IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received September 4, 2019, accepted September 21, 2019, date of publication September 25, 2019, date of current version October 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2943668

Combining Domain Knowledge Extraction With Graph Long Short-Term Memory for Learning Classification of Chinese Legal Documents

GUODONG LI, ZHE WANG, AND YINGLONG MA, (Member, IEEE)
School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China
Corresponding author: Guodong Li (lgd@ncepu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC0831404.

ABSTRACT It is of great importance for procedure retrieval to find an effective classification method of Chinese legal documents with deep semantic understanding, as the electronic documents of Chinese law have massive volume and complex structure. In this paper, a method for learning Chinese legal document classification using Graph LSTM (Long Short-Term Memory) combined with domain knowledge extraction is proposed. First, the judicial domain model is constructed based on ontologies that include top-level ontology and domain-specific ontology. Second, the legal documents are divided into different knowledge blocks through top-level ontology and domain-specific ontology. Third, information is extracted from the knowledge blocks according to the legal domain model and stored in XML files. At last, Graph LSTM is applied for classification. The experiments show that compared with the traditional classification methods of support vector machine (SVM) and LSTM, Graph LSTM has higher classification accuracy and better classification performance.

INDEX TERMS Graph LSTM, judicial field, judicial domain model, literature search.

I. INTRODUCTION

As an active and important topic in the field of information retrieval, the automatic document retrieval system is essentially a core component of modern decision support systems to reduce information overload and improve the performance of document search systems [1]. Information retrieval improves the efficiency of large data retrieval [19]. However, how to search and obtain documents that users really need or are closely related to the documents that users really want has become a key issue for effective and efficient document search [2]. This aforementioned situation is especially true for legal document searches [10], [11]. Massive judicial documents containing rich and valuable information can be mined to provide prosecutors and judges with intelligent assistant case handling services, such as case-based reasoning [3], [4] and legal citations [12], and so on. So it is necessary to find an effective and sustainable [45] method for the classification of legal documents as a large number of similar documents are difficult to obtain by traditional manual methods.

The associate editor coordinating the review of this manuscript and approving it for publication was Shahab Shamshirband.

There are few existing extraction methods applicable to the classification of Chinese legal documents except for ontology technology. Ontology is one of the few methods that are practicable in the classification of legal documents. It helps to filter out redundant or inconsistent data, generate semantically rich results, and improve the effectiveness and precision of judging document classification. ontology as an effective form of knowledge representation about the real world or one of its components [31] can clearly express concepts and relationships between concepts. Ontology-based knowledge reasoning [37], [38], semantic disambiguation [39] and information retrieval [43] have achieved many achievements.

A large number of extraction methods are not suitable for the classification of Chinese legal documents. When it comes to document classification, the text feature extraction is the most important step which focuses on how to extract key features from the text that can reflect the characteristics of the text and capture the mapping between features and categories. However, some traditionally and wide-accepted extracting methods, representing text with a large number of words, lack a semantic understanding of the document, such as rule-based methods [9], and the method of bag-of-words (BOW) [5] and others. In addition, other methods with higher-dimension vectors for text representation ignore the structural information of the text and the association between words and words, so their classification results for judicial texts are not ideal. For instance, Word Embedding [34] calculates the characteristics of the text through the word vector, Word2vec [6], [7], doc2vec [8] and so on, using semantic-based representation; the Latent Dirichlet Allocation (LDA) topic model [40] extracts feature by looking for the distribution of topics for each document and the distribution of words in each topic. Last but not least, some deep learning methods, loved by scholars all over the world and widely used for feature extraction, are also not suitable for legal documents, such as Convolutional Neural Networks (CNN) Model. The model does not work well for modeling long sequence information and is inconvenient for parameter adjustment as it continuously updates the feature vector set by backward propagation.

The classification of Chinese legal documents is a tough task in itself. Firstly, the high accuracy of classification results for Chinese legal documents is difficult to reach, as the structure of Chinese judicial texts is complex, and its text classification involves interests such as criminal sentencing and so on. Secondly, the performance of traditional extraction methods of legal documents is required to improve by capturing the high-level semantics of the text with the help of the specific knowledge of the judicial field. In Chinese legal documents, different cases involve different sentencing judgments, and there are also differences in the characteristics of the plot and the rules for the preparation of legal documents.

Grounded on the shortage of suitable method and the complexity of Chinese legal documents, a Chinese legal document classification method that combines domain knowledge extraction with Graph LSTM is put forward in the paper. Specifically, the top-level ontology and domain-specific ontology is constructed by reusing and adapting the concepts and attributes of existing legal documents. The top-level ontology and the domain-specific ontology are then combined for document representation and information extraction. After that, the neural network method Graph LSTM is utilized for text classification. Compared with the traditional methods of SVM and LSTM, the classification precision is greatly improved.

II. RELATED WORK

Ontology technology has been used in many fields. For example, Huang uses artificial neural networks to align biological ontology [7]. With the maturity of ontology engineering, ontology has been widely used in the judicial field. There are typical achievements: Sundaram *et al*. [12] introduced how to build ontology in the field of Spanish law based on the ontology development method METHONTOLOGY and the ontology engineering workbench WebODE; Chen *et al*. [14] used the Dutch criminal law system as an example to develop the LRI core legal ontology retrieval system at abstract and specific levels; and Farhoodi *et al*. [15] proposed an automatic Thai law ontology framework, including succession law ontology and family law ontology, which can automatically generate seed ontology and extend ontology using ant colony algorithm. However, the legal ontology of other countries is not appropriate for China as the legal systems of different countries are different. Moreover, China also has no ontology of legal cases. Therefore, in view of authentic cases and the characteristics of the Chinese legal system, the ontology of the legal field of China is constructed in the paper.

There are some popular feature engineering models in most existing text classification methods use. Three famous classification methods are illustrated as follows. Firstly, Chen, *et al.* [14] used the LDA model to classify short texts. The model concentrates text semantically and reduces sparseness. But in the standard LDA model, it uses the bag of words method, and each feature word is considered to be equally important, which does not apply to legal texts. The reason is that different characteristic words obviously need to correspond to different weights in the legal text. For example, the number of deaths can lead to heavier penalties. Therefore, combining certain weighting methods to adjust the weights of different feature words can effectively improve accuracy. Secondly, Farhoodi, M. *et al.* employ language models classifier based on word-level n-grams for Persian text classification [15]. This method can retain the word order information of the original sentence to a certain extent. The larger the n(number of words), the more complete the order information is retained and the features will become extremely sparse. Thirdly, Ma *et al*. [21] used WMD (Word Mover Distance) to calculate the similarity of legal documents and then classified them. This scheme considers the similarity of words sentence structure. But this method is sensitive to sentence length, which means that when measuring sentences with different lengths, the similarity will be smaller than the intuitive semantic understanding. At the same time, the method is sensitive to noisy data. Most Chinese legal documents are not stored in a structured form but in a statement-oriented form of expression. In different legal texts, two completely different words may express the same meaning.

In recent years, with the development of deep learning technology, more and more Deep Learning algorithms are used for text classification. The recursive auto encoders (RAE) proposed by Socher *et al*. [16] is one of the common Deep Learning algorithms, the core idea of which is to calculate the reconstruction error of the text sentence vector and the cross-entropy error used for the text classification. Although this model can obtain high-quality text sentence vectors, it does not have enough ability to combine long sentences. In order to optimize the model, Socher proposed the Matrix-Vector RNN model [17]. This optimized model not only has the ability to combine the word vector with sentence vectors in the RAE, but also records and modifies the combination of the core words by adding the matrix, making the model have the ability to learn logical propositions and the meaning of operators in natural language. Socher *et al*. further proposed an LSTM text sentiment classification model based on the RNN model [18]. LSTM [46] solves the long-term dependency problem that RNN cannot solve through the "door mechanism for which is more in line with the requirements of text understanding. However, both the LSTM and RNN models contain only forward information. However, neither of these models can remember the backward sequence information. Therefore, Brueckner proposed a two-way LSTM model [20], which adds a reverse layer to the LSTM model, so that the LSTM considers context information at the same time, remembers the reverse sequence information, and further obtains bidirectional non-destructive text information.

To sum up, the key issue in the classification of legal documents is to find a way to effectively capture and describe the semantic information of legal documents. And legal domain knowledge can better capture the document-level semantics of legal texts. Therefore, this paper proposes to build a knowledge domain knowledge model. The model not only considers the semantic capture of legal background knowledge but also considers the special structure of legal documents, which is assisted by experts in the field of law. Moreover, the information extraction by the model not only has a low dimension but also a high accuracy.

III. OVERVIEW OF FRAMEWORK

In this section, there is a brief introduction of the approach of classification for Chinese legal documents by combining Graph LSTM with domain knowledge extraction. The work can be divided into three parts: 1) ontology-based legal domain knowledge model; 2) Information extraction based on judicial domain model; 3) Chinese legal document classification based on Graph LSTM. As shown in Figure 1.

A. ONTOLOGY-BASED JUDICIAL DOMAIN MODEL

Domain knowledge of Chinese judgment files can be modeled by ontology methods which is a clear and detailed description of the shared concept system. The model consists of two parts, a top-level ontology, and a domain-specific ontology. The former mainly describes the general attributes of legal cases, including objective aspects, subjective aspects and judgment results, etc., while the latter is built on top-level ontology and specific terms in a certain type of case, involving a detailed description of the case. And the combining of the two parts forms a complete judicial domain knowledge model for document classification. Moreover, the domain knowledge model is based on these common concepts and category terms from *China Judgments Online* [22]. The document is a highly standardized concept formulated by the Supreme People's Court of China with many public concepts describing cases recorded. For instance, each legal file has a defined category, such as dangerous driving cases, fraud cases, etc. And each category also contains specific terms such as the dangerous behavior of the defendant in a dangerous driving case and the type of vehicle being driven. So the document has high credibility for supporting

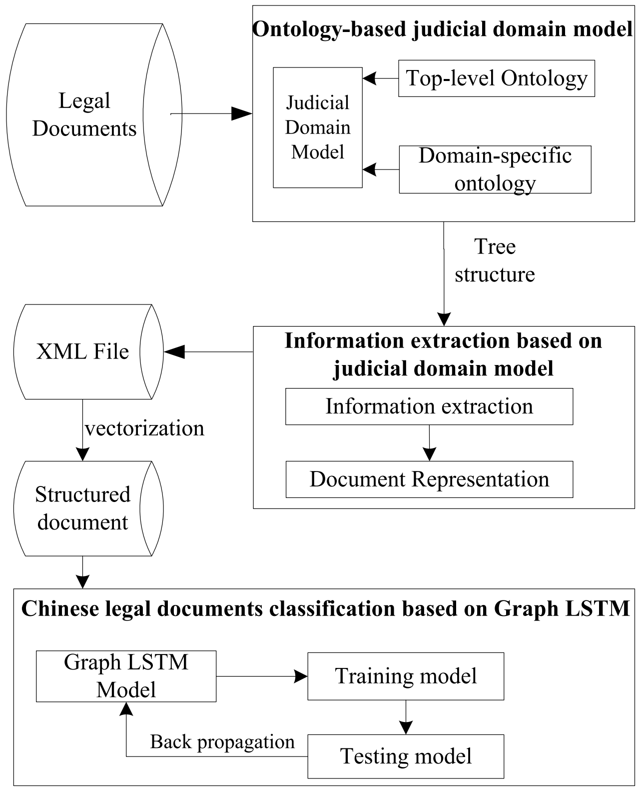


FIGURE 1. The overview of the model framework.

the model. Details about the model will be described in the fourth section of this paper.

B. INFORMATION EXTRACTION BASED ON THE JUDICIAL DOMAIN MODEL

The ontology-based judicial domain model is constructed in view of the following two matters: firstly, the structure in the Chinese judgment document is relatively fixed not only in words but also in contents as the objective aspect, the subjective aspect, and the judgment results are mostly located in fixed paragraphs; secondly, it is troublesome to accurately understand the semantics of legal documents to effectively extract keywords and improve the accuracy of legal document classification. The reason is that a judicial document contains a large amount of information, but not all information is useful for determining the sentence. Generally speaking, the model includes information extraction and knowledge representation. The process is that relevant. concept paragraph is located based on the top-level ontology of judicial domain model, and then more fine-grained knowledge is extracted from the paragraphs according to the domain-specific ontology. It is worth mentioning that the extracted knowledge is stored in an XML file, one text corresponding to an XML file, as the original legal document is unstructured text. Using XML files to represent knowledge, it not only stores the semantic information of legal documents but also facilitates vector conversion. Other information about the model will be described in detail in the fifth section of this article.

C. CHINESE LEGAL DOCUMENT CLASSIFICATION BASED ON GRAPH LSTM

The collected corpus is the original Chinese judgment documents. A series of processing such as word segmentation is carried out to finally represent the document as a set of vectors stored in the nodes of the Graph LSTM. The Graph LSTM model learns the characteristics of each node and uses it for the classification of the current node. It will be described in detail in the sixth section of this article.

IV. ONTOLOGY-BASED JUDICIAL DOMAIN MODEL

The ontology-based judicial domain model consists of two levels of ontologies, namely, the top-level ontology and the domain-specific ontology. Ontology here is mainly used to describe the relationship between concepts in a certain domain to give them a clear, and unique definition within the scope of sharing in accordance with Gruber [23] and Studer *et al.* [24]. In The ontology-based judicial domain model consists of two levels of ontologies, namely, the top-level ontology and the domain-specific ontology. Ontology here is mainly used to describe the relationship between concepts in a certain domain to give them a clear, and unique definition within the scope of sharing in accordance with Gruber [23] and Studer *et al.* [24]. In addition, there are usually two ways to build ontology model: one is a top-down approach, built by domain experts based on their expertise; the other is a bottom-up approach that extracts domain concepts and terminology constructs from appropriate documents. And ontology-based judicial domain model is developed from the two approaches, that is to use a top-down approach to build the top-level ontology, and a bottom-up approach to build domain-specific ontology.

Through careful analysis, the ontology-based judicial domain model is verified to be practicable in the classification of legal trial documents and some legal information systems There are many types of legal documents in China, such as *China Judgments Online* [22], *Faxin* [30] and so on. Among them there are some public categories and common concepts describing cases. including *defendant*, *objective aspect*, etc. The public categories can be abstracted into concepts in the top-level ontology that cover almost all types of cases with the public information proposed and maintained by domain experts. and there are also domain-specific ontologies that describe the concept and knowledge of the case, covering all possible concepts of the case, with reference to top-level ontology and documents of different crime names. The concepts in the domain-specific ontology are contained in the top-level ontology, which is a detailed classification of the concepts in the top-level ontology.

A. TOP-LEVEL ONTOLOGY

The top-level ontology describes some of the common concepts of Chinese legal documents, without paying attention to the specific details of any particular crime. The top-level ontology includes concepts such as the defendant, victim, objective aspects, subjective aspects, sentence result, and basic information, etc. The structure is shown in Figure 2.

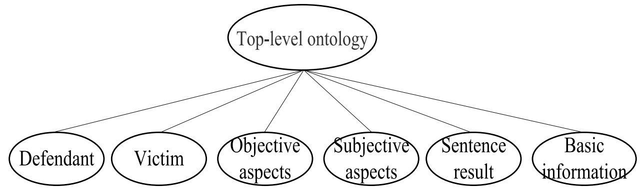


FIGURE 2. **The structure of Top-level ontology.**

A defendant refers to the natural person who commits acts that endanger the society and should be criminally responsible according to law; a victim refers to the social relationship that is protected by criminal law and is violated by criminal acts; the objective aspect is to describe the defendant and its criminal facts in the whole criminal case; the subjective aspect is about the intention or negligence of the criminal subject; the result of the judgment is the responsibility of the criminal subject; the basic information indicates which judicial organization performs the judgment and judicial action.

To sum up, the top-level ontology describes public entities that are commonly used in legal trial documents to make legal document knowledge more systematic. Furthermore, as those public entities are well-defined concepts and terminology of group consensus, the top-level ontology can describe various types of cases richly and play a decisive role in the construction of domain-specific ontology, especially in improving the efficiency of information implementation.

B. DOMAIN-SPECIFIC ONTOLOGY

The domain-specific ontology is formed by extracting keywords of a specific type of case on the basis of the top-level ontology. That is to say, specific ontology in the legal field expresses the terms that are appropriate for a particular type of case. And the specific ontology of the legal field is not constant, and it is necessary to make corresponding adjustments to specific ontology according to the crime types contained in the judgment. That's because Different types of cases and criminal judgment documents vary in terms of defendant, victim, objective aspect and subjective aspect. But it is noticeable that the domain-specific ontology of the same type of case generally does not need to be adjusted, which is for knowledge reuse.

This paper constructs domain-specific ontology for dangerous driving cases and traffic accident cases which is shown in Figure 3 and Figure 4 respectively. In Figure 3, according to the general description of the top-level ontology, the concept of defendant is very important and has an impact on the final sentence, including *identity*, *criminal history,* etc. Objective aspects contain *road type*, *vehicle type*, *dangerous behavior*, etc. The sentence result includes *fines, sentences* and so on; basic information involve *judges*, *courts*, etc.

The domain-specific ontology of traffic accidents is shown in Figure 4. Traffic accidents and dangerous driving cases

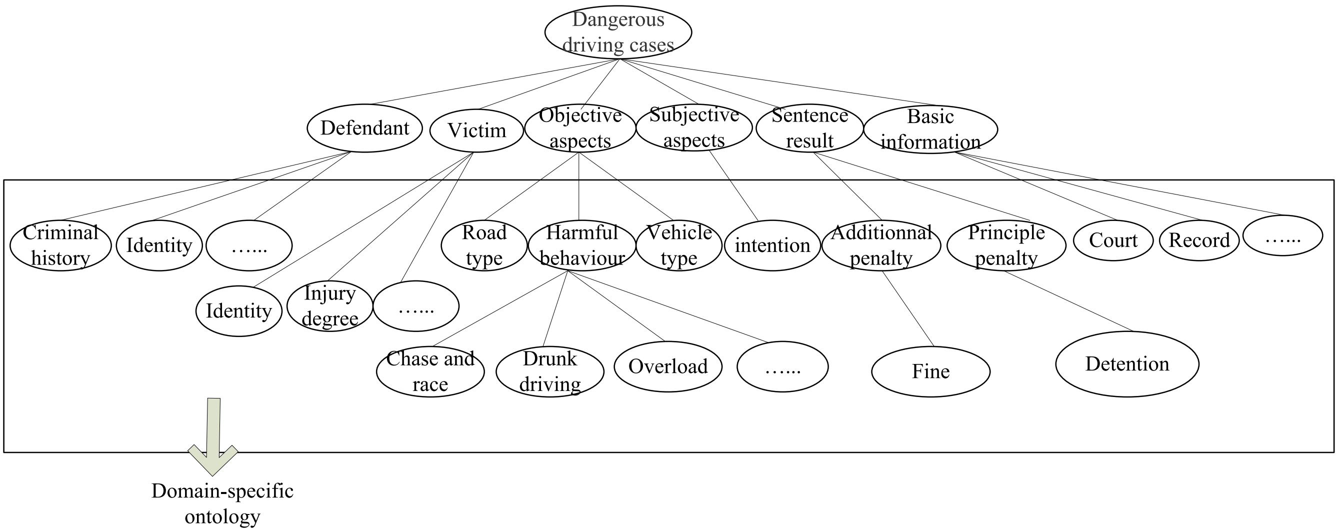


FIGURE 3. The domain knowledge model of dangerous driving cases.

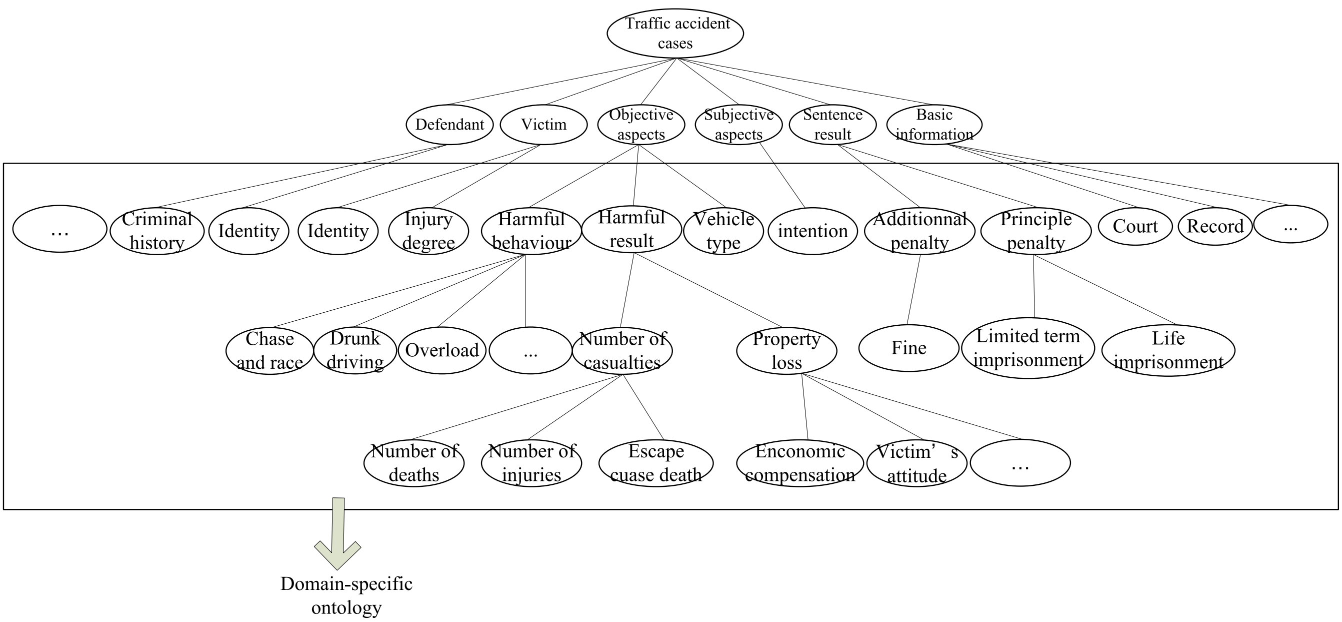


FIGURE 4. The domain knowledge model of traffic accident cases.

differ in specific domains. For example, in the traffic accident case, the judgment results are closely related to whether the death caused, the number of deaths and whether or not to escape and others.

As shown above, the top-level ontology and domain-specific ontology merge to form the judicial domain knowledge model. In the model, concepts in the ontology are classified into hierarchical structures after carefully defined, and the leaf nodes are closely related to the sentencing results. Moreover, the domain-specific ontology can be seamlessly combined with the top-level ontology in the model. For example, traffic accidents and dangerous driving cases have many of the same concepts and attributes, which have improved the reusability of the ontology to some extent. In short, the combining model allows for information extraction and text categorization.

V. INFORMATION EXTRACTION BASED ON JUDICIAL DOMAIN MODEL

In general, information extraction based on the judicial domain model is to extract keywords or key phrases that are called ''popular words'' in the legal field. It is common that legal texts are long and complex in structure, which makes their reading time-consuming and laborious. Therefore, legal practitioners must succinctly state the core legal issues in legal texts.

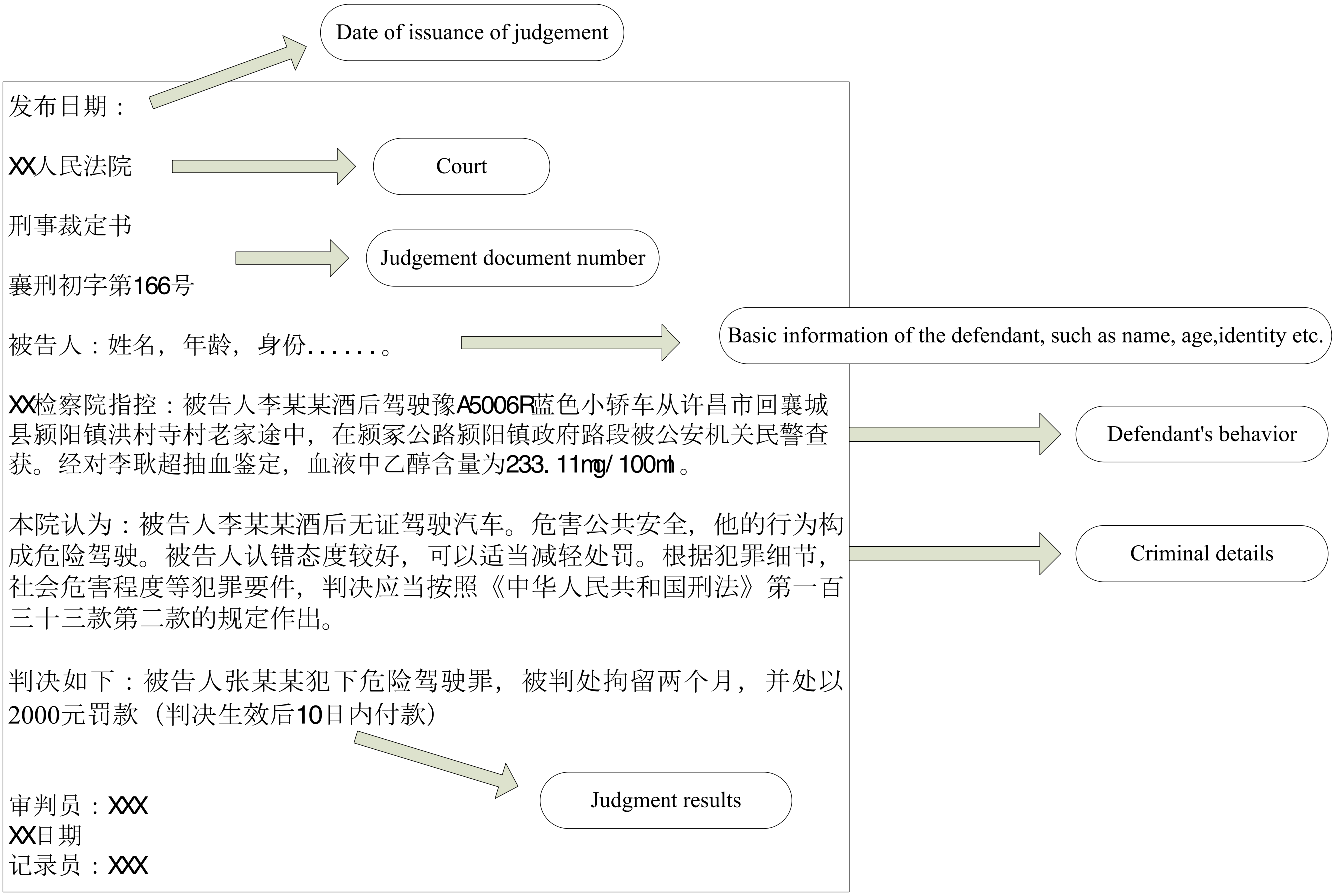


FIGURE 5. The domain knowledge model of traffic accident cases.

A. INFORMATION EXTRACYION

From the above, it is seen that information can be extracted based on mode where the leaf nodes of the judicial domain model describe the core knowledge of a type of legal case. In addition, the judicial domain model is of great help to extract keywords accurately and effectively. The reason lies in that the preparation of legal documents follows certain rules. Although different legal practitioners have different writing habits, basic grammar rules do not change. Therefore, the rules from legal documents can be summarized to facilitate information extraction.

Based on the structural information, some rules can be summarized. For example, the defendant's information mainly appears after the beginning of the document "judgment document number"; most of the harmful acts and criminal details are after the "检察院指控 (The Procuratorate accused):"; the judgment result is usually after the "检察院判决如下 (The Prosecutor's Court's decision is as follows):"; Some basic information (name of the judge, court, date of publication, etc.) is usually at the beginning and end of the document.

In brief, as the structure of Chinese legal documents is relatively fixed, the legal text can be divided into several paragraphs according to these rules and get more detailed keywords in different paragraphs. And supervised learning can be used in keyword extraction on the basis that the positions of key information are easily distinguished in legal texts. Taking the dangerous driving case as an example, and its structure is shown in Figure 5.

From Figure 5 the criminal details and judgment results of a dangerous driving case can be translated into English as follows: "The court confirmed that the defendant Li was driving the car without a license after drinking. He endangered public safety, and his behavior constituted dangerous driving. The defendant's attitude of admitting mistakes was good, and the punishment could be appropriately mitigated. According to the details of the crime, the degree of social harm, etc., the judgment shall be made in accordance with the provisions of Paragraph 2, Paragraph 3, of the Criminal Law of the People's Republic of China. The judgment is as follows: The defendant Li was guilty of dangerous driving and was sentenced to two months of detention and a fine of 2,000¥ (Payment within 10 days after entry into force of the judgment)". According to the domain knowledge model of dangerous driving cases, combined with the above rules, keywords in harmful behaviors are extracted, such as "Drunk driving" and "driving the car without a license" and so on. Keywords such as ''two months of detention'' and ''2,000¥'' can be extracted from the judgment results.

Keyword positioning is time-saving to quickly find the paragraph in which the keyword to be extracted is located without traversing the entire legal document. And it is worth mentioning that word segmentation must be performed before the processing of word frequency statistics and keyword extraction for Chinese documents. This article uses JieBa [25] for word segmentation. It is known that except for the punctuation marks, there is no inherent separator (space) between the words in the sentence in Chinese, while English words are separated by a fixed separator (space) between them. Therefore, it is necessary to conduct word segmentation in English text analysis.

B. STRUCTURED KEYWORDS

After extracting the keywords, it is necessary to summarize and organize the accumulated knowledge to make it organized and programmatic. And the XML language has brought convenience to store the keywords of tens of thousands of legal documents. XML is an extensible markup language that marks electronic files to be built and can be used to mark data and define data types. It is a source language that allows users to define their own markup language [27]. The reasons why the extracted keywords stored as XML files are as follows. One is structured keywords, which make them organized and procedural. The second is to store semantics for facilitating the conversion of vectors.

The semantic information of the XML file is taken from the judicial domain model constructed in the paper. According to the judicial domain model, the parent label of the XML file contains the judgment information, defendant, objective aspects, impacts of harmful behavior, and judgment result. The five parts are determined by the top-level ontology. Meanwhile, sub-labels in domain-specific ontology are nested as attributes under different parent tags. For example, the sub-label of the defendant includes the identity of the defendant, educational background, age, criminal record, and confession attitude. The structure after the conversion is shown in Table 1.

Different from traditional word embedding [41] method, the attributes or the leaf nodes of the judicial domain model are extracted from XML files and converted into vectors. Different nodes are represented by different numbers. For example, in a dangerous driving case, there are three nodes about whether to drive after drinking, including not drinking, drinking but not drunk, getting drunk, which are correspondent to 0, 1, and 2 respectively. If there is drunk driving in the legal documents, the corresponding drunk driving vector is 2. Finally, the transformed vectors are entered as an input vector into the Graph LSTM model.

TABLE 1. Converted XML structure.

Converted XML structure

```
<?xml version="1.0" encoding="UTF-8"?>
<Trial>
    <TrialInfo>
        <Area>
            <Province>XX</Province>
            <city>XX</city>
            <CountyDistrict>XX</CountyDistrict>
            ......
        </Area>
    </TrialInfo>
    <Defendant>
        <Status>masses/party member</status>
        <Pedigree></Pedigree>//criminal record
        ... ...
    </Defendant>
    <ObjectiveAspects>
        <Vehicle>freight car/car/motor/....</Vehicle>
        <Plot>
            <DrunkDriving></DrunkDriving>
            <Alcohol></Alcohol>
            ... ...
        </Plot>
    </ObjectiveAspects>
    <HarmfulConsequence>
        <Casualties>
            <DeathNum></DeathNum>
            <InjuredNum></InjuredNum>
        </Casualties>
        <PropertyLosses>
            <Fine></Fine>
        </PropertyLosses>
        ... ...
    </HarmfulConsequence>
    <TrialResult>
        <Dentention></Dentention>
        ... ...
    </TrialResult>
</Trial>
```

VI. CLASSIFICATION OF LEGAL DOCUMENTS BASED ON GRAPH LSTM

It is a classification task to predict the sentence of legal trial documents and recommend similar cases to judicial officials [42]. Graph LSTM is employed to classify legal documents based on sentencing in the paper.

A. THE STRUCTURE OF GRAPH LSTM

Many prediction problems can be naturally expressed as inference problems on local neighborhoods of graphs [44]. And Graph LSTM makes it possible to learn predictions directly from examples bypassing the steps of creating and adjusting reasoning models. In this paper, Graph LSTM neural network architecture is utilized based on LSTM. LSTM learns how to aggregate neighborhoods into radius D based on data, avoiding manually synthesizing a set of fixed features. By applying an LSTM to each level, the learning of LSTM is adjusted according to the distance of the target node. Finally, the graph structure is generated.

A graph $G = (V, E)$ is considered with vertex set V and edge $\mathrm{E} \subseteq V \times V$, assuming each vertex $v \in \mathrm{E}$ is marked with a feature vector $\mathrm{g}(v)$. The edge represents the association between two nodes. $\mathrm{P}(v)$ represents the feature label of the v. The graph represents the interaction between nodes, and each node's neighborhood contains information that allows inference or prediction. And the prediction of

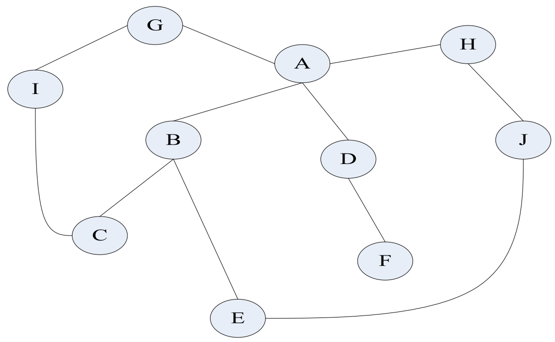


FIGURE 6. The structure of Graph LSTM.

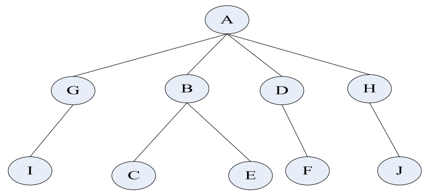


FIGURE 7. The extended tree structure of Graph LSTM.

nodes is generated in accordance with the structure of its local neighborhood and the characteristics of its nodes. Figure 6 shows the abstract structure of Graph LSTM, where each node stores a feature vector related to the sentence. Each node represents a feature vector of a different legal text.

B. FORWARD AND BACKWARD PROPAGATION

To predict the feature label of node V, the graph is expanded to a tree rooted at node v, defined as T_v, with a depth of D. As shown in Figure 6 where the graph $G = (V, E)$ and the node $v \in V$, we expand G into a tree T_A, the depth is D (here we set D to 2). As shown in Figure 7, node A has a depth of zero, T_v is traversed from the bottom-up, and the feature label of each parent node is calculated from the characteristics of its child nodes. Finally, the output label y_v of root V is generated. In training, the output y_v can be compared with the desired output $L(v)$. Those steps just mentioned are introduced in length in the following part.

Forward Propagation. Forward propagation proceeds along the tree T_A from bottom to top [44]. Figure 8 shows how the graph LSTM is applied to the expansion of Figure 6 with depth 2 to produce a prediction for node A. Consider two situations. 1) Depth d = D. Consider a node v of depth D-1 with children node $u_1, u_2, \ldots, u_k$ at depth D. Graph LSTM is used to aggregate the children node feature vector $g(u_1), g(u_2), \ldots, g(u_k)$ into the feature vector $f(v)$ of v. 2) Depth $0 < d < D$. Consider a node v with a depth of d-1 whose children are $u_1, u_2, \ldots, u_k$, the depth is d, and whose feature vector is $g(u_1), g(u_2), \ldots, g(u_k)$. The feature vector $g(u_1) \sim f(u_1), \ldots, g(u_k) \sim f(u_k)$ is obtained by connecting the feature vector $f(u_i)$ of the aggregated child node with the node feature vector $g(u_i)$, where the feature vector $f(u_i)$ is calculated by the model at depth d + 1. Finally, the feature vector $f(v)$ of v is generated.

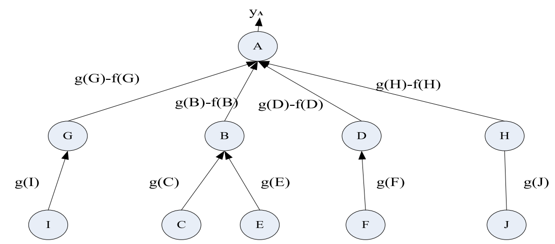


FIGURE 8. Forward propagation of Graph LSTM.

A forward propagation doesn't end until the root node is traversed. The predicted root node label is y_v.

Backward Propagation. Once the feature label y_v of the root T_v is obtained, the losses $L(y)$ and $\partial L / \partial y$ can be calculated. Then, according to the topology of the tree, this loss is propagated from the root down to the leaves of T_v (see Figure 8 again). The node v at the depth d ($0 < d \leq D$) has a feature vector $f(v)$. Backward propagation is performed by calculating the loss of f(v) to obtain $\partial L / \partial f(u_i)$ of the child node $u_1, u_2, \ldots, u_k$. At the depth d = D of the tree, the feature vector of each node is $g(u_i)$, and backward propagation is terminated.

Parameter Update. Consider the node depth $1 \leq d \leq D$ and define the parameter $w(d)$. Given the nodes $u_1, u_2, \ldots, u_k$ of depth d, in order to update the parameter $w(d)$, for each $f(u_i)$, the parameter update $\Delta_{u_i} w(d)$ is obtained by calculating $\partial L / \partial f(u_i)$. Then, the parameters of the global node of the depth d can be calculated by using the updated average $\Delta w(d) = (\Delta_{u1} w(d) + \Delta_{u2} w(d) + \ldots + \Delta_{uk} w(d)) / k$ on each node.

For backward propagation and parameter updates, it is necessary to preserve the state of each node after the step of forward propagation. The kind of preservation is the memory unit described above.

C. PARAMETER SETTINGS OF GRAPH LSTM

A fixed-length vector is taken as input, with a total of 25200 vectors, batch size set to 1000. The execution process is as follows: every time the root node is randomly selected to construct the tree, the depth is 3, and propagations are forward along the tree; the output contains the final predicted value, memory unit and a cache containing useful state for back propagation; the depth of the tree is 3, which ensures that the neighborhood of the node is not too simple or complex; and the network parameters are updated after each batch. The process was executed 50 times at random. Moreover, the softmax activation function and the adaptive learning rate algorithm (Adagrad) are used to improve the stochastic gradient descent (SGD) algorithm so as to avoid manual adjustment of the learning rate. In addition, compared with only using Adagrad algorithm, Graph LSTM neural network is more renewable, and the ability of LSTM model to learn more knowledge is stronger. The learning factor of Adagrad algorithm is set to 1.0, and the attenuation parameter is set to 0.95. The number of iterations is 2.

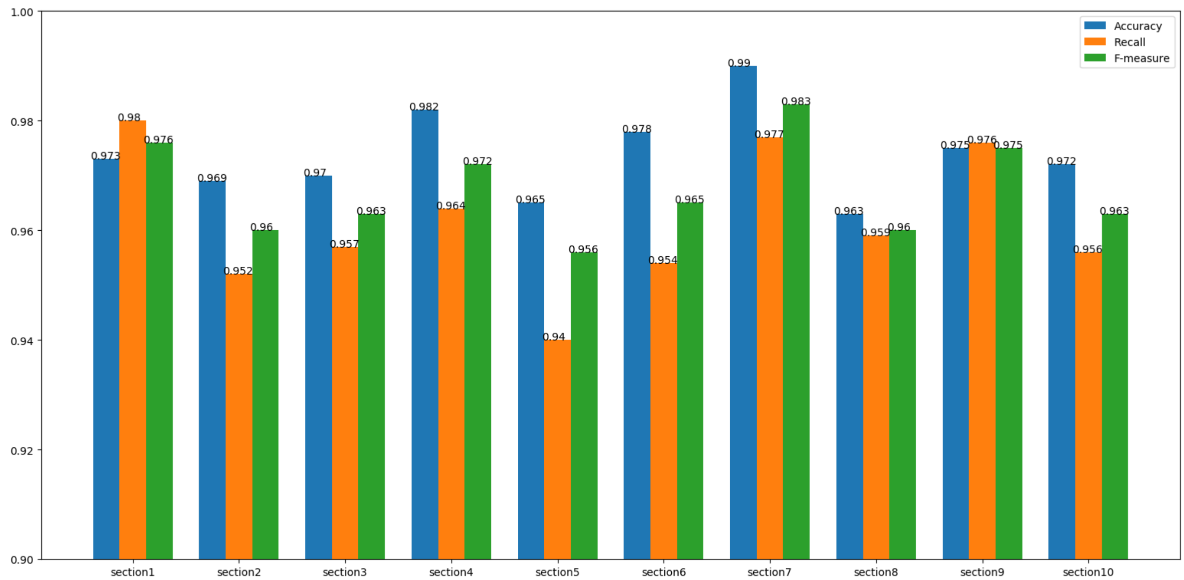


FIGURE 9. The classification results of dangerous driving case data set.

VII. EXPERIMENTS

A. DATA SET

The data set for the paper is provided by the business company of the project partner, including dangerous driving cases and traffic accident cases. Among them, there were about 25,200 dangerous driving cases and about 24,700 traffic accidents. Every document is a real trial. Those judgment documents are classified in accordance with the sentence and divided into four types, namely, criminal detention, three years or less, three to seven years and seven years or more of imprisonment.

The data set is divided into the training set and the testing set. To prevent over-fitting and to get as much valid information as possible from limited data, the ten-fold cross-validation is used to test the accuracy of the algorithm. The data set is divided into ten parts, nine of which are taken as the training set and the one remaining was tested as the testing set. The average of the correctness rates of the ten results is used as an estimate of the accuracy of the algorithm. The following experiments all use ten-fold cross-validation.

B. EXPERIMENTAL ENVIRONMENT

The experimental evaluations were made on the computer environment with a 2.8GHz processor 8GB RAM and the Python programming language.

C. COMPARATIVE EXPERIMENT

This experiment was verified using two data sets for dangerous driving and traffic accidents. And the neighbor node with a target node radius of 2 is selected to train the Graph LSTM.

The algorithm on a data set of dangerous driving cases and traffic accident cases is verified by using a ten-fold cross-validation method. The data including Accuracy, Recall, and F-measure are calculated for each validation. The experimental results are shown in Figure 9 and Figure 10.

The x-axis is the number of experiments and the *section i* is the *i*-th experiment. The y-axis is the calculated value. Blue cylinders represent Accuracy, orange represents Recall, and the green represents F-measure.

Evaluate by calculating the accuracy, recall, and F-measure of the model. The calculation method is as follows.

TP: The number of cases where the predicted value is positive and the actual value is positive;

TN: The number of cases where the predicted value is negative and the actual value is negative;

FP: The number of cases where the predicted value is positive and the actual value is negative;

FN: The number of cases where the predicted value is negative and the actual value is positive.

$$\text{Accuracy (A)} = \frac{TP + \mathrm{TN}}{TP + TN + FN + FP} \tag{1}$$

$$\text{Recall (R)} = \frac{TP}{TP + FN} \tag{2}$$

$$\text{Precision (P)} = \frac{\mathrm{TP}}{TP + FP} \tag{3}$$

$$\text{F-measure} = \frac{2\mathrm{P}^{*}\mathrm{R}}{P + R} \tag{4}$$

Since the selection of nodes is random during Graph LSTM training, the average of multiple trials has been calculated. The average Accuracy rate in the dangerous driving data set is about 0.975, the average Recall is about 0.959, and the average F-measure value is about 0.9673. And the average Accuracy classified on the traffic accident data set is about 0.941, the average Recall is about 0.936, and the average F-measure is about 0.937. From the above figures, it is seen that the classification Accuracy, Recall, and F-measure are

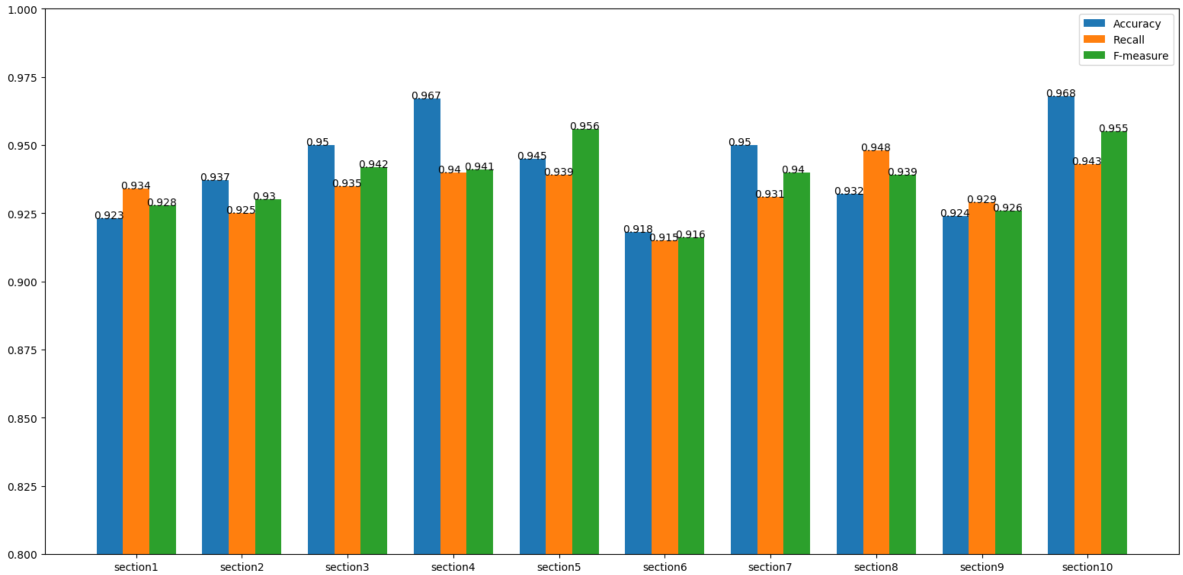


FIGURE 10. The classification results of a traffic accident case data set.

TABLE 2. Accuracy verification.

Method	Dangerous driving data set	Traffic accident data set
OJDM	0.96	0.93
SVM	0.74	0.69

higher than 90% in both the dangerous driving data set and the traffic accident data set.

In order to verify the effectiveness of the proposed method, a comparative experiment is also needed. The comparative experiment consists of two aspects, one is to verify the accuracy of the ontology-based legal model proposed in this paper, and the other is to verify the performance of the Graph LSTM algorithm.

The accuracy verification of the model is mainly to verify the validity of the key information extracted according to the legal model, which is reflected in the accuracy of the classification. The specific operation is to input the vector extracted by the ontology-based judicial domain model (OJDM) into the Graph LSTM. Original legal texts are imported into the SVM. The accuracy is shown in Table 2.

It can be seen from the table that the method proposed in this paper has higher classification accuracy. And it is evidence-based that the judicial model constructed in the paper is a good representation of the semantic information of legal trial documents.

Verifying the performance of the Graph LSTM algorithm is primarily to verify the accuracy and efficiency of the algorithm. Comparative methods include SVM and LSTM. SVM is implemented through python third-party module sklearn while LSTM is implemented by keras which is also a python third-party module.

TABLE 3. Accuracy comparison.

Symbol	Dangerous driving data set	Traffic accident data set
Graph LSTM	0.975	0.941
SVM	0.883	0.862
LSTM	0.776	0.719

The evaluation indicators are accuracy and efficiency.

1) ACCURACY

The purpose of accuracy evaluation is to verify whether the method proposed can effectively classify Chinese legal judgment documents. The LSTM is trained with mini-batch Stochastic gradient descent (MSGD). The batch size is set to 18 and iterates up to 30 times. And the average of the accuracy of multiple classification methods is calculated, as shown in Table 3.

It can be seen from the table that the accuracy of Graph LSTM is higher than that of SVM and LSTM on both data sets. So Graph LSTM has a better classification ability.

2) EFFICIENCY

The quality of an algorithm will affect the efficiency of the algorithm and the entire program. The efficiency of the method proposed in the paper is evaluated by comparing the execution time as shown in Figure 11.

As can be seen from the figure 11, the average execution time of the Graph LSTM running 10 times is 79091ms, the average execution time of the SVM is 78820ms, and the average execution time of the LSTM is 101608ms. Above all, Graph LSTM runs close to SVM. The LSTM model runs the

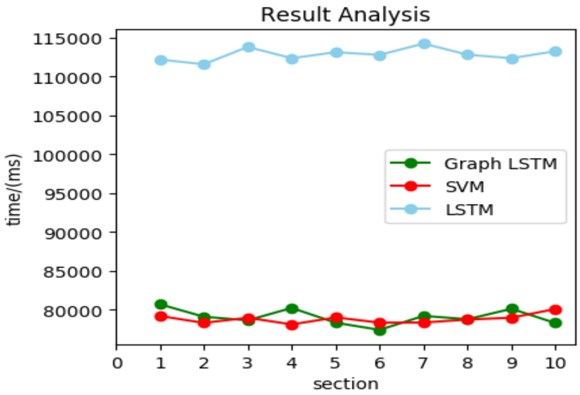


FIGURE 11. Execution time.

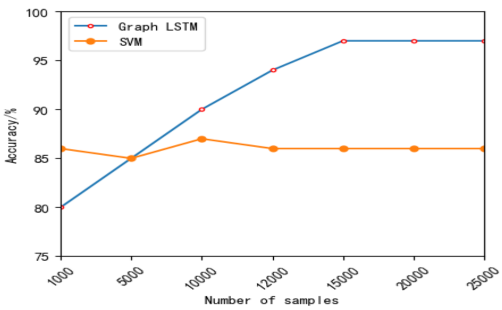


FIGURE 12. Classification effect under different data scales.

longest because the number of iterations should be larger in order to achieve higher LSTM accuracy.

The overall experimental results show that the proposed method of domain knowledge extraction combined with Graph LSTM has high computational efficiency and accuracy. The method can effectively and efficiently classify Chinese judgment documents, help judges to refer to sentencing and assist decision-making, so as to achieve similar cases with similar trial results.

This paper also explores the impact of the size of the data set on the classification of Graph LSTM through experiments and compares it with the traditional machine learning method. As can be seen from Figure 12, in the case of a small number of samples, Graph LSTM is limited by the size of the data set, and the classification effect is not as good as the traditional machine learning method; when the data set size is gradually increased, the classification effect of Graph LSTM is rapidly improved. After the data set reaches 15,000 copies, the classification effect is no longer improved, which is a common phenomenon of the deep learning model. The classification effect of SVM does not change much.

VIII. CONCLUSION

In order to realize the application of similar case recommendation and provide intelligent assistant case handling services for judicial personnel, this paper proposes a legal document classification method combining domain knowledge extraction with Graph LSTM. A legal domain knowledge model is constructed through the open categories of Chinese legal documents, and then semantic information is extracted effectively based the model and key paragraph positioning and stored in XML files. Finally, Graph LSTM is used for document classification. The data set for the paper includes approximately 50,000 original legal documents including dangerous driving cases and traffic accident cases. The experiments show that our method has an accuracy rate of over 90%, which improves the accuracy of classification and greatly reduces the time required for classification.

The contributions of this paper are as follows.

1) Ontology models are built in the legal domain by reusing and adapting existing public categories, including top-level ontology and domain-specific ontology. It consists of legal knowledge and can be extended and reused. In addition, text features are extracted based on the domain ontology model and converted into vectors, which reduce the dimensions and have strong feature representation.

2) The Graph LSTM model is applied to the judicial field to classify Chinese legal documents.

The future work outlook includes: 1) the latest methods of recommending similar cases are waiting to be found and combined with our method; 2) more Chinese judgment document classification methods are required to be explored based on semantic analysis; 3) the classification can be combined with a graphical interface for a more convenient service.

REFERENCES

[1] K. Williams, J. Wu, and C. L. Giles "SimSeerX: A similar document search engine," in *Proc. ACM Symp. Document Eng.*, 2014, pp. 143–146.

[2] Z. Zhang, L. Wang, X. Xie, and H. Pan, "A graph based document retrieval method," in *Proc. IEEE 22nd Int. Conf. Comput. Supported Cooperat. Work Design (CSCWD)*, May 2018, pp. 426–432.

[3] M. Mannai, W. Abdessalem, and H. B. Ghezala, "Overview of information extraction using textual case-based reasoning," in *Mining Multimedia Documents*, vol. 1. Boca Raton, FL, USA: CRC Press, Jan. 2017, pp. 81–91.

[4] P. R. Varshavsky, A. K. Myo, and D. V. Shunkevich, "Classification and clustering methods for improving efciency of case-based systems," *Softw. Syst.*, vol. 30, no. 4, pp. 625–631, 2017.

[5] F. Galgani, P. Compton, and A. Hoffmann, "LEXA: Building knowledge bases for automatic legal citation classification," *Expert Syst. Appl.*, vol. 42, no. 17, pp. 6391–6407, 2015.

[6] C.-Y. Chang, S.-J. Lee, and C.-C. Lai, "Weighted word2vec based on the distance of words," in *Proc. Int. Conf. Mach. Learn. Cybern. (ICMLC)*, Jul. 2017, pp. 563–568.

[7] W. Tian, J. Li, and H. Li, "A method of feature selection based on word2vec in text categorization," in *Proc. 37th Chin. Control Conf. (CCC)*, Jul. 2018, pp. 9452–9455.

[8] M. Bilgin and I. F. Şentürk, "Sentiment analysis on Twitter data with semi-supervised Doc2Vec," in *Proc. Int. Conf. Comput. Sci. Eng. (UBMK)*, Oct. 2017, pp. 661–666.

[9] G. Karystianis, K. Thayer, M. Wolfe, and G. Tsafnat, "Evaluation of a rule-based method for epidemiological document classification towards the automation of systematic reviews," *J. Biomed. Inform.*, vol. 70, pp. 27–34, Jun. 2017.

[10] R. Giri, Y. Porwal, V. Shukla, P. Chadha, and R. Kaushal, "Approaches for information retrieval in legal documents," in *Proc. 10th Int. Conf. Contemp. Comput. (IC)*, Aug. 2017, pp. 1–6.

[11] D. Thenmozhi, "A text similarity approach for precedence retrieval from legal documents," in *Proc. CEUR Workshop*, vol. 2036, 2017, pp. 90–91.

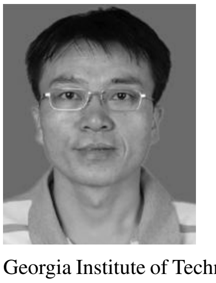
[12] A. Sundaram, L. Sundaram, D. Z. Wang, W. F. Hamilton, K. Branting, and C. Pfeifer, "Automatic semantic edge labeling over legal citation graphs," *Artif. Intell. Law*, vol. 26, no. 2, pp. 127–144, Jun. 2018.

[13] E. Ong and Y. He, "OLOBO: A new ontology for linking and integrating open biological and biomedical ontologies," in *Proc. 8th Int. Conf. Biomed. Ontol. (ICBO)*, vol. 2137, 2017, pp. 1–2.

[14] Q. Chen, L. Yao, and J. Yang, "Short text classification based on LDA topic model," in *Proc. Int. Conf. Audio, Lang. Image Process. (ICALIP)*, Jul. 2016, pp. 749–753.
[15] M. Farhoodi, A. Yari, and A. Sayah, "N-gram based text classification for Persian newspaper corpus," in *Proc. 7th Int. Conf. Digit. Content, Multimedia Technol. Appl. (IDCTA)*, Aug. 2011, pp. 55–59.
[16] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2011, pp. 151–161.
[17] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, "Semantic compositionality through recursive matrix-vector spaces," in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn. (EMNLP-CoNLL)*, 2012, pp. 1201–1211.
[18] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations fromtree-structured long short-term memory networks," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2015, pp. 1556–1566.
[19] J. Wu, S. Guo, J. Li, and D. Zeng, "Big data meet green challenges: Big data toward green applications," *IEEE Systems J.*, vol. 10, no. 3, pp. 888–900, Sep. 2016.
[20] R. Brueckner and B. Schulter, "Social signal classification using deep BLSTM recurrent neural networks," in *Proc. ICASSP*, May 2014, pp. 4823–4827.
[21] Y. Ma, P. Zhang, and J. G. Ma, "An ontology driven knowledge block summarization approach for Chinese judgment document classification," *IEEE Access*, vol. 6, pp. 71327–71338, 2018.
[22] *China Judgments Online*. Accessed: Jul. 1, 2019. [Online]. Available: http://wenshu.court.gov.cn/
[23] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," *Int. J. Hum.-Comput. Stud.*, vol. 43, no. 5, pp. 907–928, 1993.
[24] R. Studer, V. R. Benjamins, and D. Fensel, "Knowledge engineering: Principles and methods," *Data Knowl. Eng.*, vol. 25, nos. 1–2, pp. 161–197, 1998.
[25] J. J. Sun. *Jieba Chinese Word Segmentation Tool*. Accessed: Jul. 1, 2019. [Online]. Available: https://github.com/fxsjy/jieba
[26] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*. [Online]. Available: https://arxiv.org/abs/1412.3555
[27] *XMLtutorial*. Accessed: Jul. 1, 2019. [Online]. Available: http://www.runoob.com/xml/xml-tutorial.html
[28] O. Corcho, M. Fernández-López, A. Gómez-Pérez, and A. López-Cima, "Building legal ontologies with methontology and WebODE," *Law and the Semantic Web* (Lecture Notes in Computer Science), vol. 3369. Berlin, Germany: Springer, 2005, pp. 142–157.
[29] B. Joost and H. Rinke, "Ontology-based discovery of responsibility and causality in legal case descriptions," in *Proc. 17th Annu. Conf.*, Amsterdam, The Netherlands, 2004, pp. 56–68.
[30] The Supreme People's Court of China. *The Faxin Online*. Accessed: Jul. 1, 2019. [Online]. Available: http://www.faxin.cn/
[31] B. Maleszka, "A method for determining ontology-based user profile in document retrieval system," *J. Intell. Fuzzy Syst., Appl. Eng. Technol.*, vol. 32, no. 2, pp. 1253–1263, 2017.
[32] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of CNN and RNN for natural language processing," Feb. 2017, *arXiv:1702.01923*. [Online]. Available: https://arxiv.org/abs/1702.01923
[33] V.-S. Boonchom and N. Soonthornphisaj, "Legal ontology construction using ATOB algorithm," in *Proc. Int. Conf. Bus. Inf. Syst.*, in Lecture Notes in Business Information Processing, vol. 57, 2010, pp. 268–279.
[34] T. Sun, Y. Peng, F. Yang, H. Sun, N. Fan, and B. Zhang, "A self-adaptive text classification method based on multiple word embedding and neural network," *ICIC Express Lett.*, vol. 11, no. 6, pp. 1133–1141, Jun. 2017.
[35] Y. Liu, X. Liu, and H. Shen, "Towards early identification of online rumors based on long short-term memory networks," *Inf. Process. Manage.*, vol. 56, no. 4, pp. 1457–1467, 2018.
[36] G. Boella, L. Di Caro, L. Humphreys, L. Robaldo, P. Rossi, and L. van der Torre, "Eunomos, a legal document and knowledge management system for the Web to provide relevant, reliable and up-to-date information on the law," *Artif. Intell. Law*, vol. 24, no. 3, pp. 245–283, 2016.
[37] A. R. Sadik and B. Urban, "An ontology-based approach to enable knowledge representation and reasoning in worker–cobot agile manufacturing," *Future Internet*, vol. 9, no. 4, p. 90, Dec. 2017.
[38] S. Sarwar, R. García-Castro, Z. U. Qayyum, M. Safyan, and R. F. Munir, "Ontology-based learner categorization through case-based reasoning and fuzzy logic," in *Proc. Int. Conf. e-Learn.* 2017, pp. 63–159.
[39] S. Fan, Z. Hua, V. C. Storey, and J. L. Zhao, "A process ontology based approach to easing semantic ambiguity in business process modeling," *Data Knowl. Eng.*, vol. 102, pp. 57–77, Mar. 2016.
[40] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.
[41] C. S. Şahin, R. S. Caceres, B. Oselio, and W. M. Campbell, "Challenges and solutions with alignment and enrichment of word embedding models," in *Proc. 22nd Int. Conf. Appl. Natural Lang. Inf. Syst. (NLDB)*, in Lecture Notes in Computer Science, vol. 10260, 2017, pp. 260–266.
[42] L. Zakaria, C. Salim, and L. Saliha, "Scalability property in solving the density classification task," *J. Inf. Technol. Res.*, vol. 10, no. 2, pp. 60–76, Apr. 2017.
[43] N. Zhang, Y.-F. Pu, S.-Q. Yang, J.-L. Zhou, and J.-K. Gao, "An ontological Chinese legal consultation system," *IEEE Access*, vol. 5, pp. 18250–18261, 2017.
[44] R. Agrawal, L. de Alfaro, and V. Polychronopoulos, "Learning from graph neighborhoods using LSTMs," in *Proc. AAAI Workshop*, vol. 15, 2017, pp. 377–384.
[45] J. Wu, S. Guo, H. Huang, W. Liu, and Y. Xiang, "Information and communications technologies for sustainable development goals: State-of-the-art, needs and perspectives," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2389–2406, 3rd Quart., 2018.
[46] L. Rahman, N. Mohammed, and A. K. Al Azad, "A new LSTM model by introducing biological cell state," in *Proc. 3rd Int. Conf. Elect. Eng. Inf. Commun. Technol. (ICEEICT)*, Sep. 2016, pp. 1–6.

GUODONG LI received the Ph.D. degree in control theory and control engineering from North China Electric Power University, China, in 2017, where he is currently an Associate Professor with the School of Control and Computer Engineering. His research interests include computer network application and service computing.

ZHE WANG is currently pursuing the master's degree in computer application technology with North China Electric Power University, Beijing, China. Her research interests include knowledge engineering, natural language processing, and big data analysis and processing.



YINGLONG MA (M'13) received the Ph.D. degree in computer science from the Institute of Software, Chinese Academy of Sciences, Beijing, China, in 2006, and the M.S. degree in computer science from Northwestern University, Xi'an, China, in 2002.

From 2006 to 2009, he was an Associate Professor with North China Electric Power University, Beijing. From 2010 to 2011, he was a Visiting Professor with the School of Computer Science, Georgia Institute of Technology, Atlanta, GA, USA. Since 2016, he has been a Full Professor with the School of Control and Computer Engineering, North China Electric Power University. He has coauthored two books, more than 70 articles in some international and domestic journals and conferences, and more than seven inventions. His research interests include artificial intelligence and knowledge engineering, big data analysis and processing, and software engineering. He is also a member of the ACM and a Senior Member of the China Computer Foundation.