D2GCLF: Document-to-Graph Classifier for Legal Document Classification

Qiqi Wang[1†], **Kaiqi Zhao**[1‡], **Robert Amor**[1‡], **Benjamin Liu**[2‡], **Ruofan Wang**[1†]
[1]School of Computer Science, Faculty of Science
[2]Department of Commercial Law, Faculty of Business and Economics
University of Auckland, New Zealand
[†]{qwan857,rwan551}@aucklanduni.ac.nz,
[‡]{kaiqi.zhao,trebor,b.liu}@auckland.ac.nz

Abstract

Legal document classification is an essential task in law intelligence to automate the labor-intensive law case filing process. Unlike traditional document classification problems, legal documents should be classified by reasons and facts instead of topics. We propose a Document-to-Graph Classifier (D2GCLF), which extracts facts as relations between key participants in the law case and represents a legal document with four relation graphs. Each graph is responsible for capturing different relations between the litigation participants. We further develop a graph attention network on top of the four relation graphs to classify the legal documents. Experiments on a real-world legal document dataset show that D2GCLF outperforms the state-of-the-art methods in terms of accuracy.

1 Introduction

Legal Artificial Intelligence (LegalAI) (Zhong et al., 2020; Rissland et al., 2003) is a specific subject to apply the artificial intelligence technology into legal tasks including legal judgement prediction (Rosca et al., 2020; Gan et al., 2021), similar case matching (Tran et al., 2019) and law case classification (Noguti et al., 2020; Lin et al., 2012; Li et al., 2019a). In this paper, we focus on law case classification. The traditional case filing process requires experts to categorize the civil complaints manually, which is labor-intensive. For example, a court in a small town has more than 10,000 civil complaint cases per year in China. This calls for accurate machine learning techniques to improve the efficiency of the case filing process.

Recently, deep learning approaches (Wang et al., 2018a; Johnson and Zhang, 2017; Wang, 2018; Shen et al., 2018; Zhang et al., 2018; Lin et al., 2017) have significantly improved the accuracy of text classification on well-known datasets, e.g., Amazon reviews. Existing methods learn a latent representations for each document by considering the semantics and themes of the documents. In this way, documents are classified into high-level topics such as "Sport" and "Medical". Unlike traditional datasets, legal documents of different types exhibit high semantic similarity. For example, Table 1 shows two semantically similar cases in different Chinese civil categories.

Given the nature of legal classification tasks, there are two limitations of applying existing text classification models. First, document structure is a key to accurate classification, but it is ignored in most existing methods. Legal documents in different categories may only differ in some particular parts. Some previous research proposed to utilize the graph with sentence relations to address the problem (Hu et al., 2019). But not all sentences are relevant to the document class. Second, the facts and reasons in the legal cases are essential for distinguishing different dispute types, but they are not considered in existing methods. Existing word co-occurrence graphs (Zhang and Zhang, 2020; Ragesh et al., 2021; Zhang et al., 2020) are not only hard to represent the key facts but also include words that are irrelevant to the document class.

We argue that the key to accurate legal document classification is to understand the facts, which are expressed as relations between entities. For example, the debtor-creditor relationship denotes the fact that someone borrows other entities' money. Motivated by this, we designed a Document-to-Graph Classifier (D2GCLF) to address the two limitations above. D2GCLF extracts four graphs from *each* legal document to represent the facts about the key entities, i.e., plaintiffs and defendants. The four graphs are (1) *Entity-Matter* graph to describe the matters (e.g., money) associated with the entities; (2) *Entity-Action* graph to describe the actions of entities; (3) *Entity-Keyword* graph to model the general topic of the facts about

Findings of the Association for Computational Linguistics: NAACL 2022, pages 2208 - 2221

Table 1: Examples of translated legal documents of two different types in Chinese law. See Appendix A for the original Chinese documents. The class "Dispute Over Contract of Sale of Commercial Residential Housing Property" (DOCSCRH) covers extremely similar words as the other class "Dispute Over Contract for Sale and Purchase of Housing Property" (DOCSPHP). The only evidence to distinguish the two classes is the seller's identity – only real estate companies sell commercial residential house. For ethical concerns, we hide the parties' names for both cases.

Class	Documents
DOCS CRH	**Plaintiff:** LF **Defendant:** GDYZ co.,ltd **Fact and Reason:** On June 3, 2016, the plaintiff and the defendant signed a contract to purchase a house located in D City with a construction area of 19.36 square meters and a purchase amount of 367,724 yuan. The defendant promised to deliver the relevant documents of the commercial housing transfer registration to the plaintiff before May 15, 2018, which was overdue Liquidated damages were paid according to one ten thousandth of the total purchase price per day; after the signing of the contract, the plaintiff paid the defendant all the purchase price, but the defendant has not yet delivered to the plaintiff the relevant documents for the transfer of commercial housing. The defendant' s delay in handling the housing ownership certificate breached the contract, which harmed the interests of the plaintiff. Therefore, a lawsuit was filed in the court.
DOCS PHP	**Plaintiff:** ZB **Defendant:** YJ **Defendant:** WS **Fact and Reason:** The two defendants are relatives, and WS was YJ's mother-in-law. On November 24, 2015, the plaintiff and the defendant reached an agreement to purchase a house. On November 27, 2015, the plaintiff and the defendant signed a house purchase contract. On December 3, 2015, the plaintiff paid the purchase price and has been living until now. At the beginning of 2019, the plaintiff failed to urge the defendant to go through the house transfer procedures, so he sued to the court.

the two entities; and (4) *Semantic Role Labeling* (SRL) graph to model broader relations including those among third-party persons and facts. We will elaborate on each graph in Section 3.2. To learn the document representation, D2GCLF combines the four graphs and passes them to a graph representation learning module, based on the idea of graph attention networks (Velickovic et al., 2018). Then, D2GCLF uses the document representation for classification. Our main contributions include:

- We propose a new idea of leveraging facts, i.e., the relations between entities, for legal document classification.
- We propose a novel document-to-graph model for legal document classification.
- We conduct extensive experiments on a real-world legal document dataset. The proposed model outperforms the state-of-the-art text classifiers.

2 Related Work

2.1 Traditional Classification Methods

Traditional text classifiers are based on machine learning models, including Naïve Bayes, Logistic regression, etc. Naïve Bayes assumes the words in a document are independent given the class and estimates the class label with Maximum a Posteriori (Zhang and Hawkins, 2018; Fang et al., 2020). Logistic regression (Genkin et al., 2005; Ifrim et al., 2008; Panckevičius and Marcinkevičius, 2016) and Support Vector Machine (SVM) (Sathe and Aggarwal, 2019; Pranckevičius and Marcinkevičius, 2016) find decision boundaries for the document classes in the feature space. Bagging (Li et al., 2011) and AdaBoost (Bloehdorn and Hotho, 2004) classifiers ensemble multiple classification models. The bagging model chooses the best result of multiple sub-classifiers as the final decision. AdaBoost uses the sub-classifiers to focus on the fallible classification cases when training. In the legal domain, some early studies utilize these models to classify judgments (Lin et al., 2012; Li et al., 2019a).

2.2 Deep Neural Networks

Deep neural network classifiers learn latent semantic representations of the input documents, which are then used for classification. Convolutional Neural Networks (CNNs) perform convolution operations for capturing the local context in the latent representation (Li et al., 2019b; Wang et al., 2018b). Recurrent Neural Networks (RNNs) consider the sequential information of a sentence (Das et al., 2019). Long-Short Term Memory (LSTM), Gated Recurrent Unit (GRU), and other modified RNNs (Liu et al., 2016) address the gradient vanishing problem of RNN with well-designed gates in a recurrent unit. Hybrid models, such as Recurrent Convolutional Neural Networks (RCNN), combine RNN and CNN. RCNN is more robust to noise when encoding the sequential information (Lin et al., 2018).

Bidirectional Encoder Representations from Transformers (BERT) series are widely utilized in

NLP tasks. BERT is proven effective in many LegalAI tasks, such as judgment prediction. Previous studies of BERT in legal domain include the legal BERT(Cui et al., 2021; Chalkidis et al., 2020), RoBERTa(Cui et al., 2020), Chinese legal Longformer (Xiao et al., 2021), etc.

2.3 Graph-based Classifications

All the methods above focus on the semantics of documents. However, the document structure, especially for long documents, is also vital for text classification. Recent studies (Kipf and Welling, 2016; Yao et al., 2019; **?**; Liu et al., 2020) propose to represent the documents in a homogeneous or heterogeneous graph. The homogeneous graph consists of all coherent words as nodes (Ragesh et al., 2021; Zhang et al., 2020) – two words are connected if they co-occur. The heterogeneous graph utilizes the entities-sentence co-occurrence (Hu et al., 2019). Given the representative graph, a graph neural network is used to classify the documents (Veličković et al., 2017; Pal et al., 2020; **?**).

To the best of our knowledge, all existing methods classify the documents by topics, reflected by either word frequency or word co-occurrence. None of the existing methods leverage the facts described in the documents. Different from existing methods, we propose to explore the facts in the legal documents for accurate classification.

3 Document-to-Graph Classifier

3.1 Motivation

Referring back to the previous example in Table 1, we observe that a civil complaint case often contains four main components: (1) **Entity information sections** that cover the information of litigation participants; (2) **Facts** between the entities, which are essential for identifying the type of dispute; (3) **Reason** why the plaintiff sued; (4) **Miscellaneous items** that include discussions on the relevant law, procedure, evidence. Miscellaneous items are less relevant to document types, because the same law may be used in different types of disputes. Figure 1 shows the common structures for civil complaints.

The relations among the key participants are often implied in the sentences in the fact and reason sections. These relations vary across different types. Table 2 shows four example sentences relevant to different types of lending disputes. The first three examples belong to "Dispute Over Contract for Private Lending" (DOCPL). Example 1 discusses a lending action, while Example 2 and 3 mention the purpose and reason for the lending action. In Example 4 of a "Dispute Over Contract of Suretyship" (DOCS) document, the "guarantor" is pivotal to identifying the type. However, this keyword may be overlooked by a classifier unless it understands the relations among participants from the sentences.

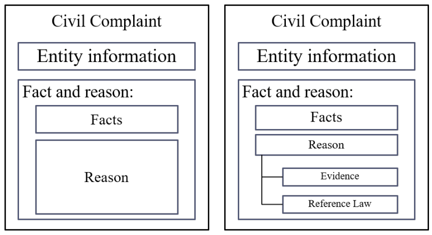

Figure 1: The common structure of Chinese Civil Complaints (Left) and a specific type (Right).

Table 2: Example of different lending disputes.

No.	Example sentence	Class
1	A borrowed B 5,000 dollars.	DOCPL
2	A borrowed B 5,000 dollars to buy C's house.	DOCPL
3	A failed in business and borrowed money from B.	DOCPL
4	A borrowed B's money, C is the guarantor. Because A did not return the money, B ask C to return.	DOCS

Motivated by the observations above, we propose a **Document-to-Graph Classifier (D2GCLF)**. D2GCLF represents a legal document by relation graphs constructed from the facts. Figure 2 shows the architecture of the D2GCLF. Our method extracts four relation graphs, each graph represents the facts associated with the main participants from different aspects (Section 3.2). Once we obtain the graphs, we learn a latent representation of each document by aggregating the information from all graphs, based on the idea of graph attention networks (Section 3.3). The document representation is then passed to a log-softmax layer to generate the class label.

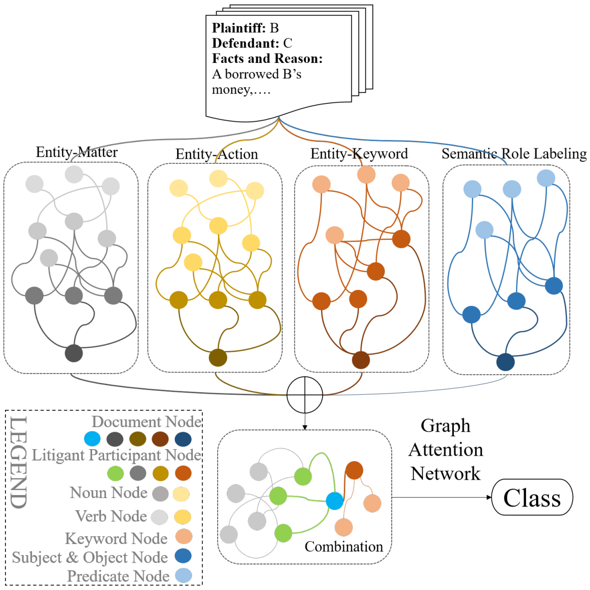

Figure 2: The architecture of D2GCLF.

3.2 Graph Extraction

Entity-Matter Graph The Entity-Matter graph is motivated by the observation that each dispute case should correspond to some matters between the plaintiffs and defendants. For example, the first case in Table 1 discusses the matter of commercial housing transfer and housing ownership. We consider the "matters" as the key evidence for identifying the class of a dispute case. Observing that "matters" are often nouns that appear in the same sentence of the plaintiffs and defendants, we extract the nouns from every sentence that contains both plaintiffs and defendants using a part-of-speech parser[1]. To understand the actions that have been taken on the "matters", we also extract the verb that describes each extracted noun. For example in Table 2, the verb "borrowed" will be extracted together with the noun "dollar". Then, we construct the Entity-Matter graph as follows: (1) create a document node, a plaintiff node (denoted by A) and a defendant node (denoted by B); (2) connect both A and B to the document node; and (3) connect the nouns extracted to both A and B, and to the verbs describing them. Figure 3 shows an example of Entity-Matter graph.

Entity-Action Graph The Entity-Action graph is motivated by the observation that each dispute case should correspond to some actions between the plaintiffs and defendants. For example, the first case in Table 1 indicates that the plaintiff has "paid" the defendant money to purchase the house. The verb "paid" is an action that has been taken by the plaintiff. Motivated by this, we extract the verb from every sentence that contains both plaintiffs and defendants as actions. To understand the actions, we also extract the object of each action. For example in Table 2, the noun "dollar" will be extracted together with the verb "borrowed". Similar to the construction of Entity-Matter graph, we connect the plaintiff and defendant to a document node. The difference is to connect the plaintiff and defendant with each "action". In addition, we attach the corresponding object to each "action" node. Figure 4 shows an example of Entity-Action graph.

[1]https://pypi.org/project/pkuseg/

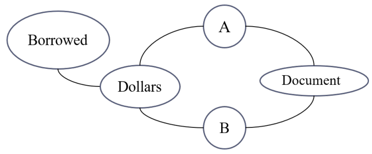

Figure 3: An example of Entity-Matter graph extracted from text: "A borrowed B dollars".

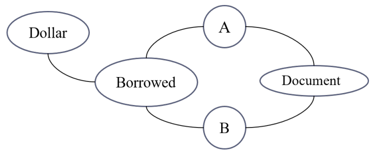

Figure 4: An example of Entity-Action graph extracted from text: "A borrowed B dollars".

Entity-Keyword Graph The Entity-Keyword graph is designed to capture the topics related to the plaintiffs and defendants. For example, the first case in Table 1 includes keywords such as "house", "transfer" and "buy", which describe the general topics of the dispute. We extract keywords from every sentence that contains all litigation participants using Textrank (Mihalcea and Tarau, 2004). For example in Table 2, "dollar" and "borrowed" are the keywords relevant to the key stakeholders. To construct the Entity-Keyword graph, we create the document node and participant nodes in the same way as the previous two graphs. Then, we connect the keywords extracted to the litigation participants. Figure 5 shows an example of the Entity-Keyword graph.

SRL Graph The above graphs only focus on the noun or verb related to the litigation participants. However, besides the main participants, other peo-

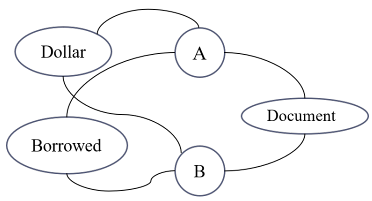

Figure 5: An example of Entity-Keyword graph extracted from text: "A borrowed B dollars".

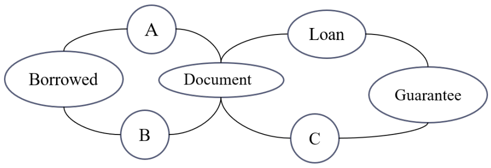

Figure 6: An example of SRL graph extracted from text: "A borrowed B dollars. C provide guarantee for the loan". In the example, "borrowed" is the predicate associated with A and B.

ple and facts may be involved in a law case. For instance, in Example 4 of Table 2, C is a third party individual, who is not the direct participant in the case but acts as a guarantor. Therefore, we construct a Semantic Role Lableing (SRL) graph to extract relations in the form of (subject, predicate, object) from every sentence in the document using the LTP tool[2]. For example in Table 2, (A, borrowed, B) and (C, guarantee, "loan") will be extracted. Then, we construct the SRL graph by: (1) connecting the extracted subjects and objects to the document node; and (2) connecting subjects and objects through the predicates. Figure 6 shows an example of the SRL graph.

Combined graph The four graphs extracted above represent the facts in the documents from different aspects, and thus they have limited presentation power if applied individually. As such, we combine the four graphs by merging the nodes that denote the same concepts from the four graphs into one, e.g., the document nodes, the plaintiff and defendant nodes, and other noun or verb nodes. The edges from the four graphs are preserved in the combined graph. Figure 7 shows an example of the combined graph.

3.3 Graph-based Document Representation Learning

Given the combined graph that represents the facts in a document, we aim to learn a latent representation for the document that encodes information

[2]http://ltp.ai/docs/quickstart.html

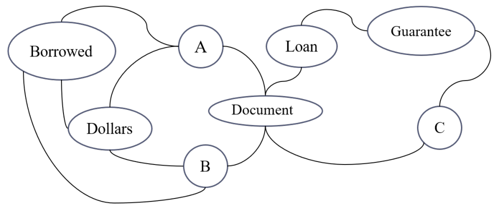

Figure 7: An example of the combined graph.

of the graph. In this work, we apply a graph attention network (Veličković et al., 2017) to aggregate the information from all nodes to the "document" node for learning document representations. Note that, our method is flexible to apply any existing graph neural networks for document representation learning. We will leave the comparison of different graph neural networks as future work.

The input of the graph attention network is a set of node features:

$$h = \{\vec{h}_1, \vec{h}_2, \dots, \vec{h}_N\}, \vec{h}_i \in \mathbb{R}^F, \tag{1}$$

where N is the number of nodes and F is the number of features for each node. We use pre-trained word embeddings (detailed in Section 4.3) as the input feature h for all nodes.

The outputs are the F'-dimension latent representations of all nodes:

$$h' = \{\vec{h'}_1, \vec{h'}_2, \dots, \vec{h'}_N\}, \vec{h'}_i \in \mathbb{R}^{F'}. \tag{2}$$

For each node, the graph attention network performs self-attention and the cross-correlation attention with neighbor nodes:

$$\begin{aligned} a &: \mathbb{R}^{F'} \times \mathbb{R}^{F'} \rightarrow \mathbb{R} \\ e_{ij} &= a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j), \end{aligned} \tag{3}$$

where $\mathbf{W}$ represents the weight, a is a single-layer feedforward neural network, e_{ij} means the importance of node j to node i, vector h is the feature vector and subscripts i, j denote the i-th node and the j-th node, respectively. Then, a softmax function is applied to regularize the attention scores α_{ij} for the adjacent nodes such that they sum to one.

$$\alpha_{ij} = softmax_j(e_{ij}) = \frac{exp(e_{if})}{\sum_{k \in \mathcal{N}_i} exp(e_{ik})}. \tag{4}$$

Then, the latent representation of a node is computed as an aggregation of its neighbors:

$$\vec{h}'_i = \sigma(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W}\vec{h}_j), \tag{5}$$

where the $\sigma(\cdot)$ is an activation function and $\mathbf{W}$ is a learnable weight matrix. In this work, we use the Rectified Linear Unit $RELU(\cdot)$ as the activation function. In order to be able to focus on the information from the scattered nodes to Document node, we apply multi-head attention, which calculate K different attention scores $\alpha_{i,j}^{k}$ and weights $\mathbf{W}^{k}$:

$$\vec{h}_i' = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}_i}\alpha_{i,j}^{k}\mathbf{W}^{k}\vec{h}_j\right). \tag{6}$$

By the above formulation, the "document" node will aggregate information from its neighbors, which in turn, aggregate information from their neighbors.

Let the latent vector representation of the document be $\vec{h}_{doc}'$, which is obtained by Eq. 6. We feed the document representation $\vec{h}_{doc}'$ to a log-softmax layer to compute the probability for a document to be a particular type by:

$$p(class|doc) = log\text{-}softmax(\vec{h}_{doc}'). \tag{7}$$

To learn the parameters of the model, we minimise the Negative Log Likelihood Loss:

$$\mathcal{L} = -\sum_{(d,l_d)\in\mathcal{D}_{train}}\log p(l_d|d), \tag{8}$$

where (d, l_d) is a pair of document d and its class label l_d in the training set $\mathcal{D}_{train}$.

4 Experiments

4.1 Dataset

Chinese civil law contains more than 400 classes. Among those 400 classes, we asked legal experts to pick up 20 classes, which are the most semantically close to each other. Table 5 shows the name and the label of each of the 20 selected classes. We then collected the 4,000 judgments from China Judgments Online[3] by choosing the latest published 200 cases in the searching result for each of the 20 classes. Following the format of the Chinese judgment, we selected the parties' information paragraph and the plaintiff's allegation paragraph from each judgment to create the statement of claim. The resulting dataset contains 4,000 civil cases, 200 cases per class. Ethic issues are discussed in Section 6.

We randomly split the dataset into 70% training set and 30% test set. For the anaphora and co-references in the indictments, we replace them with the actual name of the litigation participants.

4.2 Compared Models

We compare the proposed method with several baselines. As described in Section 2, multiple word representation methods can be used, such as TF-IDF and word embeddings. For word embedding, We use the pre-trained word embedding on People's Daily News[4]. Because, the linguistic style of the newspaper is very similar to the Chinese legal document. The comparison of different word representations, initialization, and TF-IDF, for baselines are presented in Appendix C. For neural network-based model, we keep the same parameter as the original papers and we selected the 10 best checkpoints based on performance on the validation set and report averaged results on the test set.

Machine learning methods. We compare our model with Naïve Bayes, SVM, Logistic Regression, Boosting model, Bagging model in legal document classification. The best embedding result for these models is Char-level TF-IDF representation. We utilize the suitable pre-trained Chinese word embeddings[4].

Deep learning. We compare D2GCLF with deep learning techniques including CNN, RNN, RNN-Bidirectional (BiRNN), BiLSTM, RCNN, HN-ATT (**?**)) using pre-train word embeddings[4] as input.

BERT series. We utilize the pre-trained BERT (Devlin et al., 2018)[5], RoBERTa (Cui et al., 2021)[6], and Chinese Legal Longformer (Xiao et al., 2021)[7] to compare with the proposed model. After encoding the document by the pre-trained model, we add a fully-connected layer to predict the label. Based on the pre-trained language model, BERT, we also select the Task-Scaling mechanisms (TaSc) (Chrysostomou and Aletras,

[3]All indictments used in this paper were collected on China Judgment Online`https://wenshu.court.gov.cn/`. For ethical concern, we only release the Reference Number and url for the cases we used. The index file is available on: `https://drive.google.com/file/d/1bZVv0TPSjIRsRj0OP67v8Y-K-tb-o7IE/view?usp=sharing`.

[4]https://github.com/Embedding/Chinese-Word-Vectors

[5]https://storage.googleapis.com/bert_models/2018_11_03/chinese_L-12_H-768_A-12.zip

[6]https://huggingface.co/hfl/chinese-roberta-wwm-ext-large

[7]https://github.com/thunlp/LegalPLMs

2021)[8] as baseline. TaSc allows us to learn non-contextualised information from category text.

Graph-Based series. We also include graph-based models, such as Graph Convolutional Networks (GCN) (Yao et al., 2019), GrapSAGE (**?**) and Graph Attention Networks (GAT) (Pal et al., 2020), TextING (Zhang et al., 2020). For GCN, GAT, and GrapSAGE following Yao et al. (Yao et al., 2019), we construct a document-word graph from the dataset. Specifically, we create a graph node for each document and each word. Then, a word node connects to a document node if the word is in the document. Word nodes are connected if they are in the same document. Then, we also strictly follow TextING instructions[9].

4.3 Configuration

We utilize the same pre-trained Chinese word embeddings[4] as the baseline methods. For the parameters of D2GCLF, we set the size of node representation $F' = 25$ in Eq. 2 and the number of attention heads $K = 8$ in Eq. 6, and utilize the Adam optimizer with the learning rate $5 * 10^{-5}$, and weight decay $5 * 10^{-4}$. In the experiments, the parameters of all methods are obtained by cross-validation on the training data. To validate the performance of the classifiers, we use AUC as the evaluation metric. Experiments are repeated five times and take the average on a workstation with an Nvidia GeForce RTX 3090 GPU with 24 GB memory.

4.4 Result and Discussion

Overall Comparison: The overall comparison among all methods is shown in Table 3. We observe that D2GCLF outperforms all methods and gains a 4% AUC improvement over the best baseline method. Among the traditional classification methods, the boosting model achieves the best AUC because it ensembles multiple models, each of which focus on different features. Naïve Bayes performs the worst. This is because it heavily relies on the class distribution in the training data, which could be different in the test set.

For deep neural networks, CNN performs the best while RNN performs the worst (70% of AUC). This is because RNN embeds sentences and words that may not be relevant to the class. Conversely, CNN aggregates information in a local context, which is less sensitive to noise. Among the deep learning methods, BERT performs the best, because the transformer can capture the impact from key sentences, paragraphs, or sections to some extent. As we have mentioned, the key factors for determining the class of a legal document often lies in particular sentences, paragraphs, or sections. The graph neural networks with a document-word graph perform the worst, because the graph contains irrelevant words. TextING constructs a word graph by connecting co-coherence words to represent structure information for message passing. However, it does not work well for the legal data because the graph does not capture the key facts.

Besides, deep learning methods such as CNN and RNN are less effective in legal document classification than in other domains. This is because legal documents imply complex relations among the participants, while the size of real-world legal document datasets, especially the indictment documents, is relatively small for learning a complex and deep model from plain texts. D2GCLF outperforms existing methods by representing the facts in a document as a graph.

Table 3: Classification performance

Model Name		AUC
Machine Learning	Naive Bayes	81.93%
	Logistic Regression	85.81%
	SVM	85.15%
	Bagging Model	85.35%
	Boosting Model	87.17%
Deep Learning	CNN	85.75%
	RNN	79.08%
	BiRNN	83.07%
	BiLSTM	84.67%
	RCNN	83.55%
	HN-ATT	85.14%
BERT Series	BERT	85.75%
	TaSc	85.27%
	RoBERTa	81.00%
	Legal Longformer	82.25%
Graph based	GCN	79.08%
	GAT	83.07%
	GraphSAGE	83.55%
	TextING	83.71%
Ours	D2GCLF	**91.33%**

[8]https://github.com/GChrysostomou/tasc
[9]https://github.com/CRIPAC-DIG/TextING

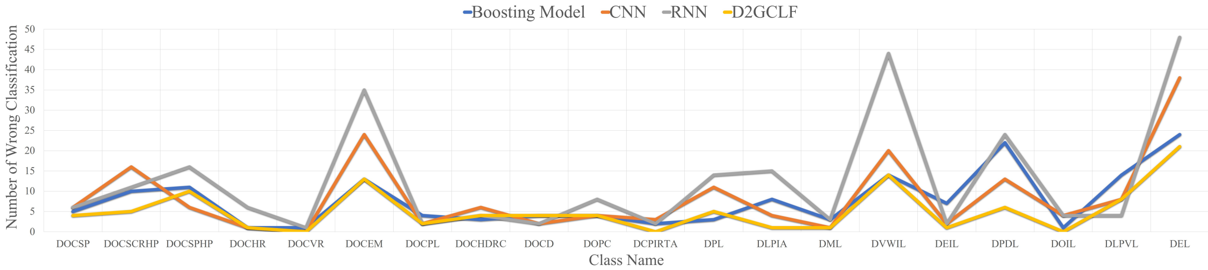

Figure 8: Number of wrong classification cases in different types.

Table 4: Ablation study for the four graphs. ⊖ denotes the variant that removes one of the four graphs from D2GCLF.

Model Name	AUC
$D2GCLF_{Entity-Matter}$	84.83%
$D2GCLF_{Entity-Action}$	83.42%
$D2GCLF_{SRL}$	82.25%
$D2GCLF_{Entity-Keyword}$	82.25%
$D2GCLF_{\ominus Entity-Matter}$	87.08%
$D2GCLF_{\ominus Entity-Action}$	87.67%
$D2GCLF_{\ominus SRL}$	87.75%
$D2GCLF_{\ominus Entity-Keyword}$	88.08%
D2GCLF	**91.33%**

Ablation Study: Next, we test the impact of different relation graphs. We first compare the performance of each graph, denoted by $D2GCLF_{graph}$, where $graph \in$ {Entity-Matter, Entity-Action, SRL, Entity-Keyword}. Then, we compare the full model with its variants by removing each graph, denoted by $D2GCLF_{\ominus graph}$.

Table 4 shows the AUC of all variants. The Entity-Matter graph performs the best among individual graphs. Besides, when the Entity-Matter graph is removed from the full model, the performance decreases the most. The result justifies that matters involving both participants are the most important factors for classifications. The performance drop of $D2GCLF_{\ominus Entity-Action}$ implies the actions of the participants provides additional information for classification. $D2GLF_{\ominus Entity-Keyword}$ results in the least performance drop. This justifies that the topics represented by keywords may have little impact on the task. The significant difference between the full and ablation models shows that the four graphs complement each other.

Case Study: We report the numbers of wrong classification cases for each model in Figure 8. Due to the space limit, we only report the results of Boosting, CNN, RNN and D2GCLF. Overall, D2GCLF generates the least wrong classifications in most classes (e.g., DPDL, DOCD and DOCSP). Specifically, the two classes *Disputes of employer liability (DEL)* and *Dispute of voluntary workers injured liability (DVWIL)* are the most difficult to identify, because they are highly related to other types. The difference between the two is whether work is paid or volunteered. Besides, the injured liability includes traffic accidents, medical liability, etc., which are similar to other dispute classes. D2GCLF performs the best, even for difficult classes because it considers the facts, e.g. traffic accidents.

5 Conclusion

This study explores a novel idea of classifying legal documents based on the facts discussed in the documents. We propose a Document-to-Graph Classifier (D2GCLF) to implement our idea by modeling the facts as four relation graphs, and applying graph attention network to learn the document representation for classification. Experiments on a real-world legal document dataset show the effectiveness of the four relation graphs and the superior performance (91.33% AUC) of D2GCLF.

6 Ethics Statement

The 4,000 Chinese cases that this research is based upon are drawn from the China Judgements Online (`https://wenshu.court.gov.cn/`), which is available for everyone to search cases' judgments once logged in. We notice that previous researches (Tsarapatsanis and Aletras, 2021; Leins et al., 2020) are worried about the ethical concerns raised in terms of applying the NLP technique into the legal domain. Some researchers believe that processed datasets cause people to harm the relevant parties as such datasets enable a more straightforward retrieval process than searching through the original datasets published by government agencies. We would like to point out that the judgments published on China Judgment Online already redacted most of the important personal information. Nevertheless, to mitigate against any remaining concerns, in our dataset, we only publish the case reference number and URL instead of the full text which is available. Besides, we only allow the index dataset to be used in research, not for any other purpose. To download or any other way to use judgments in China Judgment Online, please follow the website's terms and conditions.

References

Stephan Bloehdorn and Andreas Hotho. 2004. Boosting for text classification with semantic features. In *International workshop on knowledge discovery on the web*, pages 149–166. Springer.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904.

George Chrysostomou and Nikolaos Aletras. 2021. Improving the faithfulness of attention-based explanations with task-specific information for text classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 477–488.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 657–668.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021. Pre-training with whole word masking for chinese bert.

Monidipa Das, Mahardhika Pratama, Septiviana Savitri, and Jie Zhang. 2019. MUSE-RNN: A multilayer self-evolving recurrent neural network for data stream classification. In *2019 IEEE International Conference on Data Mining*, pages 110–119.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Shikai Fang, Shandian Zhe, Kuang-chih Lee, Kai Zhang, and Jennifer Neville. 2020. Online bayesian sparse learning with spike and slab priors. In *20th IEEE International Conference on Data Mining, ICDM*, pages 142–151.

Leilei Gan, Kun Kuang, Yi Yang, and Fei Wu. 2021. Judgment prediction via injecting legal knowledge into neural networks. In *Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*, pages 12866–12874.

Alexander Genkin, David D Lewis, and David Madigan. 2005. Sparse logistic regression for text categorization. *DIMACS Working Group on Monitoring Message Streams Project Report*.

Linmei Hu, Tianchi Yang, Chuan Shi, Houye Ji, and Xiaoli Li. 2019. Heterogeneous graph attention networks for semi-supervised short text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 4820–4829.

Georgiana Ifrim, Gökhan Bakir, and Gerhard Weikum. 2008. Fast logistic regression for text categorization with variable-length n-grams. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 354–362.

Rie Johnson and Tong Zhang. 2017. Deep pyramid convolutional neural networks for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 562–570.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kobi Leins, Jey Han Lau, and Timothy Baldwin. 2020. Give me convenience and give her death: Who should decide what uses of NLP are appropriate, and on what basis? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2908–2913.

FengGang Li, JiLi Fan, Li Wang, HuLin Zhang, and Rui Duan. 2011. A method based on manifold learning and bagging for text classification. In *2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce*, pages 2713–2716. IEEE.

Guodong Li, Zhe Wang, and Yinglong Ma. 2019a. Combining domain knowledge extraction with graph long short-term memory for learning classification of chinese legal documents. *IEEE Access*, 7:139616–139627.

Xiaoyu Li, Buyue Qian, Jishang Wei, An Li, Xuan Liu, and Qinghua Zheng. 2019b. Classify eeg and reveal latent graph structure with spatio-temporal graph convolutional neural network. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 389–398. IEEE.

Ronghua Lin, Chengzhou Fu, Chengjie Mao, Jingmin Wei, and Jianguo Li. 2018. Academic news text classification model based on attention mechanism and rcnn. In *CCF Conference on Computer Supported Cooperative Work and Social Computing*, pages 507–516. Springer.

Wan-Chen Lin, Tsung-Ting Kuo, Tung-Jia Chang, Chueh-An Yen, Chao-Ju Chen, and Shou-de Lin. 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)*, pages 140–141.

Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *CoRR*, abs/1703.03130.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*.

Xien Liu, Xinxin You, Xiao Zhang, Ji Wu, and Ping Lv. 2020. Tensor graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8409–8416.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*.

Mariana Y. Noguti, Eduardo Vellasques, and Luiz S. Oliveira. 2020. Legal document classification: An application to law area prediction of petitions to public prosecution service. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Ankit Pal, Muru Selvakumar, and Malaikannan Sankarasubbu. 2020. Magnet: Multi-label text classification using attention-based graph neural network. In *ICAART (2)*, pages 494–505.

Tomas Pranckevičius and Virginijus Marcinkevičius. 2016. Application of logistic regression with part-of-the-speech tagging for multi-class text classification. In *2016 IEEE 4th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*, pages 1–5. IEEE.

Rahul Ragesh, Sundararajan Sellamanickam, Arun Iyer, Ramakrishna Bairi, and Vijay Lingam. 2021. Hetegcn: Heterogeneous graph convolutional networks for text classification. In *The Fourteenth ACM International Conference on Web Search and Data Mining*, pages 860–868.

Edwina L. Rissland, Kevin D. Ashley, and R.P. Loui. 2003. Ai and law: A fruitful synergy. *Artificial Intelligence*, 150(1):1–15.

Constanta Rosca, Bogdan Covrig, Catalina Goanta, Gijs van Dijck, and Gerasimos Spanakis. 2020. Return of the AI: an analysis of legal research on artificial intelligence using topic modeling. In *Proceedings of the Natural Legal Language Processing Workshop 2020*, pages 3–10.

Saket Sathe and Charu C Aggarwal. 2019. Nearest neighbor classifiers versus random forests and support vector machines. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 1300–1305. IEEE.

Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. 2018. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 440–450.

Vu D. Tran, Minh Le Nguyen, and Ken Satoh. 2019. Building legal case retrieval systems with lexical matching and summarization using A pre-trained phrase scoring model. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law (ICAIL)*, pages 275–282.

Dimitrios Tsarapatsanis and Nikolaos Aletras. 2021. On the ethical limits of natural language processing on legal text. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3590–3599.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations (ICLR)*.

Baoxin Wang. 2018. Disconnected recurrent neural networks for text categorization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2311–2320, Melbourne, Australia. Association for Computational Linguistics.

Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. 2018a. Joint embedding of words and labels for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2321–2331. Association for Computational Linguistics.

Shiyao Wang, Minlie Huang, and Zhidong Deng. 2018b. Densely connected cnn with multi-scale feature attention for text classification. In *IJCAI*, pages 4468–4474.

Chaojun Xiao, Xueyu Hu, Zhiyuan Liu, Cunchao Tu, and Maosong Sun. 2021. Lawformer: A pre-trained language model for chinese legal long documents. *AI Open*, 2:79–84.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7370–7377.

Haopeng Zhang and Jiawei Zhang. 2020. Text graph transformer for document classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8322–8327. Association for Computational Linguistics.

Tianyang Zhang, Minlie Huang, and Li Zhao. 2018. Learning structured representation for text classification via reinforcement learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 6053–6060.

Yufeng Zhang, Xueli Yu, Zeyu Cui, Shu Wu, Zhongzhen Wen, and Liang Wang. 2020. Every document owns its structure: Inductive text classification via graph neural networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 334–339.

Zheng Zhang and Cole Hawkins. 2018. Variational bayesian inference for robust streaming tensor factorization and completion. In *IEEE International Conference on Data Mining*, pages 1446–1451.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. How does NLP benefit legal system: A summary of legal artificial intelligence. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5218–5230.

A Example of Legal Documents

Table 6 shows two examples of real cases with similar topics but categories in different types.

B Classes in Dataset

There are twenty classes in our experiment, such as Dispute over contract for sale and purchase, etc. Table 5 shows the full name and abbreviation of these classes.

C Baselines

Except for the baselines, which are introduced in Section 4, we also utilize the other word embedding methods to test the baselines. The full result is shown in Table 7.

Table 5: Classes in dataset

No.	Class
1	Dispute over contract for sale and purchase (DOCSP)
2	Dispute over contract of sale of commercial residential housing property (DOCSCRHP)
3	Dispute over contract for sale and purchase of housing property (DOCSPHP)
4	Disputes over contract for housing rental (DOCHR)
5	Disputes over contract for vehicle rental (DOCVR)
6	Dispute over contract of earnest money (DOCEM)
7	Dispute over contract for private lending (DOCPL)
8	Dispute over contract for housing demolition, relocation and compensation (DOCHDRC)
9	Dispute over contract for decoration (DOCD)
10	Dispute over partnership contract (DOPC)
11	Dispute over compensation for personal injury resulting from traffic accidents (DCPIRTA)
12	Disputes of product liability (DPL)
13	Disputes over liability for personal injury from animals (DLPIA)
14	Dispute of medical liability (DML)
15	Dispute of voluntary workers injured liability (DVWIL)
16	Disputes of educational institutions liability (DEIL)
17	Disputes of property damage liability (DPDL)
18	Dispute of online infringement liability (DOIL)
19	Disputes of labor providers victimization liability (DLPVL)
20	Disputes of employer liability (DEL)

Word representation techniques Most existing methods take word representations as input. As such, we apply different word representation techniques: (1) Word-level TF-IDF that calculates each word's TF-IDF in the document; (2) N-grams TF-IDF that calculates the word frequency on the n-gram basis; (3) Char-level TF-IDF that calculates the frequency of n consecutive characters; (4) Pre-trained word embedding[10] that represents the semantics of each word with a low-dimensional vector.

Machine learning methods We choose the Naive Bayes, SVM, Logistic Regression, Boosting model, Bagging model to classifier the legal documents. Given that these methods will not update the word representation vector, We run these classifers on Word-level, N-grams and Char-level TF-IDF representations.

Deep learning We compare our method to deep learning techniques including CNN, RNN, RNN-Bidirectional (BiRNN), BiLSTM, RCNN. We compare the classification performance of this category of methods with both random initialized and pre-train word embeddings.

[10]https://github.com/Embedding/Chinese-Word-Vectors

Table 6: Example legal documents of two different types

<table>
<tr><th>Class</th><th>Documents</th></tr>
<tr><td>DOCS CRH</td><td>原告: LF
被告: GDYZ co.,ltd
事实及理由: 2016 年 6 月 3 日原告与被告签订合同, 购买位于都匀市，建筑面积 19.36 平方米，购房金额为 367724 元，被告承诺于 2018 年 5 月 15 日前将商品房转移登记有关文件交付给原告，逾期按按照总购房款每日万分之一支付违约金; 合同签订后，原告向被告交付了全部购房款，但被告至今未向原告交付商品房转移登记有关文件。被告迟延办理房屋产权证书存在违约, 损害了原告的利益。故向法院提起诉讼。

(Translated)
Plaintiff: LF
Defendant: GDYZ co.,ltd
Fact and Reason: On June 3, 2016, the plaintiff and the defendant signed a contract to purchase a house located in Duyun City with a construction area of 19.36 square meters and a purchase amount of 367,724 yuan. The defendant promised to deliver the relevant documents of the commercial housing transfer registration to the plaintiff before May 15, 2018, which was overdue Liquidated damages were paid according to one ten thousandth of the total purchase price per day; after the signing of the contract, the plaintiff paid the defendant all the purchase price, but the defendant has not yet delivered to the plaintiff the relevant documents for the transfer of commercial housing. The defendant’s delay in handling the housing ownership certificate breached the contract, which harmed the interests of the plaintiff. Therefore, a lawsuit was filed in the court.</td></tr>
<tr><td>DOCS PHP</td><td>原告: ZB
被告: YJ
被告: WS
事实及理由: 二被告是亲戚关系，WS 曾是 YJ 的岳母。2015 年 11 月 24 日，原、被告达成购房合意。2015 年 11 月 27 日，原、被告签订房屋买卖合同。2015 年 12 月 3 日，原告交付完购房款，一直居住至今。2019 年初，原告催促被告办理房屋过户手续未果，故诉至法院。

(Translated)
Plaintiff: ZB
Defendant: YJ
Defendant: WS
Fact and Reason: The two defendants are relatives, and WS was YJ’s mother-in-law. On November 24, 2015, the plaintiff and the defendant reached an agreement to purchase a house. On November 27, 2015, the plaintiff and the defendant signed a house purchase contract. On December 3, 2015, the plaintiff paid the purchase price and has been living until now. At the beginning of 2019, the plaintiff failed to urge the defendant to go through the house transfer procedures, so he sued to the court.</td></tr>
</table>

Table 7: Classification performance (AUC) of all methods

Model Name	**Result**		
	WordLevel TF-IDF	***N-Gram Vectors***	***CharLevel Vectors***
Naive Bayes	45.75%	25.38%	81.93%
Logistic Regression	43.76%	26.18%	85.81%
SVM	38.78%	24.63%	85.15%
Bagging Model	39.74%	26.13%	85.35%
Boosting Model	35.41%	13.62%	87.17%
	Random Initialization		**Word Embedding**
CNN	72.72%		85.75%
RNN	39.47%		79.08%
RNN-Bidirectional	54.45%		83.07%
BiLSTM	58.58%		84.67%
RCNN	72.75%		83.55%
HN-ATT	-		85.14%
BERT	85.75%		
TaSc	85.27%		
RoBERTa	81.00%		
Legal Longformer	82.25%		
GCN	79.08%		
GAT	83.07%		
GraphSAGE	83.55%		
TextING	83.71%		
D2GCLF	**91.33%**		