construct robust word embeddings for the sake of elevating the generalization ability. Moreover, part-of-speech (POS) information (Rezaeinia et al. 2019) is injected into the fine-tuning stage of the model to improve the semantic and syntactic understanding of the text.

In conclusion, the main contributions of the paper can be summarized as follows:

A knowledge-guided civil case fact elements extraction (KGCEE) model is designed, which is capable of extracting elements of specific civil cases. Compared to other baseline models, the KGCEE model achieves the best results on all metrics.

The design of unsupervised sentence knowledge to retrain the method elevates the final extraction results. The exploitation of supervised sentence pairs knowledge provides more training data, producing results that enhance the generalization ability of the model.

The word embeddings incorporate with POS information learns to capture a large amount of semantic and syntactic information of words.

The rest of the paper is organized as follows: related works are given in Sect. 2. Section 3 elaborately describes the proposed method. Section 4 introduces the data we

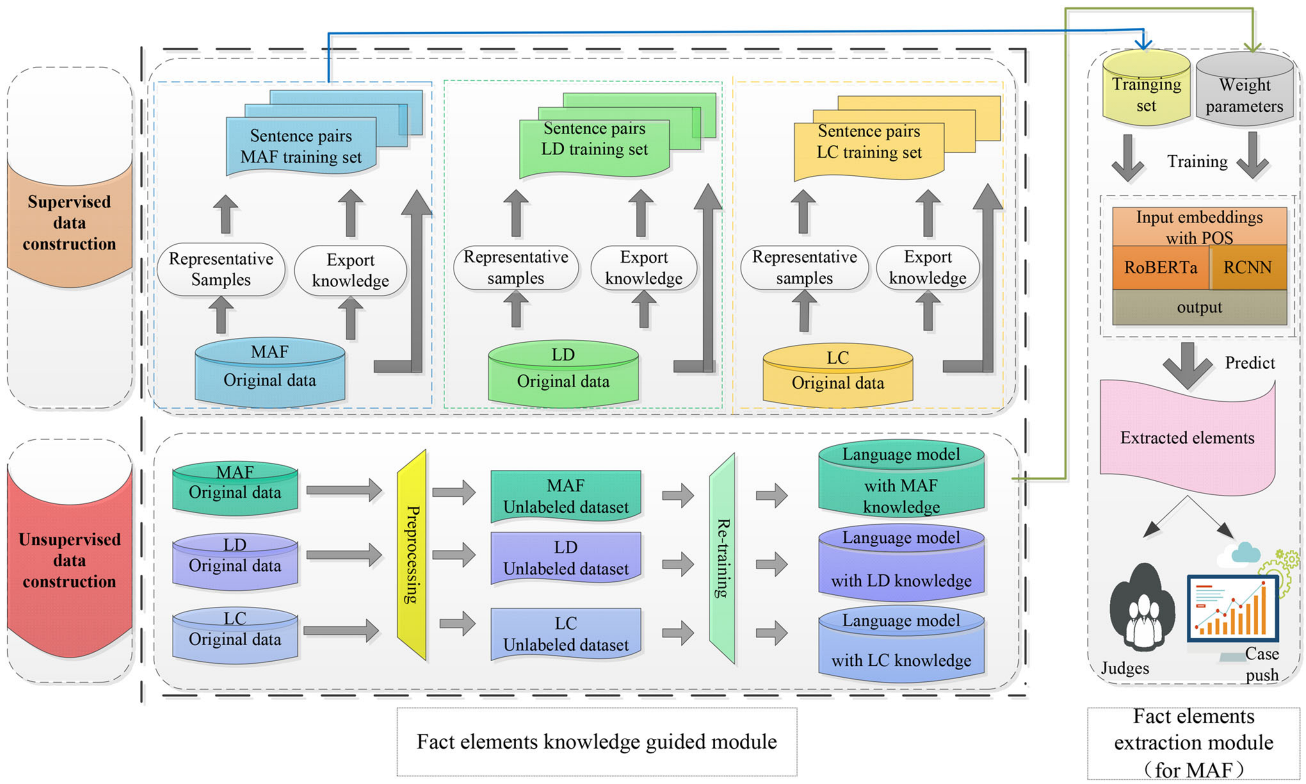


Fig. 1 The framework is divided into two modules: a fact element knowledge-guided module and a fact element extraction module. A fact elements knowledge-guided module consists of two parts: unsupervised data knowledge construction and supervised sentence pairs construction. The fact elements knowledge-guided module is to provide constrained weight parameters and training data for the fact elements extraction module. The fact elements extraction module is responsible to extract civil fact elements, which provides recommendations for judges or act as interpretable case push. Here, MAF, LD and LC refer to marriage and family, labor disputes and loan contracts. The details are shown in Sec.3. In the figure, for the sake of space, we only show the connection between MAF knowledge and the corresponding extraction module (the connection is shown by the blue and green arrows in the figure), which is the same for the connection between the other two knowledge LD, LC

used and reports the experiments. Finally, the conclusion are discussed in Sect. 5.

2 Related works

2.1 Text representation

Our work is firstly associated with text representation (Sinoara et al. 2019). One hot vector is a straightforward binary encoding that every word is composed of 1 corresponding to the index of the vocabulary and a series of 0. It solves the basic problem of distinguishing words but suffers from data sparseness and ignores the similarities between words. Latent semantic analysis (LSA) (Kim et al. 2020) and latent Dirichlet allocation (LDA) (Ekinci and Omurca 2020) are kinds of models based on the distributional hypothesis that a word can be inferred from the surrounding words, similar words appear in similar contexts. The distributional representation is to get the semantic representation of the word. Word2vec methods such as CBOW and skip-gram (Mikolov et al. 2013) are a distributed representation-based model, which generates a dense, low-dimensional, continuous vector. Each dimension of the vector represents underlying grammatical or semantic feature of the text.

Evidence suggests that pre-trained word embeddings can boost performance on a range of NLP tasks (Elnagar et al. 2020; Gargiulo et al. 2019). However, they are mainly based on the word2vec word embedding method which is non-contextual and the same word in different contexts is encoded into a fixed vector.

Recently, contextualized word embedding methods such as ELMo (Peters et al. 2018), ULMFiT (Howard and Ruder 2018) and BERT (Devlin et al. 2019) are devised by injecting the context information into the embeddings of the word. ELMo (Peters et al. 2018) is an RNN-based language model that utilizes the forward and backward LSTMs (Hochreiter and Schmidhuber 1997) independently, context information cannot be extracted from both directions at the same time, which is essentially the concatenation of two unidirectional models. ULMFiT (Howard and Ruder 2018) employs three layers of unidirectional LSTMs as the model. In summary, compared with BERT, the methods mentioned above do not make good use of context information since they own different architecture.

BERT (Devlin et al. 2019) is the encoder of the transformer model (Vaswani et al. 2017). Unlike the aforementioned methods, BERT uses a bidirectional deep transformer model in which each layer can take advantage of bidirectional contextual information. Because of its powerful representational ability, BERT has achieved great improvement in medical science (Li et al. 2020; Zhang et al. 2019a, b; Du et al. 2020; Sun et al. 2020;), image processing (He et al. 2020), construction (Fang et al. 2020).

RoBERTa (Liu and Guo 2019) is an optimized version of BERT, which shares the same architecture with BERT, but achieves better performance than BERT due to the effective training strategy. At present, there are few applications of RoBERTa. The proposed KGCEE model starts from the retraining process of the Chinese RoBERTa model. We first generate unsupervised sentences from the original data set from Chinese AI and Law (CAIL) as the input of the RoBERTa to retrain the Chinese RoBERTa model. Thus, civil domain data information is incorporated and a language model of Chinese civil texts is obtained.

2.2 Deep learning

Our work is also related to the classification task of NLP for the reason that the extraction of the element is taken as a classification task by many current works. As one of the most important NLP subtask, the text classification task has many applications, such as sentiment classification (Chen et al. 2020a, 2020b; Xia et al. 2020; Fan et al. 2020), judgment prediction (Zhong et al. 2018), rumor detection (Alkhodair et al. 2020; Liu et al. 2019a, b) and elements extraction.

Conventional deep neural networks such as text convolutional neural networks (Kim 2014; Guo et al. 2019), recurrent neural networks (Greff et al. 2017), and recurrent convolutional neural networks (Liu and Guo 2019) are proposed by researchers to implement text classification task. Text-CNN has a strong ability to extract the superficial features of the text. It is widely applied and effective in the field of short text, such as the search and dialog area. However, for long text, CNN mainly extracts features by filter window, which is limited to long-distance modeling and insensitive to word order. RNN-based models, for instance, LSTM belongs to a sequential structure, though the structure contains the temporal information of the sequence, the output at the current step is related to the hidden layer at the previous step, which means that the output at each step cannot be computed in parallel, resulting in low efficiency of RNN calculation.

In 2018, the emergence of BERT promotes the great development of natural language processing, and BERT achieves significant improvements on classification tasks. Compare to traditional deep neural networks, BERT has the following advantages. First, there is no problem of forgetting, the information about the current word containing other words does not depend on the distance, but on the correlation between them. Second, when calculating the current word, it takes the left and right context information about the word into account. Last, time sequence information is taken into consideration and can be calculated in parallel. RoBERTa employs the training strategy such as training the model with bigger batches, longer sequences, over more data and the dynamic masking mechanism, which exceeds BERT model on many NLP tasks. The dynamic masking is relative to static masking, static masking refers to randomly mask tokens during pre-training, then leaves the masked tokens unchanged. Dynamic masking is designed to take full advantage of the data, which produces different masks for the same sentence every time. Before the training samples are transferred to the model, mask operation is carried out first. By copying several copies of training data, each data has a different mask operation, and these data are, respectively, input into different epoch for training. Dynamic masks perform better than static masks when pre-training larger data for longer periods. Given this, RoBERTa is employed to complete the extraction task. To be specific, based on RoBERTa, we generate the supervised sentence pair from the original data set from CAIL as the input supervised data of the RoBERTa model, which is fine-tuned to conduct the extraction task.

2.3 Automatic extraction of fact elements

At present, researchers have explored rule-based technology to factor extraction in the field of law. (Schilder et al. 2005) extract the information of time elements that might be associated with events by a finite state sensor. They use UIMA tools to extract the grammatical structure information and temporal elements information from the legal text, and use it as the basis for matching the given description of the specific case. Bartolini et al. (2004) develop a SALEM (Semantic Annotation for Legal Management)-based legal text mining system to extract semantic elements of legal paragraphs. In the method, a two-stage strategy is adopted to extract semantic elements of legal texts. The above method requires the manual definition of semantics, syntax, business rules and rule templates, which requires a lot of human and material resources.

Some researchers also apply machine learning methods to factor extraction in the field of law. Li et al. (2019a, b) develop a text mining programming language based on

TML (Kao and Poteet 2007) to formally describe and extract the semantic information of legal elements, which serves as input to train the Markov logic network for prediction. In their follow-up work (Li et al. 2019a, b), a cognitive computing framework is proposed to extract the elements of cases. In the framework, the first-order logical basis is used to represent the semantic information of the legal elements in the judicial documents in a formal way, and then the legal elements are extracted by combining rule-based and machine learning methods. Finally, the prediction model is generated and trained to extract the legal elements and the result elements.

With the development of big data and deep learning methods, deep neural networks have surpassed classical machine learning methods in many fields, and the judicial domain is no exception (Li et al. 2019a, b; Yan et al. 2017). Li et al. (2019a, b) propose a mechanism for defining focal event elements and a two-level labeling method, which can automatically extract focal event elements from the document. The BiLSTM neural network is used to obtain the context features, and finally, the combination of CRF can constrain the information between tags to obtain the predefined event element tag types. Compared with machine learning methods, deep learning methods can automatically extract text features through their powerful text representation ability, without the need for the manual design of features, and has stronger generalization ability. Our method is mainly based on the deep learning model RoBERTa, a state-of-the-art deep contextualized language model to conduct a Chinese civil case fact elements extraction task.

3 Method

3.1 Overview

The structure of the proposed KGCEE is shown in Fig. 2, which consists of two modules: a fact elements knowledge-guided module and a fact elements extraction module. The two modules are connected by data and parameters flow, in which the former provides meaningful extra information (constrained weight parameters and training data) to the latter. The design of the structure enables the proposed KGCEE to extract complex civil fact elements. In the following, two modules are introduced in detail.

3.2 Fact elements knowledge-guided module

The fact elements knowledge-guided module consists of two components: unsupervised civil case sentences knowledge and supervised civil case sentence pairs knowledge. The unsupervised civil case sentence knowledge is developed to retrain the RoBERTa model and provide constrained weight parameters which are used to initialize model parameters in the downstream fine-tuning task (fact elements extraction task). This is to make full use of the language knowledge that RoBERTa learned in the retraining process and transfer it to the learning of downstream tasks, which makes it sensitive to the civil domain data.

3.2.1 Unsupervised civil case sentences knowledge

In fact elements knowledge-guided module, we prefer to generate knowledge that retains most of the information from the source civil text, the unsupervised civil case sentences knowledge S_u is generated by original CAIL fact description S, in which the former is made up of sentences from the latter. The knowledge generated is mainly used to guide and constrain (retrain) the original language model (RoBERTa-base, Chinese) through the Masked Language Model (MLM) task. The original Chinese RoBERTa language model that has been retrained is called the language model of Chinese civil knowledge. The retrained model learns the prior knowledge of the civil domain, and the parameters of the model are more sensitive to the specific civil case features, which aims to lay a solid foundation for the subsequent fine-tuning of the extraction (classification) task.

3.2.2 Supervised civil case sentence pairs knowledge

The supervised civil case sentences pairs knowledge is generated by the original CAIL dataset, which serves as data to conduct extraction tasks. In actual scenarios, one instance has more than one CFE, maybe a dozen or more, and it is a multiple label classification task. If conduct multi-label classification directly, it will suffer from the problem of weak generalization. One of the reasons for poor performance is that it tries to map input instance to target labels directly but fail to use more information and there are few instances for a certain fact element (label). One solution is that the multiple label prediction task is transformed into a sentence pairs task and extra knowledge can be introduced to alleviate the problem. The knowledge includes instance information, label information, and keywords from each label.

Especially, the supervised civil case sentence pairs knowledge is generated as follows:

(1) Representative instances: representative instances are manually collected by three Ph.D. students in law, for each label DV_m, there are representative sentences $S^{(m)} = \left\{ S_1^{(m)}, S_2^{(m)} \ldots S_{n_m}^{(m)} \right\}$ come from the

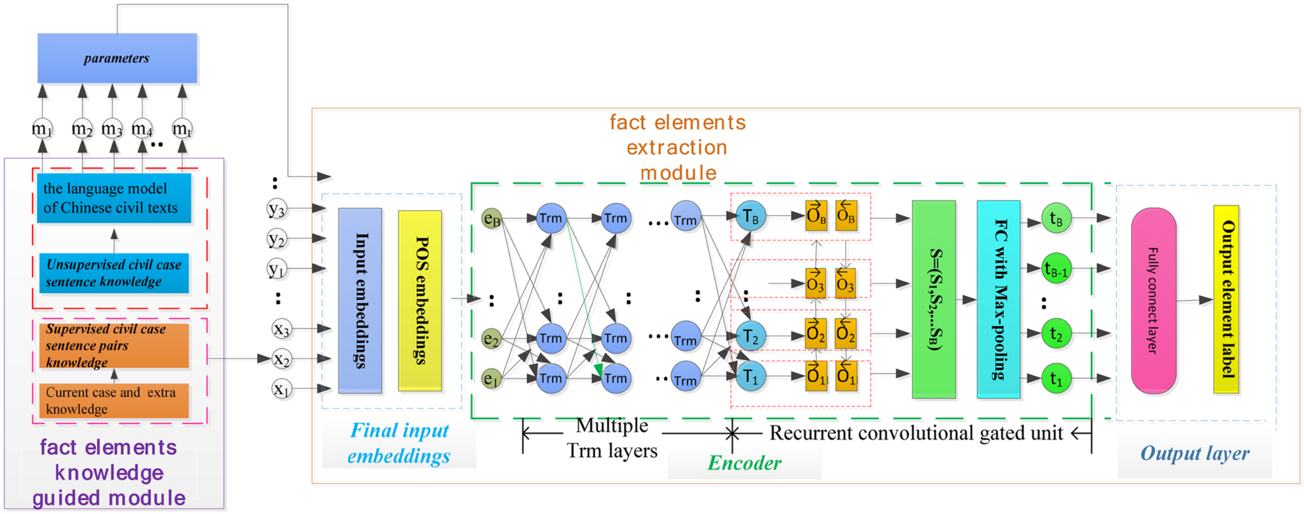

parameters
m1
m2
m3
m4
mt
the language model of Chinese civil texts
Unsupervised civil case sentence knowledge
Supervised civil case sentence pairs knowledge
Current case and extra knowledge
fact elements knowledge guided module
y3
y2
y1
x3
x2
x1
fact elements extraction module
Input embeddings
POS embeddings
Final input embeddings
eB
e2
e1
Trm
Multiple Trm layers
TB
T2
T1
O→B
O←B
O→3
O←3
O→2
O←2
O→1
O←1
S=(S1,S2,...SB)
FC with Max-pooling
tB
tB-1
t2
t1
Recurrent convolutional gated unit
Encoder
Fully connect layer
Output element label
Output layer

Fig. 2 The structure of the KGCEE model. The model consists of two modules: a fact elements knowledge-guided module and a fact elements extraction module. The fact elements knowledge-guided module is to constrain the KGCEE model and provide training data for the fine-tuning stage

civil fact descriptions $S = \{S_1, S_2, \ldots S_n\}$, while $S^{(m)} \subseteq S$,n_m is the number of sentences chosen for label DV_m, and n is the number of total instances in the original data.

(2) Export knowledge: there are also some sentences which can describe or are closely related to the label. Those sentences form an extra export knowledge $\mathrm{K}^{(m)} = \left\{k_1^{(m)}, k_2^{(m)}, \ldots, k_{l_m}^{(m)}\right\}$, which is taken from the original training sample, or a description of the tag DV_m, the content of tag DV_m or a combination of top keywords corresponding to the tag. l_m is the number of sentences to describe the label DV_m.

(3) With the representative sentences $S^{(m)}$ or export knowledge $K^{(m)}$, and the present sample S_m, a sentence pair $P_m = <$sentence-A, sentence-B, label$>$ is constructed. Here, $p_m = <S^{(m)}$ or $\mathrm{K}^{(m)}$, S_m, label$(0,1)>$, sentence-A comes from $S^{(m)}$ or $K^{(m)}$, sentence-B comes from the present training sample S_m. label(0,1) denotes whether the label information is associated with the sample. A sentence pair is also denoted as $(x_1, x_2, x_3, \ldots y_1, y_2, y_3, \ldots)$ in Fig. 2 for convenience of expression.

(4) In order to complete the sentence pairs classification task, positive and negative samples need to be constructed. If the present instance is associated with the representative sentences $S^{(m)}$ or export knowledge $K^{(m)}$, then label$(0,1)$ equals 1 and it can be treated as a positive sample, otherwise, it is a negative sample.

3.3 Fact elements extraction module

3.3.1 Input embeddings with POS information

In NLP, one of the important uses of POS tagging is disambiguation, that is, different uses of some words represent different meanings. The main motivation for introducing POS can enhance better text representations through word sense disambiguation, thus improving the result of fact elements extraction. The POS also gives a large amount of information about a word and its neighbors, syntactic information of words (verbs, adverbs, nouns, adjectives, etc.) and similarities and dissimilarities between them (Rezaeinia et al. 2019).

The input embeddings of RoBERTa P_{in} is given by adding the segment embeddings P_{seg}, token embeddings P_{tok}, and position embeddings P_{loc}:

$$P_{\text{in}} = P_{\text{seg}} + P_{\text{tok}} + P_{\text{loc}} \tag{1}$$

The POS tags of words are obtained by Hagongda (http://www.ltp-cloud.com/). And then converted to vectors P_{pos} and added to input embeddings P_{in} to get the final input embeddings P_{fin} (which is e_1, e_2, … e_{B} in Fig. 2).

The generation of POS vectors is as follows:

(1) The sentence pairs are first tokenized and denoted as $P_F \in R^{B*F}$, B is the batch size and F is the sequence length.

(2) Hagongda parser is employed to generate POS tags matrix $P_c \in R^{B*F}$.

(3) Then $P_c \in R^{B*F}$ is organized to a one-dimension vector by the Flat Operation (FO):

$$P_o = FO(P_c) \in R^{(B*F)*1} \tag{2}$$

FO is a transformation that changes the matrix $P_c \in R^{B*F}$ to a column vector $P_o \in R^{(B*F*1)}$.

(4) P_o is encoded to a series of binary vectors $P_V \in R^{(B*F)*V}$, V is the vocabulary size of the dictionary built by Hagongda parser.

(5) Last, a linear transformation(LT) operates on the binary matrix P_V, and get POS vectors:

$$P_{\text{pos}} = Pv * W \tag{3}$$

$W \in R^{V*H}$ is the weight parameter learned in the fine-tuning stage, H is the hidden size from RoBERTa. Figure 3 shows the generation of POS vectors.

3.3.2 Encoder

The encoder consists of two components: fine-tuned RoBERTa model and recurrent convolutional gated unit. The fine-tuned RoBERTa with domain knowledge is applied to initialize the input word embeddings which are continuously finetuned during the whole training process; and recurrent convolutional gated unit is used to encode the sentences and capture key information in texts. The fine-tuned RoBERTa model uses the advanced transformer architecture Trm (Vaswani et al. 2017) like BERT does. But there is a tiny difference between the transformer architecture and BERT or (RoBERTa), the latter employs two consecutive fully connect layers without ReLU function. The details of Trm used in RoBERTa are shown in Fig. 4.

(1) *Multihead self-attention*

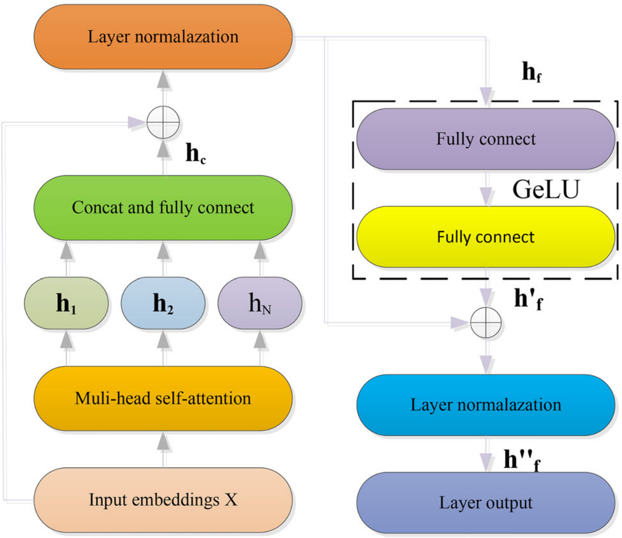


Fig. 4 Trm used in RoBERTa, which consists of three parts: multihead self-attention, layer Normalization (LN) and residual part, and two consecutive fully connected layers with GeLU.

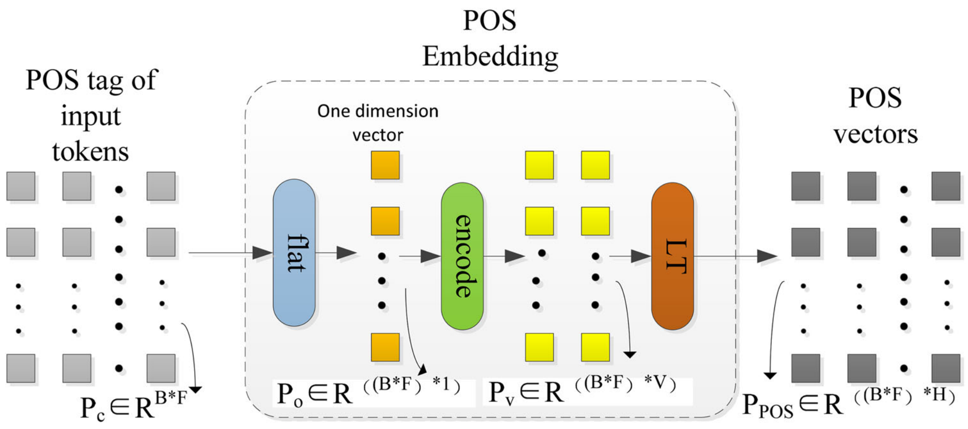
The input embeddings P_{fin} are reshaped to a two-dimensional matrix $X \in R^{B*F)*(N*H\prime)}$, N is the number of attention heads, $H\prime$ is the size of per attention head, and hidden size $H = N * H'$. The multihead self-attention mechanism is based on single head scaled dot-product attention, in which the latter can be depicted as a function that maps the input query matrix Q and key-value matrix (K–V) to an output. The final output is a weighted sum of values V, the weight is calculated by dot product function $f(Q,K) = Q * K^T$ measures the similarity between Q and K, three matrices are obtained by the linear transformation of the input embeddings $(Q,K,V) = XW^Q, XW^K, XW^V$).

For a single head h_i, the attention denoted as:

$$h_i = \text{attention}(Q_i, K_i, V_i) = \text{attention}(XW_i^Q, XW_i^K, XW_i^V)$$
$$= \text{soft max}\left(\frac{Q_i K_i^T}{\sqrt{N}}\right)V_i = \text{soft max}\left(\frac{XW_i^Q (XW_i^K)^T}{\sqrt{N}}\right)(XW_i^V) \tag{4}$$

where W_i^Q, W_i^K, and W_i^V are the weight parameters with the same dimension of $[N * H\prime, N * H\prime]$ need to train.

Fig. 3 The generation of POS embeddings. (the input tokens are first fed into the POS tagger of hagongda, then through the POS embedding and get the POS vectors)



Multihead self-attention means do the single self-attention N times, parameters $W^Q, W^K, W^V)$ are not shared between each head h_i (i = 1,2,…N), which allows the encoder to learn about different word dependencies within the same sentences. The self-attention mechanism gets a new representation of each word that takes into account contextual information. Multihead self-attention is to project the concatenation of N heads into a fully connected layer to get the same dimension of a single head:

$$\mathrm{MultiHead}(X) = \mathrm{Concat}(h_1, \ldots h_N)W^O \tag{5}$$

where $W^O \in R^{N*N*H\prime)*(N*H\prime)}$ and for the sake of representation,MultiHead(X) is denoted by h_c.

(2) *Layer Normalization (LN) and Residual*

In deep learning methods, normalization method such as batch normalization (BN) (Ioffe and Szegedy 2015) or layer normalization (LN) is employed to mitigate the problem of internal covariate shift (Ioffe and Szegedy 2015). During the training process, the distribution of input data of the hidden layer changes, which brings difficulties to the learning of the next layer of the network and causes the gradient disappearance problem. After multihead self-attention, LN is utilized to reduce the influence of internal covariate shift by fixing the input mean and variance of a layer of neurons (the input of each layer of neural network remains the same distribution).

LN is superior to batch normalization in that the former is less affected by the batch size, which is more practical. For instance, the batch size is highly dependent on the hardware environment GPU, only a small size can operate in limited hardware conditions, if the mean and variance of the data with a small batch size are obtained, it is bound to have a deviation from the total data with BN. This is not the case in LN. The application of LN is more extensive and less limited.

For BERT and RoBERTa, there are two basic models, the base and the large, both versions have more parameters and deeper networks, which can bring the gradient diffusion or explosion problem, and leads to degradation (He et al. 2016) without certain measures. Degradation is a problem that with the increase in network layers, the accuracy of the training set decreases, which is not an overfitting problem. A deep network cannot well optimized only with regularization, residual network is also necessary. Residual is to cope with degradation problem, which is caused by the increase in the network depth. The residual is introduced to retain as much information as possible from the original input embeddings X. The output of multihead self-attention h_c is projected into residual and LN layer:

$$h_f = LN(X + h_c) \tag{6}$$

(3) *Two consecutive fully connected layers with GeLU or called (Feedforward network with GeLU) (FFNWG)*

For the transformer-based attention model (Vaswani et al. 2017, after layer normalization, the feedforward module contains two linear transformations with a rectified linear unit (ReLU) in between is employed as $\mathrm{FFNWR}(h_f) = \max(h_f W_1, 0)W_2$. FFNWR is a feedforward network (FFN) with ReLU. While in BERT or ReBERTa, a tiny difference is that BERT or RoBERTa uses two consecutive fully connected layers without ReLU, the GeLU activation function is used instead as shown in Eq. 7, which is shown in the source code of BERT.

$$\mathrm{FFNWG}(h_f) = h_f' = [\mathrm{GeLU}(h_f W_1)]W_2 \tag{7}$$

where FFNWG means FFN with GeLU, $W_1 \in R^{H*H''}$ and $W_2 \in R^{H''*H}$, H'' is the extended dimension, which is a predefined hyperparameter as H, and $H''\rangle H$.

In Eq. (7), the first linear fully connected layer with W_1 of higher dimension to get more features. Then, W_2 is employed to reduce the dimension to the size of hidden size H. Dispensing with ReLU is to prevent further damage to the features. The reason is that ReLU makes the output of negative input to 0, and dimension reduction itself is a process of feature compression, with W_2 to reduce the dimension, there is no need to use ReLU. For the convenience of expression, FFNWG(h_f) is denoted by $h_f\prime$, after FFNWG, $h_f' \in R^{(B*F)*H}$. Then, the final residual and layer normalization is again employed to get the final output of this encoding layer:

$$h_f' = LN(h_f + h_f') \tag{8}$$

h_f' serves as the input of the next encode layer after several layers, the output of the RoBERTa is denoted as $T = (T_1, T_2, \ldots T_B)$, as shown in Fig. 2.

(4) *Recurrent convolutional gated unit*

As shown in Fig. 2, the recurrent convolutional gated unit contains two parts: the concatenation of output of the RoBERTa T and the output of the LSTM layers, a fully connected layer and a max-pooling operation. The recurrent convolutional gated unit aims to encode the input sentences (which is also the output of the fine-tuned RoBERTa) and extract core information t:

$$t = (t_1, t_2, \ldots t_B) = \max[(S_1, S_2 \ldots S_B) * W] \tag{9}$$

and $t \in R^{(B*H)}$ is the final output of the recurrent convolutional gated unit, $W \in R^{((H+2H_L)*H)}$ is the parameter of the fully connected layer, H_L is the hidden size of the LSTM layer, max[·] is the max pooling operation and S_i is the concatenation of $T_i \in R^{F*H}$ and output of LSTM $o_i \in R^{F*2H_L}$, which can be denoted as:

$$S_i = [\vec{o}_i, T_i, \overleftarrow{o}_i] \in R^{(F*(H+2H_L))} \tag{10}$$

B, F and H is the batch size, sequence length and hidden size of RoBERTa as mentioned before.

3.3.3 Output layer

In the output layer, the output of the recurrent convolutional gated unit t in Eq. (9) is reshaped and fed to the softmax function to classify the extracted civil fact elements, since our extraction task is transformed into a binary classification, the output is to predict whether the sample is associated with a certain label. The predicted result is shown as:

$$\hat{y} = \mathrm{softmax}(W^y t + b^y) \tag{11}$$

Here, W^y and b^y are the weight and bias parameters in the output layer.

The training objective of our proposed KGCEE is to minimize the cross-entropy between the predicted distribution $\hat{y}$ and the ground-truth label y. The loss function can be formalized as:

$$L = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\{y_i = j\} \cdot \log(\hat{y}_{ij})\right] \tag{12}$$

where m is the number of training samples, and $1\{Truth\} = 1, 1\{False\} = 0$.

4 Experiments

In this section, experimental setup including data sets, experiment parameters and evaluation metrics are described. Next, several complex experiments are conducted on the proposed KGCEE model under unsupervised data knowledge and sentence pair data generated from the original data set. Specifically, the first experiment is to assess the impact of retraining the proposed method with unsupervised civil case sentences knowledge domain data. The second experiment is to evaluate the function of POS component. Then the function of layer normalization and residual components are also investigated. Last, the proposed method is compared with other existing state-of-art methods, the generalization ability of the proposed method and error analysis are investigated.

4.1 Dataset

The original data used by the proposed model are from CAIL, sixty factor elements are designed by legal experts for case summary, related knowledge recommendation and predict judicial decisions on marriage and family cases, labor disputes cases and loan contracts cases. Figure 5 shows several instances of civil case fact elements extraction, which is from CAIL. The data format in Fig. 5 is used to complete the multi-label classification task. Since our method converts the multi-label classification task to the sentence pair task, it is necessary to introduce the data format of the sentence pair.

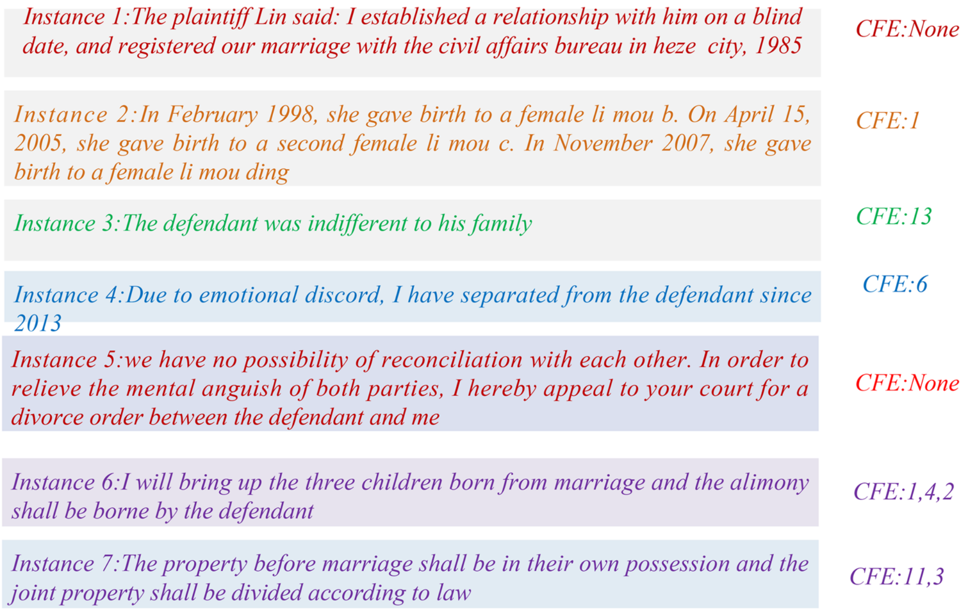
In Fig. 5, for instance, case 3 with CFE:13 and case 4 with CFE:6, the sentence pairs are as follows $<$ *(The defendant was indifferent to his family), (not fulfilling family obligations), 1* $>$, $<$ *(Due to emotional discord, I have separated from the defendant since 2013), (Separated after marriage),1* $>$ and the above constitute positive samples. The sentence pair of the positive sample consists of three components, current instance, positive label, and 1. While sentence pair of the negative sample also consists of three components, current instance, negative label, and 0. By cast to sentence pair task, it is easy to use knowledge including label information, instance information, keywords from each label. That knowledge is especially useful when there are few instances for each fact element (label), as result, the model will better generalize the knowledge learned from the training samples to the test set.

The original CAIL dataset, which consists of marriage and family (MAF), labor disputes (LD), loan contracts (LC). As shown in Fig. 6, there are 1269, 836, 635, and 2740 samples of civil fact descriptions in.

MAF, LD, LC and MLL (the mix of MAF, LD and LC). With the original fact descriptions, two stages of knowledge are generated for the retraining (constraining) stage and classification stage. For the retraining stage, the generated unlabeled data knowledge is made up of sentences, and each sentence is the training sample of the stage. There are 38,188, 32,632, 23,325 and 94,145 samples for four types of data. For the sentence pair classification stage, with the supervised civil case sentence pairs knowledge, generate 564,325, 429,685, 344,952 and 1,391,061 labeled samples for MAF, LD, LC and MLL, and each sample is a sentence pair with a single label.

The labeled samples are divided into four sets: training set, train-validation set, validation set and test set, the details are shown in Fig. 7.

Sometimes, depending on the training set and the validation set, we cannot judge whether the error of the training set and the validation set is caused by the difference between the data or the poor generalization ability of the algorithm itself. If the proportion of positive and negative sample distribution between the validation set and the training set is consistent, we would say that the gap between training error and validation error means there is a large variance problem, that the algorithm's just not



Instance 1:The plaintiff Lin said: I established a relationship with him on a blind date, and registered our marriage with the civil affairs bureau in heze city, 1985 CFE:None

Instance 2:In February 1998, she gave birth to a female li mou b. On April 15, 2005, she gave birth to a second female li mou c. In November 2007, she gave birth to a female li mou ding CFE:1

Instance 3:The defendant was indifferent to his family CFE:13

Instance 4:Due to emotional discord, I have separated from the defendant since 2013 CFE:6

Instance 5:we have no possibility of reconciliation with each other. In order to relieve the mental anguish of both parties, I hereby appeal to your court for a divorce order between the defendant and me CFE:None

Instance 6:I will bring up the three children born from marriage and the alimony shall be borne by the defendant CFE:1,4,2

Instance 7:The property before marriage shall be in their own possession and the joint property shall be divided according to law CFE:11,3

Fig. 5 The illustrative examples of the dataset from CAIL, CFE1 refers to married with children; CFE2: limit ability to raise children; CFE3:have community property; CFE4:pay maintenance fees; CFE6:separated after marriage; CFE11: personal property before marriage; CFE13:not fulfilling family obligations

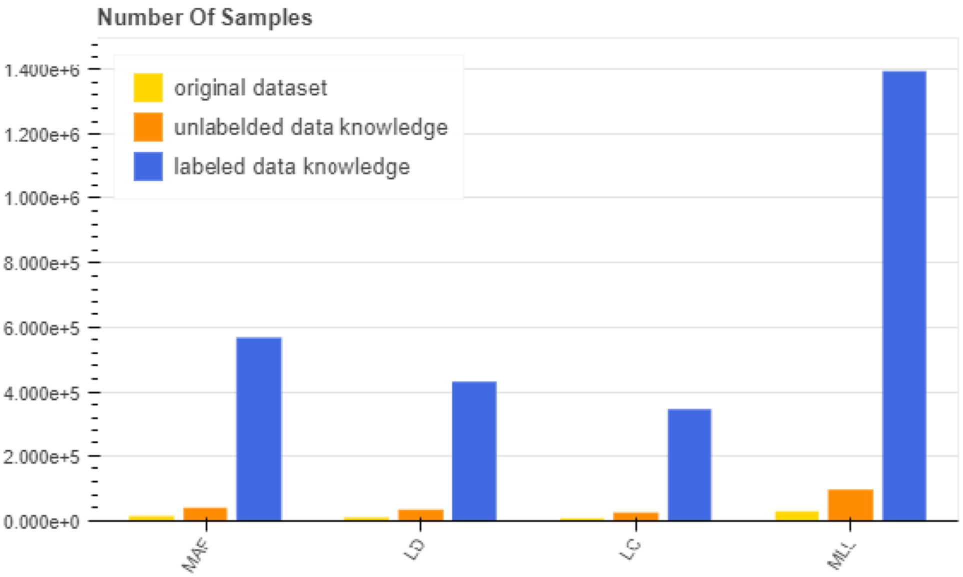


Fig. 6 Original dataset to generate unlabeled data knowledge and labeled data knowledge on four different kinds of data

generalizing well from the training set. But in the setting where the training set and validation set shares a big difference between the positive and negative distribution, we can no longer safely draw this conclusion. So to tease out these two effects, it will be useful to define a new piece of data, which we will call the training-validation set.

The train-validation set comes from the training set but is not used to train the model, which shares almost the same positive and negative distribution as the training set, but without backpropagation on it (the model has not seen the train-validation set before). The train-validation set and validation set share different positive and negative distribution but both without backpropagation. The gap between training set error and train-validation set error (variance), which illustrates the strength of model generalization (how much of the gap is since the model has not seen the data set). While the gap between the train-validation set and validation set reveals the data mismatch problem. Figure 8 is the positive and negative distribution of training set, train-validation set, validation set and test set of four typesdata MAF, LD, LC and MLL, from which we can see that train set and train-validation set shares almost the same positive and negative distribution, so does validation set and test set. Take MAF as an example, the positive rate of train, train-validation, validation and test set is 0.24, 0.25, 0.5, 0.51.

Fig. 7 Four types of datasets for classification task and each type of dataset concludes training, train-validation, validation and test set

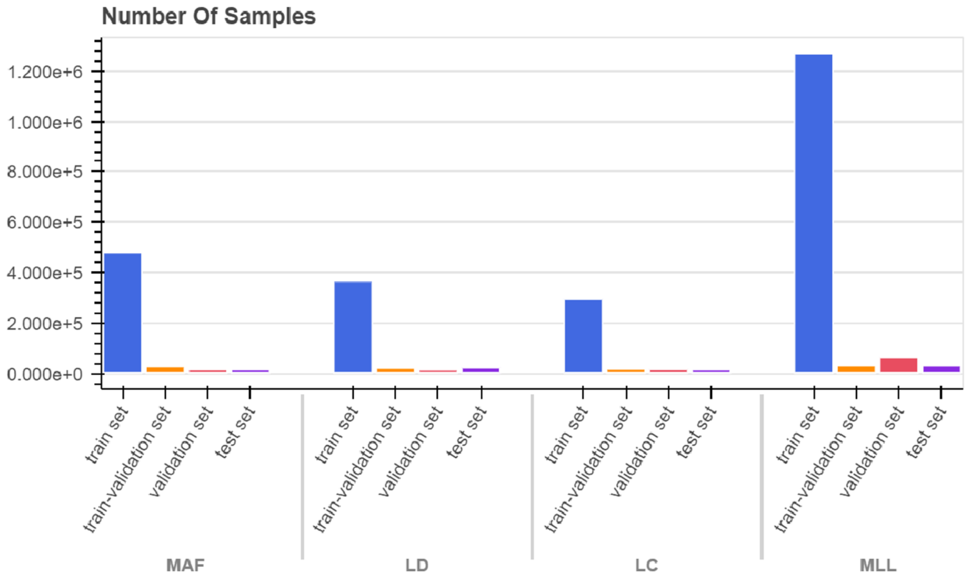
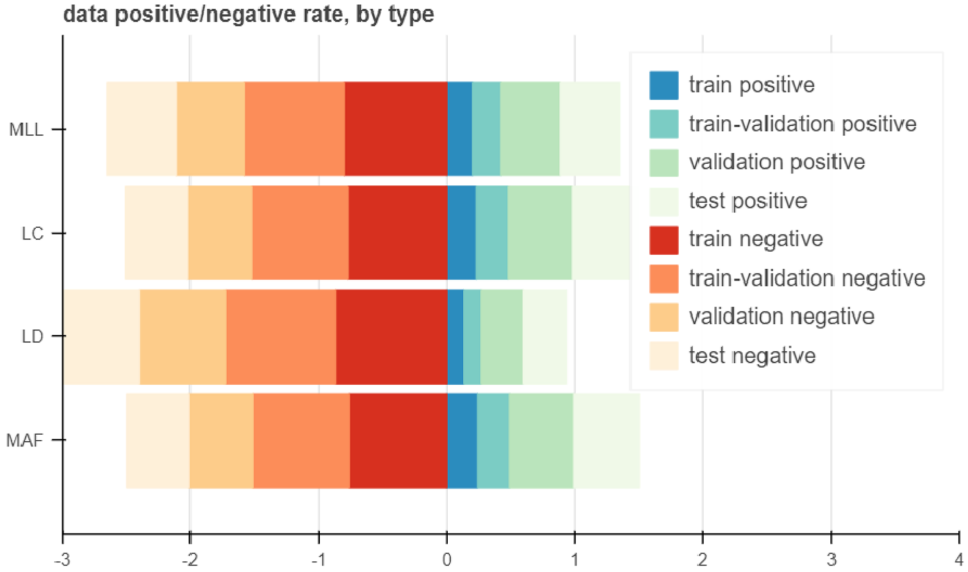


Fig. 8 Bidirectional bar chart: the positive and negative distribution of datasets on MAF, LD, LC and MLL. Take MAF as an example, the positive rate of train, train-validation, validation and test set are 0.24, 0.25, 0.5, 0.51



4.2 Experiment settings and evaluation metric

The experiments are carried out based on NVIDIA Quadro M6000 with 12 GB GPU, and tensorflow is used to implement RoBERTa of the base version. For the input embeddings, the batch size B is set to 16 for the training set after rounds of tuned on the validation set, and the sequence length F is 256, the vocabulary size V is 30.

For the encoder, the number of the Trm layers is set to 12, the number of attention heads N is 12 and the hidden size of the Trm layers H is set to 768, the intermediate size H'' is set to 3072. For the recurrent convolutional unit, the hidden size of the LSTM is 200, and the hidden size of the fully connected layer is 768, which is the same as H. The

Table 1 Parameters setting

Parameters	Tuned range	Optimal
Training batch size	[64,32,16,14]	16
Train-Validation batch size	8	8
Validation batch size	8	8
Test batch size	8	8
Max sequence length	[320,256,128]	256
Epoch	[1–5]	3
Hidden size of LSTM	[200,400]	200
Dropout rate	0.1	0.1
Warmup proportion	0.1	0.1
Learning rate	[1e−5,3e−5,5e−5]	5e-5

detailed hyperparameters of the methods are shown in Table 1.

Furthermore, the CFE extraction is taken as a sentence pair task, which is a binary classification task, so AUC (Area under curve) and accuracy are employed as evaluation metrics. The calculation of AUC takes into account the classification ability of the classifier of distinguishing both positive and negative cases, and it can still make a reasonable evaluation in the case of unbalanced samples. For example, in the anti-fraud scenario, let the fraud samples be positive examples (with only a small proportion of positive examples, assuming 0.1%), if the accuracy is used to predict, all the samples are taken as negative examples, and still the accuracy of 99.9% can be achieved. However, if using AUC, all samples are predicted to be negative, and the AUC is only 0.5, which successfully avoided the problem caused by unbalanced samples. So apart from accuracy, AUC is also taken as an evaluation metric. Since the subsequent analysis involved comparison with multi-label classification, Macro-F1 and Micro-F1 are also used as evaluation indexes.

4.3 Evaluation of retrain RoBERTa (effect of unsupervised civil case sentences knowledge in the fact elements knowledge-guided module)

The unsupervised civil case sentence knowledge aims at assisting and constraining the original Chinese RoBERTa language model, and generate the language model of Chinese civil knowledge through the Masked Language Model (MLM) task. Been exposed to a small amount of data in the domain area, allows the model to further improve the downstream sentence pair classification task.

There are two observations in this part, one is the assessment of retraining tasks, which is to investigate the task of MLM task, and the other is to investigate the effect of the unsupervised civil case sentences knowledge on the subsequent classification results. The loss and accuracy of MLM for four types of unsupervised civil case sentences knowledge data MAF, LD, LC and MLL are listed in Fig. 9, from which we can see that individual data MAF, LD and LC achieve higher classification accuracy (0.969, 0.980, 0.997) and lower loss (0.104, 0.070, 0.013) than the MLL data (0.963 and 0.148), better results come from a single type of document. It may due to the reason that the model gets confused with the mixed data when predicting the masked tokens. When the model is working on predicting the masked tokens, it is better to feed the model with the same type of data such as only MAF rather than MLL data. Different types of data are less correlated, and the model needs to pay more attention to different data, the result is the prediction loss increases and the accuracy decreases.

The effect of unsupervised civil case sentences knowledge on the proposed KGCEE is tested on eight scenarios: including retraining the RoBERTa with MAF and do the MAF sentence pair classification task; retraining with LD and conduct the LD classification task; the LC retraining task and LC sentence pair classification; retraining with MLL and MLL classification task; without MAF and fine-tuning the sentence pair task with MAF directly; without LD and conduct the sentence pair task with LD directly; without LC and do the LC fine-tuning task directly; without MLL and do the MLL fine-tuning task directly.

The classification accuracy and AUC are shown in Fig. 10, from which we can find that the slope of the line is greater than 0 on different types of data, for example, when the model is retrained with LD data, it achieves a 1.08% gain (from 95.21% to 96.29%) on classification accuracy than the model without retraining. The same is true in other cases. It indicates that retraining the model improves the downstream classification task. The main reason is that retraining the model enables the parameters to have good adaptability to the data of the special field, which can be used to initialize the downstream classification task, and better classification results can be achieved.

4.4 Evaluation of POS

In the sentiment classification task, researchers inject POS information into word vectors to improve the classification accuracy. Rezaeinia et al. (2019) develop an Improved Word Vectors (IWV), which increases the accuracy of pre-trained word embeddings in sentiment analysis. The method is mainly based on part-of-speech tagging techniques, lexicon-based approaches, word position algorithm and Word2Vec or GloVe methods. Their experiment results show that the Improved Word Vectors with POS information are effective for their sentiment classification task.

In this section, we study how the POS information affects the performance of KGCEE. As mentioned earlier, there is one most important reason why POS is added to KGCEE: POS can enhance better text representations through word sense disambiguation.

Figure 11 compares the classification accuracy and AUC performance with or without POS information on different occasions by the proposed method. As can be observed from Fig. 11, the classification accuracy increases slightly with the intervention of the POS information. When the model is trained by the proposed method, and extra POS information is injected into the model, the slope of the straight line is slightly gentle and greater than 0. For instance, the AUC index increases about 0.6% with POS

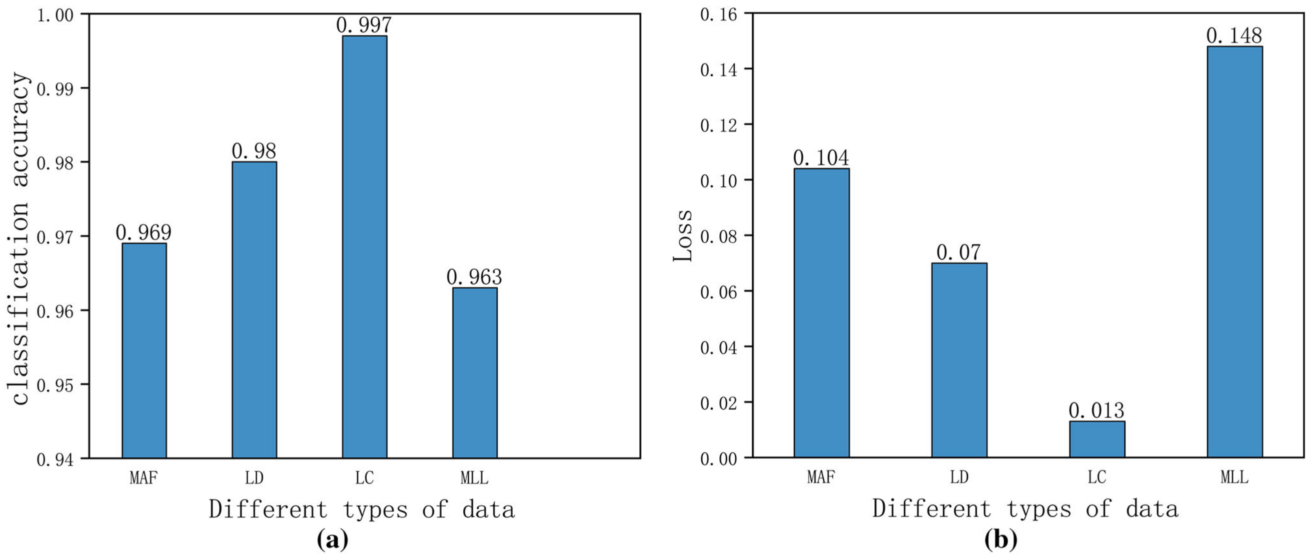


Fig. 9 **a** Accuracy of Masked Language model prediction different data; **b** Loss of Masked Language model prediction task on different data

Fig. 10 **a** Slope chart: comparing Classification Accuracy Per Data without pre-train versus with pre-train; **b** Slope chart: comparing AUC Per Data without pre-train versus with pre-train

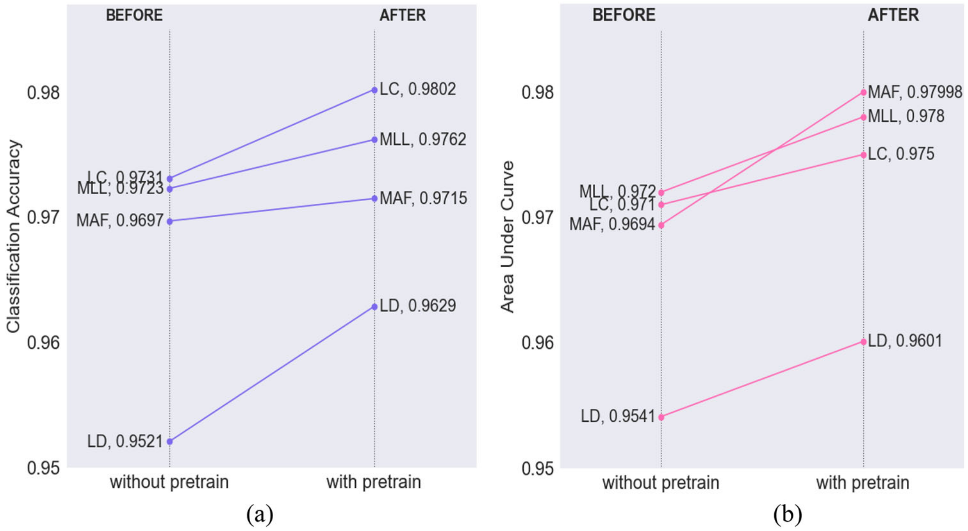
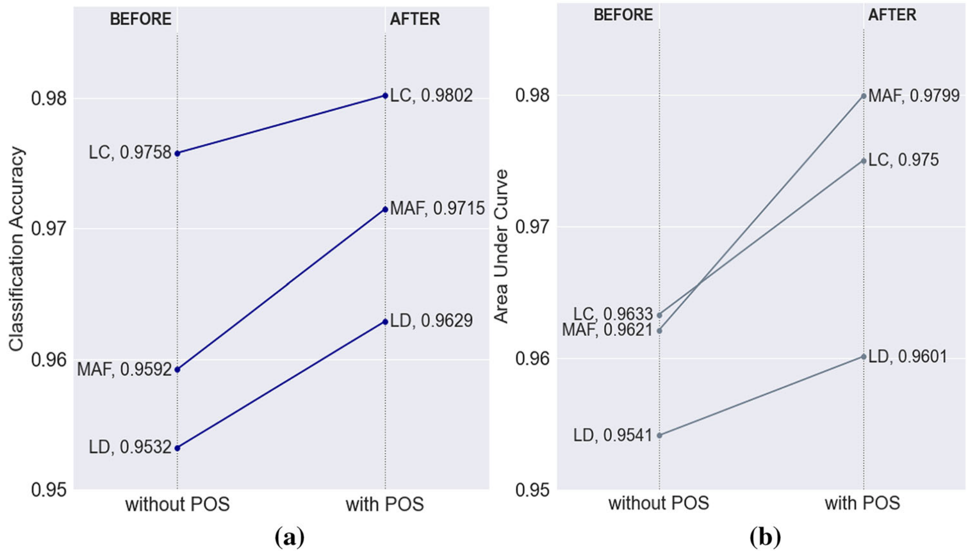


Fig. 11 **a** Slope chart: comparing Classification Accuracy Per Data without POS versus with POS. **b** Slope chart: comparing AUC Per Data without POS versus with POS



information with LD data (from 95.41% to 96.01%, shown in Fig. 11). Figure 11 points to one fact: the POS information indeed increases the classification accuracy and AUC regardless of the data type. One potential reason is that with POS information, the model conveys more semantic and syntactic information about words and can initialize the word embeddings better, which in turn leads to an elevation in the final text classification task. The injecting of POS information strength the ability to capture the semantic and syntactic explicitly.

4.5 Evaluation of layer normalization and residual component

As mentioned above, batch normalization or layer normalization can be used to improve the training speed of the network, accelerate the convergence process and increase the classification results. Layer normalization is more practical in terms of small batch size under limited GPU resources. After many trials, the optimal batch size for our GPU with 12 GB RAM is 16 with a max sequence length of 256. Residual is employed to retain as much information as possible from the original input in case of forgetting as the network depth increases.

In this experiment, two components are investigated. One is how batch size influences the classification results through different normalization methods. The other is to explore the effect of residual components on the model.

We test the proposed model with varying batch sizes containing 12,16,32, respectively, the results are shown in Tables 2, 3. From which we can find two observations. One is that the optimal batch size is 16 by layer normalization regardless of the data type. For instance, Table 3 shows that the accuracy and AUC indexes reach 97.15% and 97.99% of MAF data with layer normalization. Another observation is that when the batch size is larger, the difference between classification accuracy or AUC is tiny under those two normalization methods, but when the batch size is small to 12 or 16, the accuracy and AUC of batch normalization method are decreased, which indicates that the method is not suitable for the case of small batch size. The possible reason is that for batch normalization when the batch size is small, the calculated mean and variance values are not sufficient to represent the entire data distribution.

Table 2 Accuracy and AUC varying with different batch by batch normalization method

	MAF	LC	LD
Bath size	Accuracy/AUC	Accuracy/AUC	Accuracy/AUC
12	0.9447/0.9459	0.9485/0.9496	0.9455/0.9461
16	0.9501/0.9499	0.9489/0.9522	0.9453/0.9475
32	0.9593/0.9674	0.9697/0.9589	0.9596/0.9599

Table 3 Accuracy and AUC varying with different batch by layer normalization method

Bath size	MAF Accuracy/AUC	LC Accuracy/AUC	LD Accuracy/AUC
12	0.9622/0.9698	0.9702/0.9694	0.9551/0.9512
16	0.9715/0.9799	0.9802/0.9750	0.9629/0.9601
32	0.9694/0.9701	0.9772/0.9702	0.9624/0.9602

Figure 12 shows the influence of residual components on the classification accuracy and AUC with the different training datasets. From which we can find that those straight lines have a higher slope. The higher the slope, the greater the role of the residual component. The model with residual component achieves 20% improvements on average on classification accuracy or AUC, which is independent of data type. For instance, the classification accuracy gains 19.49% improvements (from 77.66% to 97.15%) on the MAF data set with the residual component. This is because the deep network without residual connection cannot be well optimized, resulting in increased losses or decreased classification accuracy.

4.6 Comparison with other methods

In this experiment, to further validate the effectiveness of our model, the proposed KGCEE is compared with two kinds of text representation-based models: word2vec-based models and transformer-based models. All those methods are performed on the same civil text dataset with our KGCEE.

(1) Word2vec-based models:

Text-CNN: this model is to get the word embeddings through the word2vec model and employs CNN (Kim 2014) to encode word embeddings to obtain the fact descriptions.

BiLSTM: two consecutive BiLSTM (Greff et al. 2017) layers are utilized to encode fact descriptions.

TopJUDGE: (Zhong et al. 2018) proposed a model that used CNN with max-pooling and combined with two layers of BiLSTM for legal charge prediction task on criminal law cases. We reproduced this model by replacing the criminal cases with the same civil text data set.

CBLLC: a multitask convolutional neural network (CNN) combined with bidirectional long short-time memory (BiLSTM) leveraging label co-occurrence framework, a charge prediction model proposed by (Dong et al. 2020). This model is to predict multiple labels for criminal cases

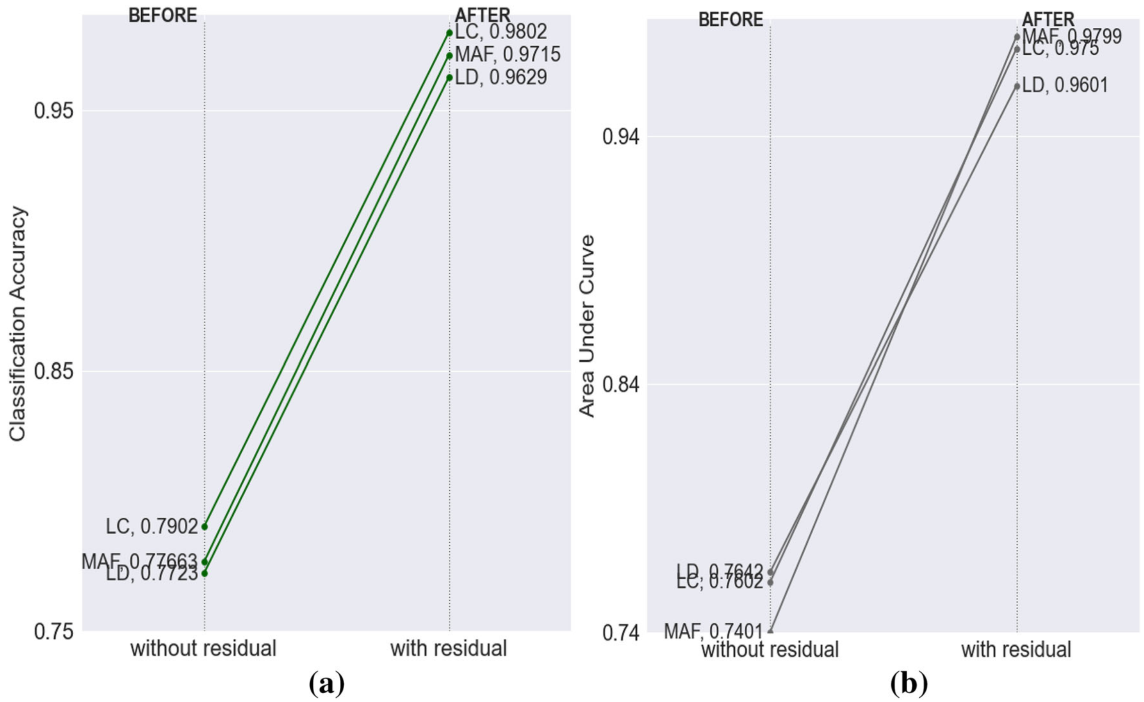


Fig. 12 **a** Slope chart: comparing Classification Accuracy Per Data without residual versus with residual; Slope chart: comparing AUC Per Data without residual versus with residual

leveraging label co-occurrence. The setting of the label co-occurrence part is removed for a fair comparison.

(2) Transformer-based models

BERT-FC: (Devlin et al. 2019) proposed a new language representation model for the downstream task. a pre-trained BERT model that combined with a fully connected layer and a softmax function for the binary classification task. For a fair comparison, hyperparameters such as numbers of attention heads, and number of encoding layers are kept the same when the network architecture is reproduced.

BERT-RCNN: BERT-RCNN replaced the FC layer with an RCNN (recurrent convolutional gated unit (Lai et al. 2015) layer.

RoBERTa-FC: an optimized RoBERT (Liu et al. 2019a, b) model pre-trained with large Chinese corpus, and fine-tuned with civil texts data to conduct classification task. Apart from the RCNN unit, all the parameters remain the same as that of KGCEE.

The experiments are carried out on a test set, and the results are shown in Table 4. Table 4 shows the classification results using three data sets by different methods. There are four observations from the results.

First, the proposed method is superior to all the other baseline methods on almost all kinds of data set. For instance, in Table 4, the KGCEE method achieves 1% improvements on average over the best BERT-FC method and gets an accuracy of 97.15% and 97.99% of AUC on the MAF data set, which demonstrates the effectiveness of our model. The improvements can be attributed to the superiority of transformer-based text representations and the fine-tuned strategy KGCEE adopts.

Second, transformed-based methods outperform all the word2vec-based methods. For example, the worst-performing transformer-based model BERT-FC achieves 94.66% accuracy and 94.95% AUC, which outperforms the best performing word2vec-based method 89.94% accuracy and 90.03% AUC. It is mainly because transformer-based methods own a strong ability to catch semantic information about words and takes the context information into account, as a result, the model can understand the sentences or words better and the downstream application performance in the NLP field is effectively improved.

Third, comparisons among word2vec-based methods.

In our earlier work, we propose a new framework called CBLLC to predict charges for criminal cases, which is a multi-label classification task. The CBLLC is mainly based on the cascade of CNN and BiLSTM to obtain the feature representation of text. The experiment shows that the CBLCC utilizes CNN and BiLSTM to encode the input text, which captures short features such as binary information and long-term dependencies. A combination of these two approaches makes it performs better than the Text-CNN method or RNN method on multi-label charge prediction task.

Table 4 Performance comparison with classification accuracy and AUC value on the test set

Methods	Data	MAF		LD		LC	
	Metrics	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC
Text-CNN		0.8933	0.9021	0.8874	0.8795	0.8798	0.9002
BiLSTM		0.8744	0.8857	0.8525	0.8421	0.8751	0.8521
ToPJUDGE		0.8961	0.8941	0.8975	0.8998	0.9005	0.9047
CBLLC		0.8994	0.9003	0.9047	0.8991	0.9001	0.9098
BERT-FC		0.9466	0.9495	0.9372	0.9388	0.9458	0.9546
BERT-RCNN		0.9522	0.9558	0.9432	0.9447	0.9557	0.9601
RoBERTa-FC		0.9655	0.9710	0.9536	0.9542	0.9688	0.9693
KGCEE		0.9715	0.9799	0.9629	0.9601	0.9802	0.9750

In the sentence pair task, there are similar results. A method such as Text-CNN has a lower accuracy of 89.33% on the MAF classification task. The main reason is that Text-CNN extracts features by filter window size, which is limited in long-distance modeling and insensitive to word order. When Text-CNN is used to encode a long sequence text, it captures short features restricted by window size, and the word order of the text is not taken into account, which cannot represent the text better. Similarly, BiLSTM does not perform very well, because it captures long-term dependencies and has its limitations: the network does not simultaneously compute losses or encode the text, which is essentially the concatenation of forward encoding and backward encoding. ToPJUDGE model or CBLLC gains 0.32% and 0.61% improvements on accuracy than TextCNN. It demonstrates the effectiveness of combining CNN and BiLSTM.

Last, comparisons between transformer-based methods.

In the general domain, RoBERTa has achieved competitive results compared with other methods on many tasks. For example, in the Glue test, RoBERTa is 3.4% higher than the BERT method on the MNLI task. In our sentence pair classification task, the advantage of RoBERTa has endured.

In our sentence pair task, compare to BERT-FC, the RoBERTa-FC achieves a 1.89% gain on the accuracy, the improvements can be attributed to the optimized training strategy of RoBERTa, such as the dynamic masking mechanism and more training data to pre-train model. Compare to the static masking from BERT, the dynamic mask is to ensure that for the same training samples to apply different mask strategies in different epochs, tokens change in each sample of different epochs, the accuracy rise by 1.89% illustrate that dynamic masking can improve the classification performance. KGCEE achieves its best accuracy and AUC when substitute RCNN for a fully connected layer, the reason behind this is RCNN encodes the text further and extracts core information which benefits the downstream task.

4.7 Evaluation of the generalization ability of proposed KGCEE

The proposed KGCEE is conducted by transforming the multi-label classification task into a sentence pairs task. That is to say, the present training data from CAIL are shaped into corresponding sentence pairs. And then construct a large number of sentence pairs by leveraging the label information, which serves as enhanced training data.

This section aims at investigating the generalization ability of the proposed KGCEE.

The multi-label classification task is conducted under a smaller dataset of CAIL, and the sentence pairs task of proposed KGCEE is carried under a larger dataset with enhanced sentence pairs of extra knowledge (which is the labeled data knowledge shown in Fig. 6). To validate the generalization ability of the proposed method, the sentence pairs method in this paper is compared with the method of multi-label classification. The results are shown in Table 5, from which we can see that for word2vec embeddings methods such as TextCNN (Kim 2014), CBLLC (Dong et al. 2020) and transformer-based method KGCEE when the task is transformed into a sentence pair task, the extraction results are both elevated a lot compared to complete multi-label classification task directly. For instance, KGCEE with sentence pair task and enhanced sentence pairs knowledge achieves 93.1% on Macro-F1, which gains 7.4% improvements than conducting multi-label task directly on MAF data. The same is true in other cases. The reason is that the former leverage label information to generate more training data and thus the generalization ability is enhanced.

To further illustrate the effectiveness of the proposed method and find out whether the result is caused by the algorithm or data itself, the experiment is conducted under MLL data with four sets (training set, train-validation set, validation set and test set) and compared with one state-of-art method RoBERTa-FC. The error rate of the four data

Table 5 Comparisons between multi-label classification task and sentence pair task

Methods	Data	MAF		LD		LC	
	Metrics	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
Text-CNN(sentence pairs)		83.3	84.5	84.7	83.6	83.1	84.2
TextCNN(multi-label classification)		80.1	81.2	81.3	80.4	79.8	78.9
CBLLC(sentence pairs)		85.1	87.3	85.3	87.7	84.2	84.6
CBLLC(multi-label classification)		81.4	82.5	80.6	82.4	79.9	81.6
KGCEE(sentence pairs)		93.1	92.8	88.2	90.5	89.7	90.3
KGCEE(multi-label classification)		85.7	87.4	83.4	85.2	83.3	85.5

Table 6 Comparisons between RoBERTa and KGCEE and algorithm and data analysis

Data splits	Positive rate	Error rate of KGCEE (%)	Error rate of RoBERTa-FC
Human error	–	2.0	2.0
Training set	0.2	4.88	6.22
Train-validation set	0.22	8.37	10.34
Validation set	0.47	10.81	12.74
Test set	0.46	11.72	14.23

split sets is evaluated in Table 6 and four conclusions can be drawn.

First, several experienced Phds in law are asked to extract civil elements, and the human error rate is 2.0% on average. From Table 6, the avoid bias (the difference value between training set error and human error) is 2.88% for KGCEE and 4.22% for RoBERTa-FC, which illustrates that the former has a better fitting ability to the training set and closer to human performance.

Second, the gaps between training error and train-validation error are assessed, from Table 6 we can see that our method obtains the smaller gap 3.49%, the smaller the gap, the better the algorithm. The reason is that the gap is caused by the algorithm instead of the data mismatch problem (the gap is caused by the data itself) since the training set and train-validation set share the same positive and negative distribution. Our method achieves the smallest training error and train-validation error, though the method has not "seen" the train-validation before (there is no backpropagation on the train-validation dataset), it can be generalized to the train-validation dataset, indicating that the algorithm error (variance) is small.

Third, for different methods, the consistent small gaps (2.44% for KGCEE and 2.40% RoBERTa-FC) between the train-validation and the validation set indicate that there exists no data mismatch problem and the algorithm is good at processing data with different positive and negative distribution since the train-validation and the validation share different positive and negative distribution.

Last, the error between validation and test set is evaluated, again, our method obtains the smallest gap of 0.91% illustrates that there is no overfitting of the validation set, and the proposed KGCEE generalizes the good results from the validation set to the test set and shows strong generalization ability.

4.8 Error analysis

Through the analysis of the model results, we find that the model performs the worst on LD data, as shown in Fig. 10 and Fig. 11. As can be seen from Fig. 10, the accuracy and AUC of this method in LD data are the lowest, which are 96.29% and 96.01%, respectively, regardless of whether the method had been retrained or not. Similar results are also shown in Fig. 11, no matter whether POS information is injected, the accuracy and AUC of LD dataset are also the lowest, which are 1%-2% lower than those of other datasets. After analysis, we find that the LD dataset has the lowest positive rate of 0.13 in the training set, which is lower than 0.24 and 0.23 in MAF and LC. That is to say, the number of positive samples in the data set is insufficient. Therefore, the model may not be able to distinguish between positive and negative samples in LD data correctly, the final result is the lowest index in LD dataset.

On the other hand, we analyze some wrong cases and several examples are shown in Table 7. In these examples, the ground labels are all positive (1), but our model predicts negative (0) as the result. We summarize the following characteristics of the wrong cases: for the label *Had*

Table 7 Error analysis

Data type	Instance	Gold label	Predicted label
MAF	< Had children out of wedlock, the house is now occupied by the defendant and his daughter >	1	0
LD	< pay compensation of double wages for not signing a labor contract, it is requested that the defendant be ordered to pay 4,500 yuan in arrears and 3,447 yuan in overtime, 7,500 yuan in double wages without signing a labor contract, and 1,250 yuan in compensation for terminating the contract >	1	0
LC	< a written commitment to repay the debt, XX Coal Mine of XX City has stamped on the column of a borrower of "Notice of Transfer of Creditor's Rights and Collection" and is willing to continue to fulfill the repayment obligations under the original loan contract >	1	0

children out of wedlock in MAF dataset, for label *pay compensation of double wages for not signing a labor contract* in LD dataset, for the label *a written commitment to repay the debt* in LC dataset, their corresponding training samples are the least, which is 8722, 7882 and 5185. The most important thing is that their positive and negative ratios differ too much and their positive samples are too few, only 95, 165 and 33, respectively. Therefore, the model tends to predict the positive sample to the negative sample, which leads to the poor effect of the model for this kind of data.

5 Conclusion

In this paper, we propose a knowledge-guided extraction model for civil elements in marriage and family, labor disputes and loan contracts. Compared to other state-of-art models, our KGCEE model applies unsupervised sentences as guidance knowledge. Such an approach enables the fact elements knowledge-guided module generates the constrained model that more sensitive to civil text-domain data, which is further to transfer the sensitive knowledge and initialize the parameters in the downstream task and elevate the extraction performance as a result.

The supervised sentence pairs constructed in the fact elements knowledge-guided module generates robust word embeddings in the extraction process, which carries significant rich label information features and ensures a high-quality generalization ability.

Also, we employ the fine-tuned RoBERTa model to initialize the word embedding and inject POS information into the KGCEE model. Such word vectors include syntactic information about words and rich context semantic information. Finally, the proposed KGCEE outperforms other baseline models on different data sets, and the elements extracted can provide case summary for judges and assist them in making efficient decisions on civil disputes.

As future work, it would be intriguing to extend the model to other civil areas such as house property or copyright disputes. The large version of RoBERTa with more layers and parameters achieves better performance on different kinds of NLP tasks. The proposed KGCEE is mainly based on the base version of RoBERTa due to hardware constraints. On this condition, we will explore the use of a large version in element extraction by using a method such as training the layers that have an impact on extraction results and freezing other layers to obtain better results than the base version. Besides, due to the co-occurrence or exclusion relationship between different element tags, it is also a future research direction to generate more data and more difficult tasks through tag relationship.

Acknowledgements The authors are grateful to the editors and the anonymous reviewers for their insightful comments and suggestions, which have improved the quality of the paper immensely. This work is supported by National Key R & D Program of China, under Grant Nos.2018YFC0830800.

Data availability All data included in this study are available upon request.

Declarations

Conflict of interest All authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Alkhodair S-A, Ding S-H, Fung B, Liu J (2020) Detecting breaking news rumors of emerging topics in social media. Inf Process Manag. https://doi.org/10.1016/j.ipm.2019.02.016

Bartolini R, Lenci A, Montemagni S, Pirrelli V, Soria C (2004) Semantic mark-up of Italian legal texts through NLP-based techniques. In: Proceedings of the 4th international conference on language resources and evaluation, pp 795–798

Burdisso S-G, Errecalde M, Montes-Y-Gomez M (2019) A text classification framework for simple and effective early depression detection over social media streams. Expert Syst Appl 133:182–197

Chen H, Luo X (2019) An automatic literature knowledge graph and reasoning network modeling framework based on ontology and natural language processing. Adv Eng Inform 42:100959. https://doi.org/10.1016/j.aei.2019.100959

Chen L, Lee C, Chen M (2020a) Exploration of social media for sentiment analysis using deep learning. Soft Comput 24(11):8187–8197. https://doi.org/10.1007/s00500-019-04402-8

Chen F, Yuan Z, Huang Y (2020b) Multi-source data fusion for aspect-level sentiment classification. Knowl-Based Syst. https://doi.org/10.1016/j.knosys.2019.07.002

Devlin J, Chang M-W, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers). Minneapolis, Minnesota: Association for Computational Linguistics, pp 4171–4186. https://doi.org/10.18653/v1/N19-1423

Dong H, Yang F, Wang X (2020) Multi-label charge predictions leveraging label co-occurrence in imbalanced data scenario. Soft Comput. https://doi.org/10.1007/s00500-020-05029-w

Du Y, Pei B, Zhao X, Ji J (2020) Deep scaled dot-product attention based domain adaptation model for biomedical question answering. Methods 173:69–74. https://doi.org/10.1016/j.ymeth.2019.06.024

Ekinci E, Omurca S-I (2020) Concept-LDA: incorporating Babelfy into LDA for aspect extraction. J Inf Sci 46(3):406–418. https://doi.org/10.1177/0165551519845854

Elnagar A, Al-Debsi R, Einea O (2020) Arabic text classification using deep learning models. Inf Process Manag. https://doi.org/10.1016/j.ipm.2019.102121

Fan Z, Li G, Liu Y (2020) Processes and methods of information fusion for ranking products based on online reviews: an overview. Inf Fusion 60:87–97. https://doi.org/10.1016/j.inffus.2020.02.007

Fang W, Luo H, Xu S, Love P, Lu Z, Ye C (2020) Automated text classification of near-misses from safety reports: an improved deep learning approach. Adv Eng Inf. https://doi.org/10.1016/j.aei.2020.101060

Gargiulo F, Silvestri S, Ciampi M, De Pietro G (2019) Deep neural network for hierarchical extreme multi-label text classification. Appl Soft Comput 79:125–138. https://doi.org/10.1016/j.asoc.2019.03.041

Gonzalez JA, Hurtado LF, Pla F (2020) Transformer based contextualization of pre-trained word embeddings for irony detection in Twitter. Inf Process Manag. https://doi.org/10.1016/j.ipm.2020.102262

Greff K, Srivastava K-J, Steunebrink B, Schmidhuber J (2017) LSTM: a search space odyssey. IEEE Trans Neural Netw Learn Syst 28(10):2222–2232. https://doi.org/10.1109/TNNLS.2016.2582924

Guo B, Zhang C, Liu J, Ma X (2019) Improving text classification with weighted word embeddings via a multi-channel TextCNN model. Neurocomputing 363:366–374. https://doi.org/10.1016/j.neucom.2019.07.052

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, NV, USA. IEEE Computer Society, pp 770–778

He J, Zhao L, Yang H, Zhang M, Li W (2020) HSI-BERT: hyperspectral image classification using the bidirectional encoder representation from transformers. IEEE Trans Geosci Remote Sens 58(1):165–178. https://doi.org/10.1109/TGRS.2019.2934760

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Howard J, Ruder S (2018) Universal language model fine-tuning for text classification. In: Proceedings of the 56th annual meeting of the association for computational linguistics, vol 1, Long Papers. Melbourne, Australia: Association for Computational Linguistics, pp 328–339. https://doi.org/10.18653/v1/P18-1031

Hu Z, Li X, Tu C, Liu Z, Sun M (2018) Few-shot charge prediction with discriminative legal attributes. In: Proceedings of the 27th international conference on computational linguistics. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp 487–498. https://www.aclweb.org/anthology/C18-1041

Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd international conference on machine learning, Lille, France, PMLR 37, pp 448–456

Kao A, Poteet S (2007) Natural language processing and text mining. ACM Sigkdd Explor Newslett 7(1):115

Kim Y (2014) Convolutional neural networks for sentence classification. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, pp 1746–1751. https://doi.org/10.3115/v1/D14-1181

Kim S, Park H, Lee J (2020) Word2vec-based latent semantic analysis (W2V-LSA) for topic modeling: a study on blockchain technology trend analysis. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2020.113401

Lai S, Xu L, Liu K et al (2015) Recurrent convolutional neural networks for text classification. Proceedings of the twenty-ninth AAAI conference on artificial intelligence, pp 2267–2273

Li J, Zhang G, Yan H, Yu L, Meng T (2018) A Markov logic networks based method to predict judicial decisions of divorce cases. In: 2018 IEEE international conference on smart cloud (SmartCloud), New York, NY, pp 129–132. https://doi.org/10.1109/SmartCloud.2018.00029

Li J, Zhang G, Yu L, Meng T (2019a) Research and design on cognitive computing framework for predicting judicial decisions. J Sig Process Syst 91(10):1159–1167

Li C, Sheng Y, Ge J, Luo B (2019) Apply event extraction techniques to the judicial field. In: The 2019 ACM international joint conference on pervasive and ubiquitous computing and the 2019 ACM international symposium, pp 492–497

Li X, Zhang H, Zhou X (2020) Chinese clinical named entity recognition with variant neural structures based on bert methods. J Biomed Inf. https://doi.org/10.1016/j.jbi.2020.103422

Lin W, Kuo T, Chang T, Yen C, Chen C, Lin C (2012) Exploiting machine learning models for Chinese legal documents labeling, case classification, and sentencing prediction. In: Proceedings of the 24th conference on computational linguistics and speech processing (ROCLING 2012). Chung-Li, Taiwan: The Association for Computational Linguistics and Chinese Language Processing (ACLCLP), pp 140–141. https://www.aclweb.org/anthology/O12-1013

Liu G, Guo J (2019) Bidirectional LSTM with attention mechanism and convolutional layer for text classification. Neurocomputing 337:325–338. https://doi.org/10.1016/j.neucom.2019.01.078

Liu Y, Chen Y, Ho W (2015) Predicting associated statutes for legal problems. Inf Process Manag 51(1):194–211. https://doi.org/10.1016/j.ipm.2014.07.003

Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Stoyanov V (2019) Roberta: a robustly optimized BERT pretraining approach. CoRR abs/1907.11692

Liu Y, Jin X, Shen H (2019b) Towards early identification of online rumors based on long short-term memory networks. Inf Process

Manag 56(4):1457–1467. https://doi.org/10.1016/j.ipm.2018.11.003

Luo B, Feng Y, Xu J, Zhang X, Zhao D (2017) Learning to predict charges for criminal cases with legal basis. In: Proceedings of the 2017 conference on empirical methods in natural language processing. Copenhagen, Denmark: Association for Computational Linguistics, pp 2727–2736. https://doi.org/10.18653/v1/d17-1289

Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. CoRR, abs/1301.3781

Mikolov T, Sutskever I, Chen K, Corrado G, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th international conference on neural information processing systems, Curran Associates Inc, pp 3111–3119

Moradi M, Dorffner G, Samwald M (2020) Deep contextualized embeddings for quantifying the informative content in biomedical text summarization. Comput Methods Programs Biomed 184:105117. https://doi.org/10.1016/j.cmpb.2019.105117

Peters M, Neumann M, Iyyer M, Gardner M, Zettlemoyer L (2018) Deep contextualized word representations. In: Conference of the North American chapter of the association for computational linguistics: human language technologies, vol 1 (Long Papers). New Orleans, Louisiana: Association for Computational Linguistics, pp 2227–2237. https://doi.org/10.18653/v1/N18-1202

Rezaeinia SM, Rahmani R, Ghodsi A, Veisi H (2019) Sentiment analysis based on improved pre-trained word embeddings. Expert Syst Appl 117:139–147. https://doi.org/10.1016/j.eswa.2018.08.044

Schilder F, Graham K, James P (2005) Event extraction and temporal reasoning in legal documents. In: Proceedings of the 2005 international conference on Annotating, extracting and reasoning about time and events, pp 59–71

Sinoara R-A, Camacho-Collados J, Rossi R-G, Navigli R, Rezende S-O (2019) Knowledge-enhanced document embeddings for text classification. Knowl-Based Syst 163:955–971. https://doi.org/10.1016/j.knosys.2018.10.026

Sun C, Yang Z, Wang L, Zhang Y, Wang J (2020) Attention guided capsule networks for chemical-protein interaction extraction. J Biomed Inf. https://doi.org/10.1016/j.jbi.2020.103392

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: Proceedings of the 31st international conference on neural information processing systems. ACM, pp 6000–6010

Xia H, Yang Y, Pan X, Zhang Z, An W (2020) Sentiment analysis for online reviews using conditional random fields and support vector machines. Electron Commer Res 20(2):343–360. https://doi.org/10.1007/s10660-019-09354-7

Yan Y, Zheng D, Lu Z, Song S (2017) Event identification as a decision process with non-linear representation of text. arXiv:1710.00969

Zablith F, Osman I-H (2019) ReviewModus: text classification and sentiment prediction of unstructured reviews using a hybrid combination of machine learning and evaluation models. Appl Math Model 71:569–583. https://doi.org/10.1016/j.apm.2019.02.032

Zhang F, Fleyeh H, Wang X, Lu M (2019a) Construction site accident analysis using text mining and natural language processing techniques. Autom Constr 99:238–248. https://doi.org/10.1016/j.autcon.2018.12.016

Zhang X, Zhang Y, Zhang Q, Ren Y, Qiu T, Ma J, Sun Q (2019b) Extracting comprehensive clinical information for breast cancer using deep learning methods. Int J Med Inf. https://doi.org/10.1016/j.ijmedinf.2019.103985

Zhao F, Li P, Li Y, Hou J, Li Y (2019) Semi-supervised convolutional neural network for law advice online. Appl Sci Basel 9(17):3617. https://doi.org/10.3390/app9173617

Zhong H, Guo Z, Tu C, Xiao C, Liu Z, Sun M (2018) Legal judgment prediction via topological learning. In: Proceedings of the 2018 conference on empirical methods in natural language processing, Brussels, Belgium: Association for Computational Linguistics, pp 3540–3549. https://doi.org/10.18653/v1/D18-1390