Information Processing and Management 59 (2022) 102798

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

# A comparative study of automated legal text classification using random forests and deep learning☆

Haihua Chen [a], Lei Wu [b], Jiangping Chen [a], Wei Lu [c], Junhua Ding [a,*]

[a] Department of Information Science, University of North Texas, Denton, TX, 76203, USA
[b] Faculty of Education, Shandong Normal University, Jinan, Shandong, 250014, China
[c] School of Information Management, Wuhan University, Wuhan, Hubei, 430072, China

ARTICLE INFO

*Keywords:*
Legal text classification
Machine learning
Deep learning
Domain concept
Word embedding
Random forests

ABSTRACT

Automated legal text classification is a prominent research topic in the legal field. It lays the foundation for building an intelligent legal system. Current literature focuses on international legal texts, such as Chinese cases, European cases, and Australian cases. Little attention is paid to text classification for U.S. legal texts. Deep learning has been applied to improving text classification performance. Its effectiveness needs further exploration in domains such as the legal field. This paper investigates legal text classification with a large collection of labeled U.S. case documents through comparing the effectiveness of different text classification techniques. We propose a machine learning algorithm using domain concepts as features and random forests as the classifier. Our experiment results on 30,000 full U.S. case documents in 50 categories demonstrated that our approach significantly outperforms a deep learning system built on multiple pre-trained word embeddings and deep neural networks. In addition, applying only the top 400 domain concepts as features for building the random forests could achieve the best performance. This study provides a reference to select machine learning techniques for building high-performance text classification systems in the legal domain or other fields.

## 1. Introduction

Legal text classification aims to identify the category of a legal text based on the association between the legal text and that category (Boella et al., 2011). It is the foundation of building intelligent legal systems which become important tools for lawyers due to the exponentially increasing amount of legal documents and the difficulties in finding rulings in previous similar cases for argumentation.[1] Legal text classification has been investigated with different natural language processing strategies (Octavia-Maria et al., 2017). For example, Palau and Moens (2009) identified argumentative propositions, argumentative function, and argumentative structure from the European Court of Human Rights (ECHR) legal texts.[2] Boella et al. (2011) classified an Italian

☆ This research is partially supported by United States NSF grant #1852249, NSA grant #H98230-20-1-0417, and grant #KFKT2019A19 from the State Key Laboratory for Novel Software Technology in Nanjing University, China.

* Corresponding author.
*E-mail addresses:* haihua.chen@unt.edu (H. Chen), ccnustone@yeah.net (L. Wu), jiangping.chen@unt.edu (J. Chen), weilu@whu.edu.cn (W. Lu), junhua.ding@unt.edu (J. Ding).

[1] Also called "legal judgment prediction", aims at automatically predicting the outcome of a court case, given a text describing the case's facts (Chalkidis, Androutsopoulos et al., 2019). It is typically formalized as a text classification task.

[2] Also called "argument identification" or "argument mining", aims at automatically identifying and extracting the structure of inference and reasoning expressed as arguments presented in legal texts (Moens et al., 2007). Text classification is usually used to classify an argument or set of arguments according to its argument type.

https://doi.org/10.1016/j.ipm.2021.102798
Received 1 March 2021; Received in revised form 10 July 2021; Accepted 17 October 2021
Available online 17 November 2021

legal text into a relevant domain. Aletras et al. (2016) predicted the ruling, law area, and the ruling issued date of an ECHR legal document.[3] Octavia-Maria et al. (2017), Şulea et al. (2017) applied machine learning techniques to predict the ruling of the French Supreme Court and the law area to which a case belongs. Nguyen et al. (2018) used recurrent neural network-based models for recognizing requisite and effectuation parts in Japanese legal texts. Chalkidis, Androutsopoulos et al. (2019) focused on the legal judgment prediction task on ECHR cases. Ji, Tao, et al. (2020) incorporated the legal classification task into the information extraction task as a multi-task learning problem for evidence extraction from Chinese court documents.

Most of the existing research focuses on international legal texts such as Chinese cases, European cases, and Australian cases. Few studies explored text classification for U.S. legal texts. Since a classification model that works well in one legal language may not achieve similar performance in other legal languages, more studies are needed in this area to test the applicability of proposed methods. Furthermore, existing legal text classification research focuses more on short texts with small label sets (binary or around five labels). They may not be as effective for long texts and large-scale label text classification in the legal domain. Recently, many studies assume a complex model, such as deep learning, would produce a state-of-the-art (SOTA) system for text classification. Our preliminary experimental results on legal text classification using TF-IDF (Salton & Buckley, 1988) features and Random Forests (RFs) algorithms outperformed language model BERT (Devlin et al., 2019) with deep neural network LSTM (Sundermeyer et al., 2012), the SOTA approach in many NLP tasks.

However, processing legal texts is more challenging compared to other domains due to the complexity of legal sources and legal language (Ma et al., 2020; Nazarenko & Wyner, 2017). Such challenges are mainly reflected in the following aspects: First, the lexicon in the legal domain is rich. It includes unique legal expressions (e.g. charged or defendant) some of which may have internal structure (e.g. without prejudice to any claim), which are mostly standardized or codified in dictionaries of legal language (Nazarenko & Wyner, 2017). Second, there are legal sub-areas, e.g. family, criminal, and contract law, each of which codify different terminologies. Both the legally-specific lexical items within and across legal sub-areas are important for the processing of legal documents (Pudaruth et al., 2018). Third, terminology in the legal texts is often mixed with the vocabularies of the field (domain terminology) or ordinary vocabularies (domain-independent terminology). For instance, a legal text about cyberlaw will discuss how legal concepts, e.g. obligations or rights, apply to specific domain terminology for aspects of computers and communication technologies, e.g. WiFi or passwords, that are neither legally defined, nor legal concepts (Nazarenko & Wyner, 2017). In addition, considering the variation in time, jurisdictions, and languages, the lexical complexities proliferate. Traditional features such as TF-IDF is not capable for building a high-quality legal text classification systems. Domain concept, key concepts extracted within a target domain (Šajatović et al., 2019), might be effective features for legal text classification.

Specifically, this paper targets legal text classification of large-scale case documents from the United States supreme court. The research objectives of this paper are as follows. (1) To develop an effective and efficient machine learning model for legal text classification with large-scale, long U.S. case documents. (2) To compare traditional feature-based machine learning with deep learning-based approaches for selecting an appropriate strategy that fits legal text classification.

To achieve these purposes, we build two types of legal text classification systems on a dataset with 30,000 full U.S. case documents in 50 categories: one is a domain concept-based RFs classifier, and the other is the pre-trained word embeddings-based deep learning classifier. For the former, the domain concepts are extracted using the method mentioned in Meijer et al. (2014), and they are further filtered using principal component analysis (PCA) (Wold et al., 1987). The selected domain concepts then serve as the features for building the RFs classifier. For the deep learning-based text classification, texts are embedded using a pre-trained language model such as Word2vec, GloVe, or BERT, and CNN or RNN is the learning model. The experimental results show that the random forests classifier based on domain concepts significantly outperforms the deep learning-based one regarding the accuracy, recall, precision, and F1 score in different experimental settings. In addition, only the top 400 domain concepts chosen as the features for building the random forests can achieve the best performance of legal text classification. Finally, we discuss the model selection strategies for building domain-specific machine learning systems by taking legal text classification as an example. Our code and dataset are available in GitHub.[4]

In summary, the contributions of our paper are summarized as follows:

- We apply domain concepts to legal text classification based on PCA and RFs to demonstrate its powerful ability for legal text. The domain concept model can extract effective features from legal texts.
- We conduct a systematic comparative study by using domain concept-based machine learning algorithms and pre-trained word embeddings-based deep learning algorithms, respectively. The experiments on a subset of the SigmaLaw dataset with approximately 30,000 full legal case documents in 50 categories demonstrate the efficacy of the domain concept-based RFs model.
- The third contribution of this research is a framework of model selection for text classification. Based on previous experimental results and literature, the framework includes the strategy for selecting machine learning models in terms of four indicators: data, performance, computation, and interpretation.

[3] Also called "court ruling prediction", aims to identify which laws apply to a case and what the ruling might be based on the case description (Octavia-Maria et al., 2017). Existing court rulings are usually used to train a classifier to perform the prediction.

[4] https://github.com/unt-iialab/Legal-text-classification

## 2. Related work

Literature related to this study are reviewed regarding four aspects: (1) legal text classification, (2) word embeddings for text classification, (3) domain concept extraction (DCE) and its applications, and (4) the selection and comparison of machine learning models for NLP tasks.

### 2.1. Legal text classification

Legal text classification is to identify the category of a legal text based on the association between the legal text and that category (Boella et al., 2011). The specific tasks for legal text classification include: law area classification (Aletras et al., 2016; Boella et al., 2011), ruling identification (Aletras et al., 2016), argument mining (Palau & Moens, 2009), and court decision prediction (Octavia-Maria et al., 2017). Many studies have been conducted on legal classification. For example, Palau and Moens (2009) identified argumentative propositions, argumentative function, and argumentative structure in ECHR legal text. Boella et al. (2011) classified an Italian legal text into a relevant domain. Aletras et al. (2016) predicted the ruling, law area, and the ruling issued date of an ECHR legal document. Octavia-Maria et al. (2017) and Şulea et al. (2017) applied machine learning techniques to predict the ruling of the French Supreme Court and the law area to which a case belongs. Ji, Tao, et al. (2020) incorporated the legal classification task to the information extraction task as a multi-task learning problem for evidence extraction from Chinese court documents. Later, they applied the same legal texts (Ji, Tao, et al., 2020) for speakers coreference resolution (Ji, Gao, et al., 2020).

Compared to the general texts such as texts in social media and online newspapers, the legal texts are usually much longer and have a more complex structure, making the classification of legal text challenging (Boella et al., 2011). Boella et al. (2011) applied a TF-IDF features-based SVM model for legal area classification and achieved 76% on the F1-measure. However, the dataset only contained six categories with 32, 75, 3, 25, 83, and 5 documents in each category. A KNN-based legal text classification that used the strategy of splitting the legal documents into equal-sized segments with plurality voting achieved 83.5% accuracy (Pudaruth et al., 2018). The dataset contained 18 categories with 401 legal judgments. Both studies above suffer two issues: the training data is insufficient, and the documents under each category are imbalanced. It might not be sufficient to train a high accurate classifier. Palau and Moens (2009) reported 80% on accuracy regarding argument or non-argument classification of a legal sentence. Evidences have shown binary legal text classification is not as challenging as multiple category text classification.

Recently, deep neural network models have been attracting increasing attention from the legal domain. For example, Nguyen et al. (2018) proposed several recurrent neural network-based models for recognizing requisite and effectuation parts in Japanese legal texts, they also argued that using external features and in-domain pre-trained word embeddings can improve the performance of legal NLP tasks. Leitner et al. (2019) compared the BiLSTMs and CRFs approaches on named entity recognition in German legal documents, showing that the BiLSTMs slightly outperform the CRFs approach. Li et al. (2019) proposed a multichannel attentive neural network model for legal judgment prediction on Chinese criminal cases. Chalkidis, Fergadiotis et al. (2019) compared several neural classifiers for extreme multi-label text classification with 57k legal documents from the EUR-Lex database and 4271 labels in total. The experimental results showed that BiGRU with label-wise attention networks achieved the best performance (69.8%) regarding the F1-measure. For the same task and on the same dataset as Chalkidis, Fergadiotis et al. (2019) and Shaheen et al. (2020) conducted a competitive study using transformer models including BERT, RoBERTa, DistilBERT, and XLNet, finding that RoBERTa achieves the SOTA performance of 75.8% regarding the F1-measure. Soh et al. (2019) compared different deep learning models for legal area classification on the Singapore Supreme Court judgments. However, the best classifier built on BERT-Large could only achieve 60.7% on the F1-measure. In summary, existing research mainly focuses on international legal texts such as Chinese cases, European cases, and Australian cases. Little attention is paid to text classification for U.S. legal texts even though the legal language in different countries is quite different. It is unclear which text classification strategy is most appropriate for U.S. legal documents. Furthermore, the performance of existing approaches is about 70% even with very few categories (around five categories). There is still room for significant improvement.

### 2.2. Word embedding for text classification

Word embedding represents words in a low-dimensional continuous space based on the learning from a large amount of texts (Ethayarajh, 2019). Word embedding can capture the semantic information of text, which is crucial for text representation. Many studies have implemented the word embedding technique to text classification task (Jain et al., 2019; Reimers et al., 2019). By comparing contextual word embeddings with static word embeddings for text classification and clustering, Reimers et al. (2019) showed that BERT is more powerful than other word embeddings such as Word2vec, GloVe, and ELMo. The BERT-Large model, especially – a version of BERT models which is trained with 24-layer, 1024-hidden, 16-heads, 340M parameters – can achieve almost the same as human performance (Reimers et al., 2019). However, in the disaster tweet classification task, the conclusion is the opposite: Word2vec and GloVe both display stronger results (Jain et al., 2019). The reason could be that contextual word embeddings such as ELMo and BERT need to be fine-tuned on a specific dataset to get better results instead of directly using pre-trained embeddings. However, BERT can achieve a better performance in general as soon as the dataset is large enough (Jin et al., 2020), which encourages us to explore the potential of using BERT on legal text classification.

Unlike the text classification tasks mentioned above, there are three main challenges for legal text classification. (1) Legal cases are usually long texts that may consist of thousands of words, which might relate to only one area of law or several areas of

law (Pudaruth et al., 2018). (2) Legal cases in different categories may contain a high percentage of general description information, thereby not having enough unique information for identifying between categories (Pudaruth et al., 2018). (3) The lexicon in the legal domain is rich, it is difficult for the pre-trained language models to learn the word representations of the legal terminologies. (4) The number of labels for the cases in this research is as large as 50, which may cause data sparsity issues for text classification (Liu et al., 2017). Results on long text classification (Adhikari et al., 2019) and extreme multi-label text classification (Liu et al., 2017) have been reported. Although the typical BERT input is usually not the full document, it still outperforms other representations with learning algorithms such as learning to rank, SVM, CNN, and HAN. Meanwhile, deep learning, such as CNN and RNN, has been demonstrated effectively in multi-label text classification (Chen, Ye, et al., 2017). It would be interesting to investigate whether word embeddings, together with deep learning algorithms, could improve legal text classification.

### 2.3. Domain concept extraction and its applications

Different approaches have been developed to extract terms from domain-specific corpora (Kang et al., 2014; Matas, 2017; Meijer et al., 2014; Šajatović et al., 2019). Domain concept extraction (DCE) starts with extracting and filtering candidate terms, followed by scoring and ranking candidate terms (Šajatović et al., 2019). For example, the DCE tool CFinder first extracts key concept candidates based on linguistic patterns and then calculates the importance of the candidates within the target domain. Statistical and domain-specific knowledge is considered to select the final concepts (Kang et al., 2014). Another approach on automatic taxonomy construction from texts (ATCT) first extracts nouns as terms from text documents. Four measures – including domain pertinence, lexical cohesion, domain consensus, and structural relevance – are then combined to filter candidate terms (Meijer et al., 2014). Compared to other methods, the ATCT framework for domain concept extraction not only performs well but also demonstrates its robustness in different domains (Meijer et al., 2014). Therefore, we use this approach to extract domain concepts from our legal corpus. The detailed process will be presented in Section 3.1.

### 2.4. Model selection and comparison

Numerous models have been proposed in the past few decades for text classification (Altınel & Ganiz, 2018; Kowsari et al., 2019; Li et al., 2020). These models can be divided into shallow models and deep learning models, according to how features are extracted and applied (Li et al., 2020). Shallow models emphasize feature extraction and selection based on domain knowledge, as well as classifier design. In contrast, deep learning modeling can automatically perform feature extraction and learn well without domain knowledge (Li et al., 2020). General SOTA studies declare that complex deep learning models such as BERT are more fit for text classification (Li et al., 2020). However, our experiments on legal text demonstrate that complex deep learning models do not fit all text classification tasks. Compared to machine learning models such as RFs, deep learning models require a large amount of data and expensive computational resources (Kowsari et al., 2019). To select an appropriate model for text classification, we should consider data size, data characteristics, the complexity of building a model, and computational resources.

Studies have been conducted on text classification to get the best classifier by using different models. These models are compared on different text classification tasks or the same tasks with different datasets. For example, to test the robustness of BERT, three models including BERT, word-based CNN, and word-based LSTM are compared on text classification using five datasets and textual entailment using two datasets, respectively (Jin et al., 2020). In the litigation code classification task, the BERT fine-tuned model is compared with the TF-IDF-based logistic regression model, TF-IDF-based XGBoost model, and chronology-enhanced XGBoost model on a dataset with 51,948 examples with 40 labels. Although many comparison studies have been conducted, it has remained too confused to select a machine learning model for domain-specific text classification.

The above discussions show that, although different techniques have been explored for legal text classification, there is still much room to improve. In this paper, we investigate the application of domain concepts as features with a RFs classifier and compare the effectiveness of using multiple pre-trained word embeddings with deep learning algorithms. Based on the experimental results and analysis, we further develop a framework that can guide the classification model selection for domain-specific text classification.

## 3. Methodology and experiment design

This comparative study aims to develop an effective and efficient machine learning model for legal text classification with large-scale, long U.S. case documents. We built two types of legal text classification systems and conducted evaluations of the two systems. The research steps include feature extraction, feature selection, document representation, and the construction of machine learning and deep learning systems for classification. For machine learning-based classification systems, we use TF-IDF and domain concept for feature extraction and PCA for feature selection. Random forests is implemented as the classification algorithm. As for deep learning-based classification, we use Word2vec, GloVe, and BERT for document representation, while TextCNN, BiLSTM, and BiLSTM with Attention are used for the classifier learning. In the following sections, we will first describe techniques used in each step, then present the overall experimental design of the study.

### 3.1. Domain concept extraction

Domain concepts extraction includes three steps: 1. term extraction, 2. term filtering, and 3. term weight calculation. The whole process is shown in Fig. 1.

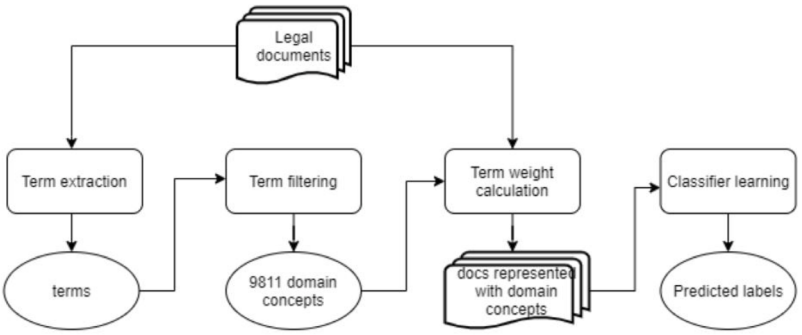


**Fig. 1.** An illustration of domain concept extraction pipeline.

### 3.1.1. Term extraction

Term extraction aims to extract nouns in the legal text corpus since nouns are usually used for labeling concepts (Meijer et al., 2014). To acquire nouns from texts, we use the part-of-speech (POS) tagger embedded in NLTK, which can achieve 94.0% on accuracy. The input of NLTK POS tagger is a sequence of words, and the output is a POS tag attached to each word. After this process, we keep all the noun terms with a length longer than one (Kurfalı & Östling, 2019) and the frequency larger than 50 as the candidate domain concepts.

### 3.1.2. Term filtering

Term filtering is used to select the most relevant terms for a specific domain from the terms extracted. We apply the term filtering approach proposed in Meijer et al. (2014), which uses domain pertinence (DP) and domain consensus (DC) (Sclano & Velardi, 2007) to determine the domain relevance of a term. DP aims to measure the representativeness of a term for a particular domain corpus. The more frequently a term appears in the domain corpus and the less frequently it appears in a contrastive corpus, the higher is the domain pertinence value (Meijer et al., 2014). DP is calculated as follows:

$$DP_{D_i}(t) = \frac{freq(t/D_i)}{max_j(t/D_j)} \tag{1}$$

where $freq(t/D_i)$ denotes the count of term $t$ in domain corpus $D_i$ and $D_j$ represents the count in a contrastive corpus.

DC aims to judge whether a term frequently appears across the domain corpus documents. It is defined as follows:

$$DC_{D_i}(t) = -\sum_{d_k \in D_i} n_{freq}(t/d_k) \cdot \log(n_{freq}(t/d_k)) \tag{2}$$

where $n_{freq}(t/d_k)$ is the normalized frequency of term $t$ in document $d_k \in D_i$.

Finally, DP and DC are combined by the following Eq. (3) to generate the domain score of a term $t$ that appears in the domain corpus $D_i$:

$$Score(t, D_i) = \alpha \frac{DP_{D_i}(t)}{max_t(DP_{D_i}(t))} + \beta \frac{DC_{D_i}(t)}{max_t(DC_{D_i}(t))} \tag{3}$$

where $\alpha$ and $\beta$ are the weights of $DP_{D_i}(t)$ and $DC_{D_i}(t)$, respectively, while $max_t(DP_{D_i}(t))$ and $max_t(DC_{D_i}(t))$ are the highest domain pertinence value and the highest domain consensus value found in the domain corpus $D_i$, respectively.

The term filtering strategy was evaluated on the taxonomy extraction task in the fields of economics and medicine. The taxonomic F-measure, indicating the quality of the concept broader–narrower relations compared to the benchmark, was 68.16% and 65.16% in the two fields, respectively. It demonstrated the effectiveness of this method (Meijer et al., 2014).

### 3.1.3. Term weight calculation

Each domain concept candidate has a distinct role in representing specific information in the legal domain. Therefore, each domain file can be represented as a collection of the concept terms $t_1, t_2, \ldots, t_n$. However, the frequency of domain concepts cannot always keep their potential relationships due to the different locations of domain concepts in the same domain files. To solve this problem, we combine the frequency and the domain score as the term weight, which can be calculated with the following equation:

$$Weight(t, d_j) = freq_{D_i}(t/d_j) \cdot Score(t, D_i) \tag{4}$$

Each legal domain category has its domain concepts. For example, the legal domain categories include *Health Law, Insurance Law, Immigration Law, and Family Law*. The weight of a domain concept indicates its distribution in the category. To the best of our knowledge, few works have been reported on text classification using domain concepts, especially in the legal domain.

### 3.2. Feature selection with PCA

To identify the most effective feature combination, the feature ranking PCA (Wold et al., 1987) is applied to select features for the Random Forests. We follow the strategy described in Song et al. (2010) to extract the principal domain concepts from the initial domain concepts. There are four steps:

1. Co-variance matrix computation: To identify the correlations between different domain concepts.
2. Eigenvectors and Eigenvalues computation based on the co-variance matrix: To identify the principal domain concepts by ordering the eigenvectors by their eigenvalues in descending order.
3. Feature vector construction: To choose whether to keep all these domain concepts or discard those of lesser significance and use the remaining domain concepts as the feature vector.
4. Recast the legal texts along the principal domain concepts: Represent the legal texts with the principal domain concepts.

### 3.3. Word embeddings for document representation

Word2vec (Mikolov, Chen, et al., 2013; Mikolov, Sutskever, et al., 2013) and GloVe (Pennington et al., 2014) are two widely used word embedding models (Rezaeinia et al., 2017). BERT (Devlin et al., 2019) is one of the latest word embedding models that have produced the best performance in many NLP tasks. Therefore, we experiment with all three embedding models to compare their performance. Firstly, a word embedding model is used to convert each word in a legal document into a vector. Then the sequences of word vectors representing the sentences in the document are produced as input sequences to a deep learning model for the text classification.

Word2vec contains two learning models, continuous bag-of-words (CBOW) and continuous skip-gram (Skip). Both are trained on the Google News corpus with considering negative sampling and without considering negative sampling, respectively (Mikolov, Chen, et al., 2013). In this study, we use 300-dimension Word2vec embeddings and set the input sequence length as eight according to the average length of the sentences in the experimental corpus, whose statistics are shown in Table 1. We skip all words with total frequency in the corpus lower than five, as the default setting in Gensim's implementation. Since each legal document in our data collection contains many sentences and paragraphs, we calculated each sentence's average vector to present the whole document. We iterate through our entire corpus with a window size of 16 and generate an average vector for each document.

Unlike Word2vec, which learns word representations from local context windows, GloVe takes the advantages of the matrix factorization methods that can exploit global statistical information (Pennington et al., 2014). We use 300-dimension GloVe word embeddings, and each word in the corpus is mapped to a 300-dimensional vector using GloVe.

Compared to Word2vec or GloVe embeddings, which only produces a fixed context-independent representation for each word, BERT embedding produces the word-level representation based on the information of the entire sentence (Li et al., 2019). Therefore, the same word could have a different representation if the word appears in different sentences. BERT was trained on the BookCorpus dataset and text passages from Wikipedia in English (Devlin et al., 2019). Two pre-trained BERT models with different sizes are available: BERT-Base with 12-layer, 768-hidden, 12-heads, 110M parameters and BERT-Large with 24-layer, 1024-hidden, 16-heads, 340M parameters. In this paper, we use the BERT-Base-Uncased,[5] which is a type of BERT-Base model.

### 3.4. Classification algorithms

In this section, we introduce the four classification algorithms: random forests (Breiman, 2001), TextCNN (Kim, 2014), BiLSTM (Graves et al., 2013), and BiLSTM with Attention (Liu & Guo, 2019).

#### 3.4.1. Random forests

According to a comparison study on 179 classifiers with 121 datasets, RFs achieved the best performance (Fernández-Delgado et al., 2014). RFs is also a high performer in text classification since it mitigates the inherent challenges involved in textual data such as high dimensionality, sparsity, and noisy feature space (Islam et al., 2019). The workflow of the RFs algorithm used in this study includes three steps. (1) For the vast number of features in text data, many trees trained on the random subset of features are required for RFs. In this paper, we use the bagging algorithm to create random samples. Given a dataset 1 (80% legal documents and 9811 features), it creates a new dataset 2 by sampling all the documents randomly with replacement from the original data. About 1/3 of the documents from dataset 1 are left out, known as Out of Bag (OOB) samples. (2) The model was trained on dataset 2. The OOB sample is used to determine an unbiased estimate of the error. (3) Several trees are grown, and the final prediction is obtained by voting.

[5] https://storage.googleapis.com/bert_models/2018_10_18/uncased_L-12_H-768_A-12.zip

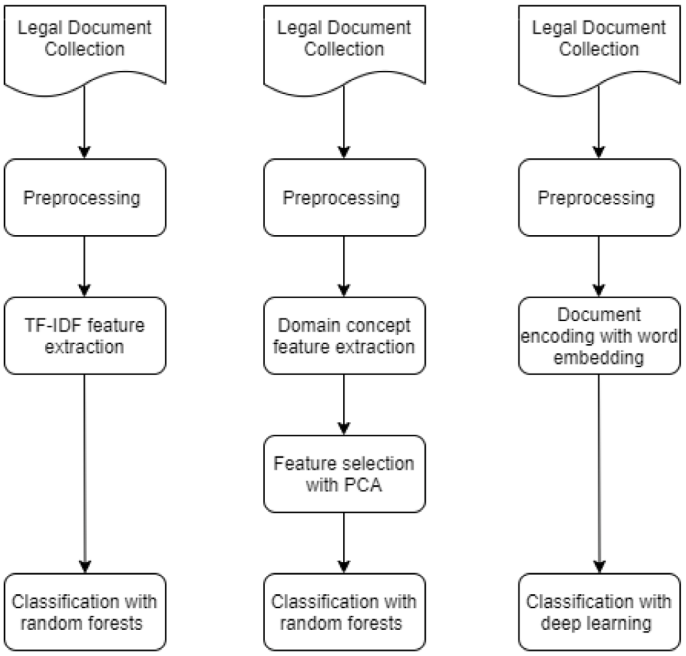


**Fig. 2.** Illustration of the experimental design. Left: TF-IDF with random forests as classifier. Middle: Domain concept with random forests as classifier. Right: Word2vec, GloVe, and BERT for document representation with TextCNN, BiLSTM, and BiLSTM with Attention algorithms.

#### *3.4.2. TextCNN*

TextCNN is a convolutional neural network (CNN) for text. In TextCNN, document matrices generated by word embeddings are input into a CNN to perform classification. In the convolutional layer, every filter performs convolution on each word and different feature maps are firstly generated. A max-overtime pooling operation is then used to select the most critical features, and the activation function is used in this step. Finally, the extracted features are concatenated as the penultimate layer and passed to a fully connected SoftMax layer to predict the probability distribution over class labels (Guo et al., 2019). To avoid over-fitting, we use dropout and batch normalization on the penultimate layer and weight vectors to reduce the parameters, as recommended by Gong and Ji (2018).

#### *3.4.3. BiLSTM*

Different from TextCNN, Long Short-Term Memory (LSTM) is a recurrent neural network (RNN) architecture with long short-term memory units as hidden units (Chen, Xu, et al., 2017). To make the legal text contain the context information, BiLSTM incorporates a forward LSTM layer and a backward LSTM layer to learn information from both preceding and following tokens (Chen, Xu, et al., 2017). Firstly, the word vectors generated by Word2vec, GloVe, or BERT are used as the inputs of the BiLSTM model. The BiLSTM layer keeps the sequential order between the data. It allows detecting the links between the previous inputs and the output. Then, the outputs of the BiLSTM model, which concatenate the forward and backward context representations, are used as representations of the legal text. Finally, the text vectors are input into the neural network classifier to predict the category of the legal text.

#### *3.4.4. BiLSTM with attention*

Even BiLSTM can capture more comprehensive and rich semantics by learning the feature vector from both the forward and backward directions, though not all words in legal texts contribute equally to the representation of the document. For example, in the legal text "The Court expresses no view about whether equitable considerations of laches and acquiescence may curtail the Tribe's power to tax the retailers of Pender", words such as "laches", "Tribe", "tax", and "Pender" are more important than other words in representing the meaning of this sentence, meaning that more "attention" should be given for these words. Therefore, in this paper, we add an attention mechanism to the network model to learn the weight for each word, making important features more prominent. Attention-based BiLSTM models have recently shown better results in text classification tasks (Vaswani et al., 2017).

### *3.5. Experiment design*

As shown in Fig. 2, the proposed method for legal text classification includes three steps. (1) Data preprocessing: in this stage, we perform lemmatization, convert all the words into lower case, and remove stop words using NLTK.[6] (2) Document representation: we

[6] https://www.nltk.org/

**Table 1**
Descriptive statistics of the dataset. Categories are listed in descending order by number of documents.

| Category id | Category name | Doc count | Sentences avg | Words avg |
|---|---|---|---|---|
| 42 | Immigration law | 1292 | 135 | 2743 |
| 20 | Habeas corpus | 1129 | 220 | 4327 |
| 45 | Insurance law | 1085 | 154 | 3075 |
| 41 | Health law | 1020 | 171 | 3314 |
| 36 | Family law | 889 | 160 | 3275 |
| 30 | Environmental law | 881 | 235 | 4589 |
| 16 | Contracts | 826 | 156 | 3224 |
| ... | ... | ... | ... | ... |
| 51 | Trademark | 225 | 183 | 3787 |
| 62 | Military law | 218 | 188 | 3588 |
| 29 | Entertainment law | 213 | 224 | 4434 |
| 66 | Professional malpractice | 207 | 166 | 3394 |

represent documents with TF-IDF or domain concepts or pre-trained word embeddings for machine learning or other deep learning models, respectively. (3) Text classification: we experiment with four classifiers – three deep learning models: TextCNN, BiLSTM, and BiLSTM with Attention – and a traditional machine learning model, random forests.

## 4. Empirical evaluation

To test the effectiveness and robustness of the proposed domain concept-based machine learning framework on legal text classification, we conduct experiments on the public dataset SigmaLaw (Sugathadasa et al., 2017). This section presents the dataset, the experiment, the evaluation process, the experiment results, and the discussion.

### *4.1. The dataset*

We use dataset SigmaLaw[7] published by Sugathadasa et al. (2017) for legal text classification. This dataset contains over 35,000 legal case documents collected from online repositories by web crawling, covering 78 areas of law practice. To ensure enough training data for the machine learning models, we select 50 areas whose documents are larger than 200. The descriptive statistics, including total documents (Doc count), average sentences for each document (Sentences avg), and average words for each document (Words avg) of part of the dataset, is presented in Table 1.

To ensure the quality of the dataset, we check how many duplicated documents exist in the dataset from three aspects:

- Check the data collection process: As mentioned previously, the legal case documents were collected from a single source (SigmaLaw) by web crawling. It is unlikely that much duplication was included.
- Check the similarity between documents: We randomly divide the dataset into 80 and 20 percent, then calculate and rank the similarity between the documents in the two subsets. To assure the reliability, we repeat the process for ten times and the max similarity scores are 0.3456, 0.5537, 0.4357, 0.5321, 0.3386, 0.4462, 0.3965, 0.5212, 0.4850, and 0.3754, respectively. The results confirm that the duplication issue is unlikely to exist.
- Check the similarity between the domain concept sets extracted from different sizes of data: When increasing the data size by ten percent each time, the similarity between different domain concept sets extracted from that data is around 97%. The results demonstrate that the domain concept model can extract a comprehensive feature set for text classification.

### *4.2. Baselines*

We compare the proposed domain concept-based machine learning framework on legal text classification with the following baselines: the first group is TF-IDF-based machine learning methods while the next is deep learning-based methods.

- We first compare our method with TF-IDF features combined with different classifiers, which is the most widely used method in text classification. The features are extracted based on the bag-of-words (1-gram to 3-gram) from the whole training dataset; we select the same amount of features as the domain concept model. Since RFs outperforms all the other classifiers, such as SVM, Logistic Regression, NB, we only report the results of RFs in this paper. For the RF classifier, we apply the same parameters as the parameters set for the domain concept model.
- We also compare our method with deep learning models, including LSTM, BiLSTM, TextCNN, and BiLSTM with Attention. For the document representation, we explore Word2vec, GloVe, and BERT.

[7] https://osf.io/qvg8s/wiki/home/

**Table 2**
The overall results of legal text classification.

| Model | Accuracy | Recall | Precision | F1 score |
|---|---|---|---|---|
| TF-IDF + Random forests | 47.03% | 31.37% | 40.92% | 29.82% |
| Skip + TextCNN | 38.47% | 32.36% | 38.20% | 32.70% |
| Skip + BiLSTM | 44.03% | 39.24% | 43.23% | 39.79% |
| Skip + BiLSTM + Attention | 49.23% | 45.52% | 47.96% | 44.13% |
| CBOW + TextCNN | 36.96% | 31.81% | 36.08% | 32.02% |
| CBOW + BiLSTM | 44.42% | 40.32% | 42.79% | 39.65% |
| CBOW + BiLSTM + Attention | 47.58% | 43.53% | 48.29% | 43.07% |
| GloVe + TextCNN | 32.31% | 27.11% | 37.19% | 27.21% |
| GloVe + BiLSTM | 42.69% | 37.64% | 45.67% | 37.96% |
| GloVe + BiLSTM + Attention | 45.47% | 47.96% | 41.26% | 41.46% |
| BERT + TextCNN | 34.81% | 30.43% | 38.97% | 29.62% |
| BERT + BiLSTM | 41.12% | 36.99% | 42.67% | 37.06% |
| BERT + BiLSTM + Attention | 48.81% | 45.19% | 48.17% | 44.15% |
| Domain concept + Random forests | **84.49%** | **69.13%** | **74.71%** | **68.42%** |

### *4.3. Evaluation metrics*

We use recall, precision, and F1 score as metrics to evaluate the performance on each category since they are the most frequently used evaluation metrics for text classification (Li et al., 2020). For the overall performance, we use macro-average precision, recall, accuracy, and macro-average F1 score (the average for different classes) as the evaluation indicators.

### *4.4. Experiment setup*

We train the models on an Ubuntu 18.04.3 LTS machine with 1 NVIDIA Tesla Titan V GPU, 8 Intel(R) CPUs (i7-9700 @3.00 GHz), and 128 GB of RAM. We set the batch size to 32, with a max sequence length of 128 and a learning rate of 2e-5 to ensure that the GPU memory is fully utilized. The dropout probability is always kept at 0.1. We use Adam with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We empirically set the max number of the epoch to 16 and save the best model on the validation set for testing. We conduct five-fold cross-validation to avoid over-fitting.

The hyper-parameter settings for baselines are described as follows:

- TextCNN mainly includes two types of layers: the convolution filter and the max-pooling layer. To capture the potential features, three one-dimensional convolution filters are adopted. The number of filters and kernel size of the first one is 256 and 5, respectively. A max-pooling layer is then conducted to decrease the number of network parameters, and the corresponding pooling size is $3 \times 3$. Subsequently, the number of filters and the kernel size of the second one is 128 and 5, respectively. Notably, the same pooling layer is further used after the second convolution. At the end of the TextCNN architecture, a one-dimensional convolution filter with 64 filters and three kernel sizes are conducted to detect any final unseen features.
- Compared to CNN, BiLSTM has great benefits in processing the sequence information through unit collaboration and detecting deep features by forward and backward LSTM, which can solve the gradient deficiency problems. In BiLSTM, the cell and three gates work together to control the information flow. In this paper, 100 BiLSTM units are used to construct the deep learning models.
- Although BiLSTM performs well in detecting words, embedding instead of local correlations of words with Attention works better. However, different varieties of words play different roles in representing legal knowledge. The Attention mechanism has the function of detecting which words are more important in the legal domain, and it assigns the weight of words to highlight the differences. We use the SeqSelfAttention mechanism developed by Keras to further classify legal text in this study.

### *4.5. Experiment results and analysis*

#### *4.5.1. Main experiment results*

The overall results for legal text classification are presented in Table 2. The results for Domain Concept + Random Forests is based on all the 9811 features generated by the domain concept extraction on the whole corpus.

We make the following observations:

- Domain Concept + Random forests shows the strongest performance compared with the other baselines, demonstrating the effectiveness of domain concepts in legal text classification.
- In terms of all the evaluation metrics, TF-IDF + Random forests and Domain Concept + Random forests outperform other pre-trained embedding + deep learning models, demonstrating the traditional feature-based machine learning methods can sufficiently apply domain-specific information to legal text classification in the scenarios where terms distribution is obviously different between categories.

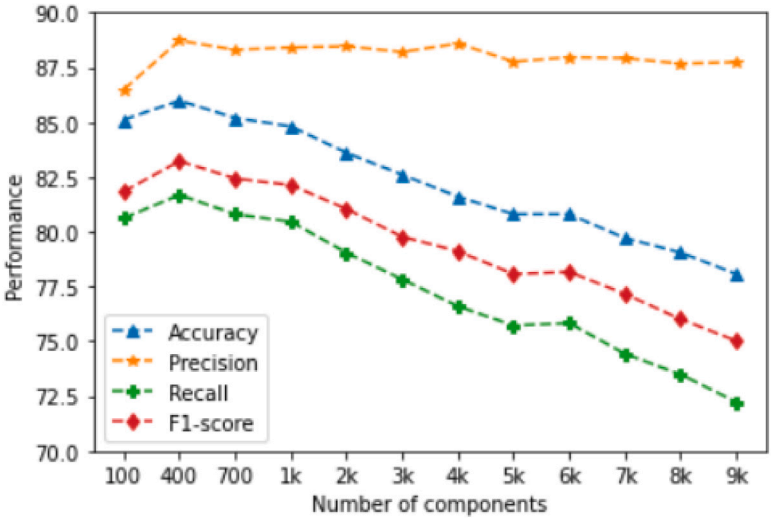


**Fig. 3.** Chart showing the performance of the domain concept with feature selection using PCA; parameters: n-estimators = 72, max-depth = 10, min-samples-leaf = 5, random-state = 0. The highest accuracy (85.98%), precision (88.72%), recall (81.68%), and F1-score (83.22%) are achieved using the top 400 principal domain concepts.

- The performance of Domain Concept + Random forests improves the baseline method TF-IDF + Random forests by 37.46%, 37.76%, 33.79%, and 38.60% on accuracy, recall, precision, and F1 score, respectively, indicating that the domain concept features are more effective than TF-IDF features for our legal text classification task.
- BiLSTM outperforms TextCNN, while BiLSTM + Attention outperforms BiLSTM, indicating that RNN models are better than CNN models in our legal text classification task, and further verifying the effectiveness of the Attention mechanism in legal text classification.

Our experimental results also reflect why the feature-based machine learning approach is still more popular than pre-trained embeddings-based methods in the legal text analysis field, although the latter has been proven effective in many NLP tasks.

#### *4.5.2. Impact of selecting different number of domain concepts as features*

We extract different amounts of principal domain concepts as the feature vector by following the steps introduced in 3.2. Fig. 3 shows the classification performances using different amount of features ranked by PCA. The highest F1 score (recall and precision) with the Random Forest classifier is obtained when the top 400 features are used: 85.98%, 88.72%, 81.68%, and 83.22% on accuracy, precision, recall, and F1 score respectively. Meanwhile, PCA can also improve the performance stability of the classifier.

Based on a bootstrapping strategy, we also conduct a t-test between the principal domain concepts that are kept and removed by PCA. We average the domain score of the domain concepts in the remaining and removed domain concepts, respectively. The t-test ($p < 0.001$) shows a significant difference between the two groups of domain concepts. It demonstrates the effectiveness of PCA in selecting important domain concepts for the legal text classification.

#### *4.5.3. Results of different categories*

When checking the performance on each category, we find eight categories that achieve 0.0 on accuracy when all of the 9811 features are selected, while none of the categories achieves 0.0 on accuracy when the top 400 features are applied, as shown in Table 3. We notice that the lowest performance on the F1 score is 39%. Twenty categories achieve over 90% on the F1 score. Moreover, most of the categories (72%) achieve quite high performance (80% or over on the F1 score). The performance is not significantly affected by different domains or training data size, which suggests that the top 400 domain concepts as features with RFs classifier is also robust and stable in practice. These results interestingly show that a small portion of the domain concepts can capture the most information in the different legal domains.

#### *4.5.4. Most effective features overall and on each category*

To in-depth analyze which of the domain concepts are among the most effective features, we list the top 30 domain concepts and their score (weight) in Table 4. Table 5 presents the top 10 domain concepts for five categories. Those terms in both tables are the roots after stemming and lemmatization. We find that the possibilities of these terms appearing in each category are different. For example, the possibilities of the term *prison* in categories 42, 20, 45, 41, 36, 30, 16, 44, 34, and 1 is 24.85%, 72.54%, 2.21%, 13.73%, 9.45%, 2.72%, 1.94%, 13.1%, 32.95%, and 12.13%, respectively. In one aspect, it provides good interpretability of why the domain concept model is more fit for legal text classification. In another aspect, it also explains why these terms can be the most important features in the legal classification task.

**Table 3**
The classification performance on each category when the top 400 principal domain concepts were selected by PCA.

| Category | F1 score | Category | F1 score | Category | F1 score |
|---|---|---|---|---|---|
| 1 | 79% | 21 | 93% | 43 | 81% |
| 4 | 75% | 22 | 95% | 44 | 86% |
| 5 | 91% | 24 | 89% | 45 | 91% |
| 6 | 82% | 25 | 84% | 51 | 93% |
| 7 | 98% | 26 | 99% | 52 | 93% |
| 8 | 80% | 28 | 97% | 56 | 96% |
| 9 | 74% | 29 | 95% | 62 | 84% |
| 10 | 87% | 30 | 96% | 65 | 95% |
| 11 | 69% | 31 | 99% | 66 | 57% |
| 12 | 91% | 33 | 76% | 67 | 81% |
| 13 | 66% | 34 | 87% | 68 | 83% |
| 14 | 79% | 36 | 96% | 69 | 69% |
| 15 | 56% | 38 | 77% | 71 | 84% |
| 16 | 84% | 39 | 54% | 73 | 99% |
| 18 | 74% | 40 | 39% | 75 | 94% |
| 19 | 67% | 41 | 80% | 78 | 83% |
| 20 | 93% | 42 | 93% | | |

**Table 4**
The top 30 most effective domain concepts (after stemming and lemmatization) and their scores (average domain score on the 50 categories). All the 400 features and their domain score are available on GitHub.

| Domain concept | Score | Domain concept | Score | Domain concept | Score |
|---|---|---|---|---|---|
| leixing | 0.1668 | detour | 0.0624 | clean | 0.0563 |
| mad | 0.0806 | brb | 0.0613 | highlight | 0.0563 |
| insur | 0.0782 | rephras | 0.0610 | aqueduct | 0.0558 |
| depress | 0.0764 | congruent | 0.0607 | cotati | 0.0551 |
| cinch | 0.0739 | assort | 0.0606 | trumpet | 0.0551 |
| veteran | 0.0679 | tick | 0.0598 | reason | 0.0550 |
| lender | 0.0669 | conglomer | 0.0597 | contractor | 0.0550 |
| efor | 0.0669 | blink | 0.0588 | maynard | 0.0550 |
| dispossess | 0.0660 | enclosur | 0.0568 | jack | 0.0548 |
| sorema | 0.0628 | dill | 0.0564 | evil | 0.0545 |

**Table 5**
The top 10 domain concepts for five categories.

| #Rank | Cate_1 | Cate_16 | Cate_20 | Cate_36 | Cate_42 |
|---|---|---|---|---|---|
| 1 | inim | countless | asylum | depress | bia |
| 2 | misco | richter | reason | leixing | environment |
| 3 | cue | alteriu | leixing | writ | deshaney |
| 4 | aqueduct | leixing | gunter | trooper | leixing |
| 5 | conglomer | lang | bulwark | censu | reason |
| 6 | har | veteran | writ | tall | wildli |
| 7 | leixing | lender | laser | asylum | notat |
| 8 | ssue | efor | recap | mad | misidentifi |
| 9 | enclosur | purg | katrina | telecommun | tower |
| 10 | holloway | valueless | par | vote | swab |

#### *4.5.5. Running time analysis*

We also examined the running time of the domain concept-based machine learning method with respect to the size of features and the running time of the pre-trained embeddings-based deep learning models. We find that the domain concept-based machine learning method is much faster than the pre-trained embeddings-based deep learning models, no matter the feature size of the former. In addition, it is time-consuming to convert the large scale of the long text into the vector space by using the pre-trained embeddings. The analysis demonstrates that domain concepts are more effective and efficient than the pre-trained embeddings.

#### *4.5.6. Discussion*

The reason word embeddings-based deep learning models fail to achieve the expected performance might be two-fold: first, texts in a particular field, such as legal, financial, and medical texts, contain many specific words, or domain experts use intelligible slang and abbreviations, which make the existing pre-trained word embeddings challenging to work with (Li et al., 2020). A domain-specific pre-training word embedding can overcome this issue (Nguyen et al., 2018). The other reason might be the lack of training data in a specific domain: deep learning requires a large amount of data. For example, the average length of the legal texts in the research is around 4000 words, and when word embeddings such as BERT was used to encode the texts in the 50 categories, the

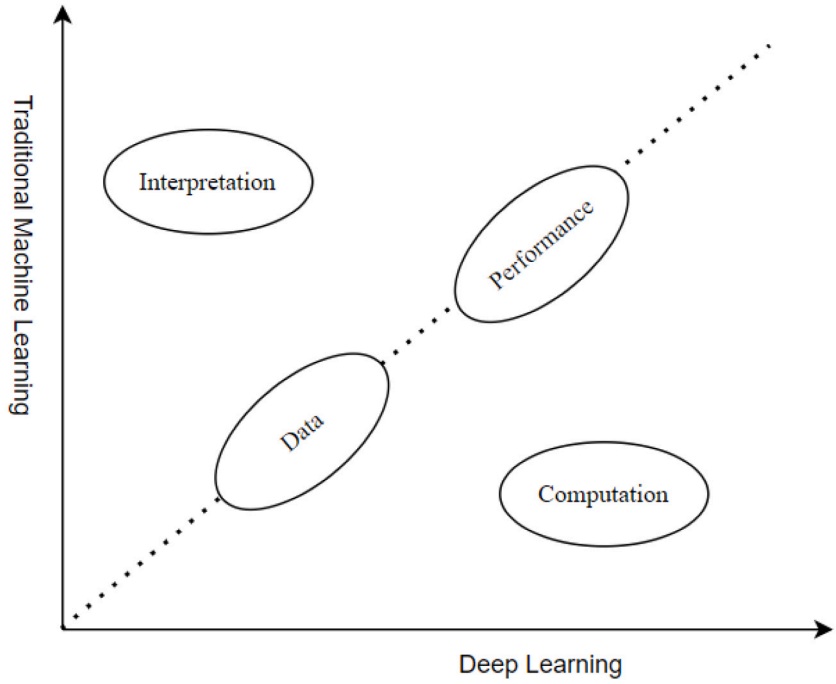


**Fig. 4.** The framework of model selection for domain-specific text classification. Traditional machine learning models are better in interpretation, while deep learning models have a higher computation requirement. As for data and performance, we need to consider different scenarios for model selection.

parameters could be several millions. The amount of documents in Table 1 in each category as training data cannot achieve decent predictive capabilities. Therefore, when only a small sample text data is available, deep learning is unlikely to outperform other approaches (Kowsari et al., 2019). However, it is usually more challenging to acquire labeled data in a specific domain than the general domain.

In this situation, we can use external features such as domain concepts to build the classifier. In one aspect, the domain concept-based machine learning approach would need less training data since the features are pre-selected based on domain knowledge. In another aspect, the difference of the terminology (i.e., the candidates of domain concepts) distribution between two categories in domain-specific data (such as the legal text collection in this paper) normally is significant. Therefore, the appropriately selected domain concepts could accurately separate texts into different categories. For example, terms such as *immigration*, *statutes*, *regulations*, *naturalization* are frequently used in immigration law cases, while *environment*, *natural*, *protection*, *land*, and *chemicals* are more often used in environmental law cases. The domain concepts may capture more effective features than word embedding with limited training data. As shown in the above experiment, only a small portion of principal domain concepts can generate a high-performance and robust legal text classifier. In addition, it provides good interpretability of why the domain concept model is a better fit for legal text classification.

## 5. Further discussions and implications

The experimental results indicate that the proposed domain concept-based random forests classifier outperforms the word embeddings-based deep learning algorithms on legal text classification, suggesting the effectiveness of domain concepts as features for domain-specific text classification. Researchers can apply the domain concepts for building high-performance and robust domain-specific intelligent systems, especially when in-domain pre-trained word embeddings are not available. In this section, we will further discuss the implications of our research.

According to Li et al. (2020), deep learning models such as BERT can get better performance on most text classification datasets, which means that we can always implement deep learning models first to get SOTA results. However, this conclusion does not fit the domain-specific text classification tasks, as demonstrated by the comparative study conducted above. When we select a machine learning model for building the classifier, many factors need to be considered. Although many studies have applied different algorithms, either shallow models or complex deep learning models, to text classification, few have discussed model selection strategies. Based on the experimental study in previous sections and a comprehensive investigation of existing research, we propose a framework with four factors that need to be considered when choosing a machine learning model for text classification: **data**, **performance**, **computation**, and **interpretation**. (See Fig. 4).

### 5.1. Data

"Data" means a machine learning model should be selected based on the characters of the data. For a text classification task, data is essential for model selection. The text data mainly can be divided into short text or long text, monolingual or cross-language, single-label or multi-label, binary class or multiclass, less sample text or huge amount of text, general text or domain-specific text, and so on. The characteristics of these data will largely determine the feature extraction quality, the model training efficiency, and, finally, the performance of the classifier. Therefore, the input datasets should be analyzed for classification. For example, for long text classification, deep learning models such as RNN and LSTM might be a better choice since they can capture valuable contextual information. This can be proved by the legal text classification results reported above: RNN-based models outperform CNN-based models even with the same text representation.

However, towards small datasets, especially in specific domains such as legal, financial, and medical, where texts contain many specific words, or domain experts use intelligible slang or abbreviations, shallow models are generally more suitable than deep learning models (Li et al., 2020). However, deep learning models are superior in handling a large amount of data since complex architectures can be designed for feature learning and model training. Naturally, a shallow learning model is prior to a deep learning model for the legal text classification in this paper. Compared to text classification with binary classes or only a few classes, large label space raises research challenges such as data sparsity and scalability (Liu et al., 2017). Binary classifiers are not as effective as multiclass classifiers. Both the RFs and deep learning models have been proven suitable for multiclass classification (Liu et al., 2017; Prinzie & Van den Poel, 2008). Given RFs' robustness and competence for analyzing large feature spaces (Prinzie & Van den Poel, 2008), we combine it with the domain concept models, which can extract a set of effective features from the legal text dataset. Our innovative model, combined with the domain concept for feature extraction and RFs for classification, achieves SOTA performance for legal text classification.

### 5.2. Performance

"Performance" means how a classifier can accurately classify texts into different categories; both the whole dataset and the category level should be measured. Performance is the most important indicator to evaluate a classifier. Since the deep learning models consist of artificial neural networks that simulate the human brain to automatically learn high-level features from data (Li et al., 2020), they usually get better performances than shallow models in many NLP tasks. In other words, if we want to achieve good performance for text classification tasks, we can try to implement deep learning models first, with BERT-based models in particular. However, in domain-specific datasets, the advantages of BERT are weakened since it is trained in the general text corpus. Usually, we need to retrain or fine-tune the pre-trained models with domain data, which is costly. If we can use a better strategy, for example, the domain concept model in this paper, to effectively extract features, shallow models such as RFs can also achieve comparative results, as is demonstrated in this paper.

### 5.3. Computation

"Computation" refers to the resources, cost, and time required. It is another essential factor that needs to be considered during model selection. According to the comparative study in this paper, the domain concept with RFs classifier trained on a CPU machine can outperform BERT with RNN models, which is trained on a NVIDIA Tesla Titan V GPU machine, and the training time is much shorter. Therefore, for domain-specific text classification, when the computation is limited, we can consider a bootstrap aggregation model such as RFs, and a boosting model such as XGBoost. RFs and XGBoost have arguably the potential to provide excellent performance recently (Kowsari et al., 2019). The domain concept model is promising for extracting features as the input of these models. Otherwise, by increasing the training data, the computation power, and the complexity of model architecture should be the best strategy to get better classification performance.

### 5.4. Interpretation

One of the well-known drawbacks of deep learning models is interpretability. Deep learning is a black-box model, as the feature extraction and model training process are opaque, and the implicit semantics and output interpretability are poor. Furthermore, most of the deep learning models are challenging to reproduce. On the contrary, for shallow models, we can accurately explain why the model improves performance. For example, the domain concept-based RFs improves the legal text classification performance after feature selection. We can easily analyze how different features contribute to model improvement. Moreover, by analyzing the correlation between features and categories, we can figure out how to balance the performance on each category.

Although many factors need to be considered for model selection, the data and the computational ability determine that only limited models can be selected. For example, with a small dataset and a limited computation resource, shallow models should be the first choice. However, for domain-specific text classification, graph neural network-based models can obtain excellent performance with external resources such as domain knowledge graph and ontology. In most situations, models for text classification should be selected by the trade-off between data and classification performance and computational resources and model interpretability.

## 6. Summary and future work

In this paper, we investigated the problem of how to select a suitable machine learning model for legal text classification. To answer this question, we conduct a comparative study of legal text classification using two different types of approaches: domain concept-based classification using random forests, and word embeddings-based using deep neural networks. Domain concept-based random forests apply the ATCT framework (Meijer et al., 2014) to extract useful terminologies from the legal collection as features and use the random forests algorithm to train the classifier. In addition, we compare this method with several deep learning models based on multiple pre-trained models including Word2vec, GloVe, and BERT. Experimental results on the sub-dataset of SigmaLaw show that the domain concept-based random forests classifier increases the accuracy, recall, precision, and F1 score by 35.26%, 21.17%, 26.54%, and 24.27%, respectively, over the best performance of the pre-trained word embeddings-based deep learning approaches, and also outperforms the TF-IDF-based random forests classifier. We also investigate how the size of the domain concept affects the classification performance, finding that the top 5% of the domain concepts can generate the most effective and robust random forests classifier. Finally, we propose a framework with guidelines on four factors including data, performance, computation, and interpretation for model selection in domain-specific text classification.

In the future, we will investigate the theoretical relationship between domain concept quality and the performance of legal text classification. Another promising direction, considering the extensive recent attention to knowledge graph research, is to construct a high-quality legal knowledge graph with domain concept information using graph neural network theory for this task.

## CRediT authorship contribution statement

**Haihua Chen:** Data curation, Conceptualization, Methodology, Software, Formal analysis, Visualization, Writing – original draft. **Lei Wu:** Methodology, Software, Validation. **Jiangping Chen:** Reviewing and editing. **Wei Lu:** Reviewing and editing. **Junhua Ding:** Supervision, Project administration, Funding acquisition, Reviewing and editing.

## Acknowledgments

The authors would like to thank Marie Bloechle and Huyen Nguyen at the University of North Texas for editing the language and writing of the paper. The authors are grateful to all the anonymous reviewers for their precious comments and suggestions.

## References

Adhikari, A., Ram, A., Tang, R., & Lin, J. (2019). DocBERT: BERT for document classification. arXiv:1904.08398.

Aletras, N., Tsarapatsanis, D., Preoţiuc-Pietro, D., & Lampos, V. (2016). Predicting judicial decisions of the European court of human rights: A natural language processing perspective. *PeerJ Computer Science*, *2*, Article e93.

Altınel, B., & Ganiz, M. C. (2018). Semantic text classification: A survey of past and recent advances. *Information Processing & Management*, *54*(6), 1129–1153.

Boella, G., Di Caro, L., & Humphreys, L. (2011). Using classification to support legal knowledge engineers in the Eunomos legal document management system. In *Fifth international workshop on juris-informatics*.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Chalkidis, I., Androutsopoulos, I., & Aletras, N. (2019). Neural legal judgment prediction in English. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 4317–4323).

Chalkidis, I., Fergadiotis, E., Malakasiotis, P., Aletras, N., & Androutsopoulos, I. (2019). Extreme multi-label legal text classification: A case study in EU legislation. In *Proceedings of the natural legal language processing workshop 2019* (pp. 78–87).

Chen, T., Xu, R., He, Y., & Wang, X. (2017). Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. *Expert Systems with Applications*, *72*, 221–230.

Chen, G., Ye, D., Xing, Z., Chen, J., & Cambria, E. (2017). Ensemble application of convolutional and recurrent neural networks for multi-label text categorization. In *Proceedings of the 2017 international joint conference on neural networks* (pp. 2377–2383).

Devlin, J., Chang, M. -W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: human language technologies*: *Vol. 1*, (pp. 4171–4186).

Ethayarajh, K. (2019). How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing* (pp. 55–65).

Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *The Journal of Machine Learning Research*, *15*(1), 3133–3181.

Gong, L., & Ji, R. (2018). What does a TextCNN learn? arXiv:1801.06287.

Graves, A., Mohamed, A. -R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 6645–6649).

Guo, B., Zhang, C., Liu, J., & Ma, X. (2019). Improving text classification with weighted word embeddings via a multi-channel TextCNN model. *Neurocomputing*, *363*, 366–374.

Islam, M. Z., Liu, J., Li, J., Liu, L., & Kang, W. (2019). A semantics aware random forest for text classification. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 1061–1070).

Jain, P., Ross, R., & Schoen-Phelan, B. (2019). Estimating distributed representation performance in disaster-related social media classification. In *Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining* (pp. 723–727).

Ji, D., Gao, J., Fei, H., Teng, C., & Ren, Y. (2020). A deep neural network model for speakers coreference resolution in legal texts. *Information Processing & Management*, *57*(6), Article 102365.

Ji, D., Tao, P., Fei, H., & Ren, Y. (2020). An end-to-end joint model for evidence information extraction from court record document. *Information Processing & Management*, *57*(6), Article 102305.

Jin, D., Jin, Z., Zhou, J. T., & Szolovits, P. (2020). Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 8018–8025).

Kang, Y. -B., Haghighi, P. D., & Burstein, F. (2014). CFinder: An intelligent key concept finder from text for ontology development. *Expert Systems with Applications*, *41*(9), 4494–4504.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 conference on empirical methods in natural language processing* (pp. 1746–1751).

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. *Information*, *10*(4), 150.

Kurfalı, M., & Östling, R. (2019). Noisy parallel corpus filtering through projected word embeddings. In *Proceedings of the fourth conference on machine translation: Volume 3, Shared Task Papers, Day 2* (pp. 277–281).

Leitner, E., Rehm, G., & Moreno-Schneider, J. (2019). Fine-grained named entity recognition in legal documents. In *Semantic systems. The power of AI and knowledge graphs: 15th international conference* (pp. 272–287).

Li, X., Bing, L., Zhang, W., & Lam, W. (2019). Exploiting BERT for end-to-end aspect-based sentiment analysis. In *Proceedings of the 5th workshop on noisy user-generated text* (pp. 34–41).

Li, Q., Peng, H., Li, J., Xia, C., Yang, R., & Sun, L. (2020). A survey on text classification: From shallow to deep learning. *ACM Computing Surveys*, *37*(4), Article 35.

Li, S., Zhang, H., Ye, L., Guo, X., & Fang, B. (2019). MANN: A multichannel attentive neural network for legal judgment prediction. *IEEE Access*, *7*, 151144–151155.

Liu, J., Chang, W. -C., Wu, Y., & Yang, Y. (2017). Deep learning for extreme multi-label text classification. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval* (pp. 115–124).

Liu, G., & Guo, J. (2019). Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing*, *337*, 325–338.

Ma, M., Podkopaev, D., Campbell-Cousins, A., & Nicholas, A. (2020). Deconstructing legal text_Object oriented design in legal adjudication. arXiv preprint arXiv:2009.06054.

Matas, N. (2017). Comparing network centrality measures as tools for identifying key concepts in complex networks: A case of Wikipedia. *Journal of Digital Information Management*, *15*(4), 203–213.

Meijer, K., Frasincar, F., & Hogenboom, F. (2014). A semantic approach for extracting domain taxonomies from text. *Decision Support Systems*, *62*, 78–93.

Mikolov, T., Chen, K., Corrado, G. S., & Dean, J. (2013). Efficient estimation of word representations in vector space. In *Proceedings on the international conference on learning representations*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Moens, M. -F., Boiy, E., Palau, R. M., & Reed, C. (2007). Automatic detection of arguments in legal texts. In *Proceedings of the 11th international conference on artificial intelligence and law* (pp. 225–230).

Nazarenko, A., & Wyner, A. (2017). Legal NLP introduction. *Association pour le Traitement Automatique des Langues*.

Nguyen, T. -S., Nguyen, L. -M., Tojo, S., Satoh, K., & Shimazu, A. (2018). Recurrent neural network-based models for recognizing requisite and effectuation parts in legal texts. *Artificial Intelligence and Law*, *26*(2), 169–199.

Octavia-Maria, Zampieri, M., Malmasi, S., Vela, M., P. Dinu, L., & van Genabith, J. (2017). Exploring the use of text classification in the legal domain. In *Proceedings of 2nd workshop on automated semantic analysis of information in legal texts*.

Palau, R. M., & Moens, M. -F. (2009). Argumentation mining: The detection, classification and structure of arguments in text. In *Proceedings of the 12th international conference on artificial intelligence and law* (pp. 98–107).

Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing* (pp. 1532–1543).

Prinzie, A., & Van den Poel, D. (2008). Random forests for multiclass classification: Random multinomial logit. *Expert Systems with Applications*, *34*(3), 1721–1732.

Pudaruth, S., Soyjaudah, K., & Gunputh, R. (2018). An innovative multi-segment strategy for the classification of legal judgments using the k-nearest neighbour classifier. *Complex & Intelligent Systems*, *4*(1), 1–10.

Reimers, N., Schiller, B., Beck, T., Daxenberger, J., Stab, C., & Gurevych, I. (2019). Classification and clustering of arguments with contextualized word embeddings. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 567–578).

Rezaeinia, S. M., Ghodsi, A., & Rahmani, R. (2017). Improving the accuracy of pre-trained word embeddings for sentiment analysis. arXiv:1711.08609.

Šajatović, A., Buljan, M., Šnajder, J., & Bašić, B. D. (2019). Evaluating automatic term extraction methods on individual documents. In *Proceedings of the joint workshop on multiword expressions and WordNet* (pp. 149–154).

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, *24*(5), 513–523.

Sclano, F., & Velardi, P. (2007). TermExtractor: A web application to learn the shared terminology of emergent web communities. In *Enterprise interoperability II* (pp. 287–290).

Shaheen, Z., Wohlgenannt, G., & Filtz, E. (2020). Large scale legal text classification using transformer models. arXiv:2010.12871.

Soh, J., Lim, H. K., & Chai, I. E. (2019). Legal area classification: A comparative study of text classifiers on singapore supreme court judgments. In *Proceedings of the natural legal language processing workshop 2019* (pp. 67–77).

Song, F., Guo, Z., & Mei, D. (2010). Feature selection using principal component analysis. In *Proceedings international conference on system science, engineering design and manufacturing informatization* (pp. 27–30).

Sugathadasa, K., Ayesha, B., de Silva, N., Perera, A. S., Jayawardana, V., & Lakmal, D. (2017). Synergistic union of word2vec and lexicon for domain specific semantic similarity. In *2017 IEEE international conference on industrial and information systems* (pp. 1–6).

Şulea, O. -M., Zampieri, M., Vela, M., & van Genabith, J. (2017). Predicting the law area and decisions of French Supreme Court cases. In *Proceedings of the international conference recent advances in natural language processing* (pp. 716–722).

Sundermeyer, M., Schlüter, R., & Ney, H. (2012). LSTM neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., & Gomez, A. N. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008)..

Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, *2*(1–3), 37–52.