Journal of Nonlinear and Convex Analysis
Volume 26, Number 6, 2025, 1581–1592

Yokohama Publishers
ISSN 1880-5221 ONLINE JOURNAL

A TOPIC EMBEDDINGS-BASED LSTM APPROACH FOR CHINESE LEGAL TEXT CLASSIFICATION

YANGWU ZHANG*, GUOHE LI, LIJIE CUI, XIAO PU, LINGYAN BIAN, AND LEI SHI

Abstract. Legal text classification provides people with additional search functions when the search and application of similar cases are now required to ensure the uniform and proper implementation of laws during the case process. Deep learning has been widely used to analyze and process text classification in the corpus of natural language. However, most deep learning models adopt either bag-of-words or word embeddings without considering the long sequential text, whereas standard legal texts belong to long texts. We propose the topic-embeddings framework of Long Short-Term Memory (LSTM) to leverage LDA, which models segments embedding as input of LSTM. Each segment embedding is a context-varying vector in LSTM related to one paragraph of a legal text. According to the subdivision of Chinese legal practice area, topic-embeddings LSTM can set the best the number of topics and the size of embeddings. The experimental results on China legal corpus suggest that our model outperforms SVMs, KNNs, DTs, RFs, MLP, AdaBoost, and NB in terms of precision, recall, and F1.

1. Introduction

The development of AI in the legal sector can boost efficiency and cut costs in legal matters [20]. Legal text classification provides the legal profession with other search functions when they take a new case facing to classify its civil, economic, or tort liability laws [11, 18]. The deep learning model can outperform in short text, such as movie reviews, tweets, blogs, etc. However, the model of deep learning in the legal industry doesn't work as well as expected. One reason is the length of legal text and insufficient labeled samples for model training [6, 15].

Deep learning in the legal domain has a significant problem in terms of legal text representation [9]. The traditional bag of words model will ignore related and critical information existing in the context [16]. Although word embedding can capture contextual information, the increase in sentence length makes the model even more complicated [1], so it is hard to train the model with a massive volume of parameters.

This paper proposes the topic-embedings framework of Long Short-Term Memory based on Latent Dirichlet Allocation. It classifies legal areas in the corpus of Chinese

2020 *Mathematics Subject Classification.* 60E05, 62F15.

Key words and phrases. Topic model, LDA, LSTM, Chinese legal text, classification, embeddings.

This work was supported by Research Foundation of China University of Petroleum-Beijing at Karamay(NO.XQZX20240032) and Education Ministry's Collaborative Education Program with Industry (NO.231007632264016).

*Corresponding author.

legal text, carrying out experiments to study the model's performance using real-world data from China Judgments Online. Our study aims to solve the sequence and length of legal text in deep learning by performing segments on sequentially processed paragraphs in a topic-guided manner.

2. Related work

Mikolov, Chen et al. proposed the model of word embedding architectures for computing continuous vector space representations of words [12], which yield better results from syntactic and semantic word similarities. Chalkidis, Androutsopoulos et al. experimentally compared several contract element extraction methods that use manually written rules and classifiers with word embedding and part-of-speech embedding on the dataset with gold contract element annotations [3].

Blei and Lafferty used the Latent Dirichlet Allocation model to analyze document collections and other data distribution in a mixture of topics [2]. Talley, O'Kane et al. developed a protocol built on Latent Semantic Analysis and Regular Expression [19], which could be used broadly in force majeure provisions amid business law. Sulea, Zampieri et al. proposed the technique of text classification methods to classify the law area from features map utilizing Latent Semantic Analysis and predict the decision of cases judged by the French Supreme Court [17]. Dieng, Ruiz et al. developed the dynamic embedded topic model by using dynamic Latent Dirichlet Allocation and word embedding [5], parameterized by the inner product between the word embedding and assigned topic.

Hochreiter and Schmidhuber introduced a gradient-based method called long short-term memory (LSTM) [7], which can learn to bridge minimal time lags by opening and closing access to the constant error flow within multiplicative gate units. Cho, Van Merriënboer et al. used two recurrent neural networks (RNN) to model RNN Encoder–Decoder [4]. One RNN encodes a sequence of tokens of symbols into a fixed-length vector space representation, and the other decodes the vector representation into another sequence of tokens of symbols. Jozefowicz, Zaremba et al. conducted a thorough architecture search to determine whether the LSTM is best over ten thousand different RNN architectures [8].

Vaswani, Shazeer et al. proposed the Transformer architecture to avoid complex recurrent or convolutional neural networks in a seq2seq model [21], which is based solely on attention mechanisms. Peters, Neumann et al. introduced a model of deep contextualized word representation learned from the internal states of a deep bidirectional language pre-trained model on a massive volume of a text corpus [14]. Mokanov provided a legal qualification to a set of facts based on fact2law using deep learning [13]. Krasadakis et al. presented the current state-of-the-art NLP tasks related to Law Consolidation, highlighting the challenges that arise in low-resource languages [10].

3. Methodology

3.1. **Framework.** Legal areas in China are generally subsets of substantive and procedural laws regimes. The former includes civil law, criminal law, administrative law, tort, economic law, or business law. The latter contains civil procedure

law, criminal procedure law, and administrative procedure law. Moreover, these law departments are separated into many different branches to resolve disputes. Lawyers and clients need to determine the legal area governing the involved dispute or case. We can introduce a novel model of legal text classification to assist the legal profession.

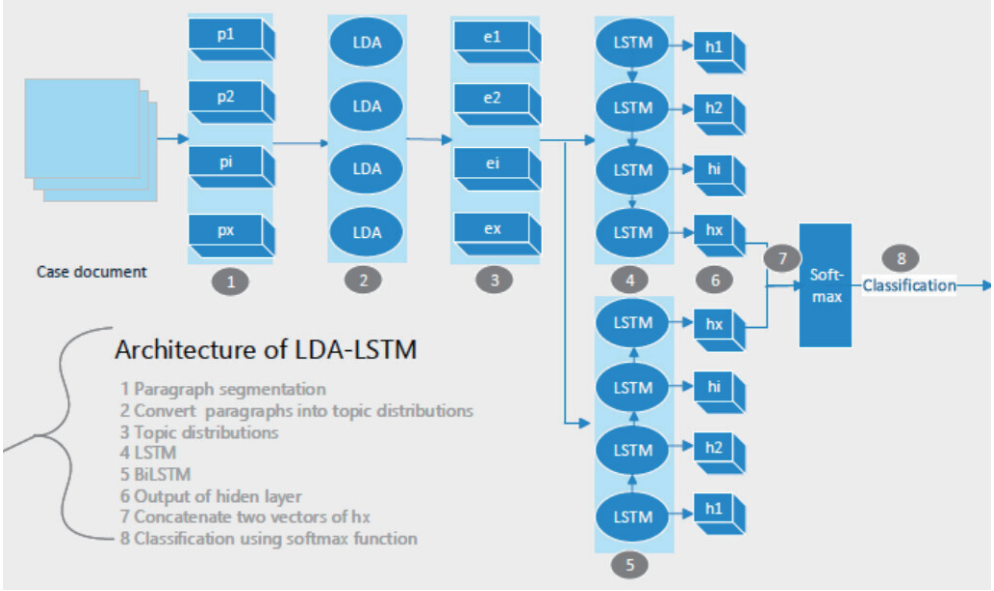

FIGURE 1. Architecture of topic embeddings LSTM

Fig. 1 shows the details and flow of our framework where a case document is processed through it. Step 1 segments a case document into multiple paragraphs by using paragraph marks, such as carriage return or line feed, then convert it into the representation of Bag-of-Words (say term frequency). Step 2 leverages LDA to map representation of term frequency into the paragraph-topic distribution assuming a paragraph consists of some latent topics. At the same time, a latent topic contains many words defined in the vocabulary. Step 3 performs normalization in a batch to improve the smoothness of the surface of errors that guided the update of parameters. Step 4 processes the inputs of sequential vectors through the RNN cell with forgetting gates and memory gates, which may retain contextual information related to the input sequences. Step 5 reverses the sequential input vectors to feed the LSTM network to acquire more contextual knowledge. Step 6 gains a series of outputs of the hidden layer where each RNN cell receives the previous output of the neuron apart from the current input sequence. Step 7 concatenates two vectors of the LSTM hidden layer where they are the output of the last sequential status of RNN cell in bidirectional LSTM. Step 8 feeds the flattened outputs of concatenating vectors of LSTM to the Softmax function for legal area classification.

3.2. **Generative statistical method of a Chinese legal text.** LDA is one of the generative statistical probability models where the latent variables exist, i.e., a probability distribution over a pre-defined number of topics. In contrast, a topic is

the collection of many different words defined in the vocabulary mixed with different proportions. According to the idea of LDA, a Chinese legal text is produced as follows:

1. Sampling from document-topic distribution related to the legal area may be performed in a word's position in the document. A specific topic is assigned to the word's position as a latent topic behind this position.
2. Sampling from topic-word distribution related to the thesaurus and text may be carried out in a position of word assigned to a specific latent topic. A certain word of the pre-defined vocabulary is given to the position of the word.
3. The above steps are repeated until all words are generated in a document. The specific process can refer to https://gitee.com/cupkzyw/legal-bi-lstm/blob/master/images/gendocument.pdf.

3.3. **Topic Embeddings LSTM.** A collection of Chinese legal text is denoted as D with the collection, where $\mid D \mid$ is the number of samples in the collection D. Term frequency is taken into consideration to characterize the input, and there are N terms in the pre-defined vocabulary. Thus D can also be described as equation 3.1:

$$D = \begin{bmatrix} d_{11} & \cdots & d_{1n} & \cdots & d_{1N} \\ \vdots & \ddots & d_{mn} & \ddots & \vdots \\ d_{|D|1} & \cdots & d_{|D|n} & \cdots & d_{|D|N} \end{bmatrix}. \tag{3.1}$$

Each document is processed into a vector, referring to times of occurrences of the *n-th* term in the vocabulary. We aim to find out a set of θ and ϕ to minimize the distance of Kullback–Leibler divergence based on the observed evidence on the training corpus, and the optimal θ and ϕ are denoted as θ^* and ϕ^*. Thus, the output of LDA include θ^* and ϕ^*, i.e.

$$\theta^* = \begin{bmatrix} \theta_{11} & \cdots & \theta_{1k} & \cdots & \theta_{1K} \\ \vdots & \ddots & \theta_{mk} & \ddots & \vdots \\ \theta_{|D|1} & \cdots & \theta_{|D|k} & \cdots & \theta_{|D|K} \end{bmatrix} \tag{3.2}$$

and

$$\phi^* = \begin{bmatrix} \phi_{11} & \cdots & \phi_{1v} & \cdots & \phi_{1V} \\ \vdots & \ddots & \phi_{kv} & \ddots & \vdots \\ \phi_{K1} & \cdots & \phi_{Kv} & \cdots & \phi_{KV} \end{bmatrix}. \tag{3.3}$$

θ_{mk} is known as the probability of topic k in the document m. In the same way, ϕ_{kv} is referred to the probability of the word v within the topic k. Due to the length of Chinese legal text, we consider segments.

Definition 3.1 (Document Segmentation)**.** Given $A = \langle D, T, W \rangle$, D is the collection of corpus, T is the collection of topics, and W is the collection of words. If $m \in D$, $m \times R = \{p_1, p_2, \ldots, p_x, \ldots, p_X\}$ denotes that document m contains X

paragraphs. $A \times R$ reflects new datasets through segmentation for D in case T and W.

According to segmentation, the document m is divided into $|X|$ separate paragraphs, where each of them is fed to LDA to infer. In other words, the LDA model has been fully trained on datasets $A \times R$, i.e., pre-trained model. Thus, the inference result of m is as follows:

$$out\,(m) = \begin{bmatrix} em_{11} & \cdots & em_{1k} & \cdots & em_{1K} \\ \vdots & \ddots & em_{xk} & \ddots & \vdots \\ em_{|X|1} & \cdots & em_{|X|k} & \cdots & em_{|X|K} \end{bmatrix}. \tag{3.4}$$

Definition 3.2 (Paragraph Embedding)**.** Given em_x is a vector representing the paragraph x in document m, and the dimensionality of em_x is K. em_x is defined as follow:

$$em_x = [em_{x1}, em_{x2}, \ldots, em_{xk}, \ldots, em_{xK}], \tag{3.5}$$

where em_{xk} is referred to probability of topic k in paragraph x of document m, then the K dimensional vector em_{x-1} representing the topic distribution of paragraph $x-1$ of document m is fed to the LSTM cell of sequential status $x-1$ as sequential data. Similarly, the vector em_x representing the topic distribution of paragraph x of document m is fed to the LSTM cell of sequential status x, and em_{x+1} representing the topic distribution of paragraph $x+1$ of document m is fed to the LSTM cell of sequential status $x+1$. Furthermore, each LSTM cell also receives the output vector of the LSTM cell of the previous cell, as is shown in Fig. 2.

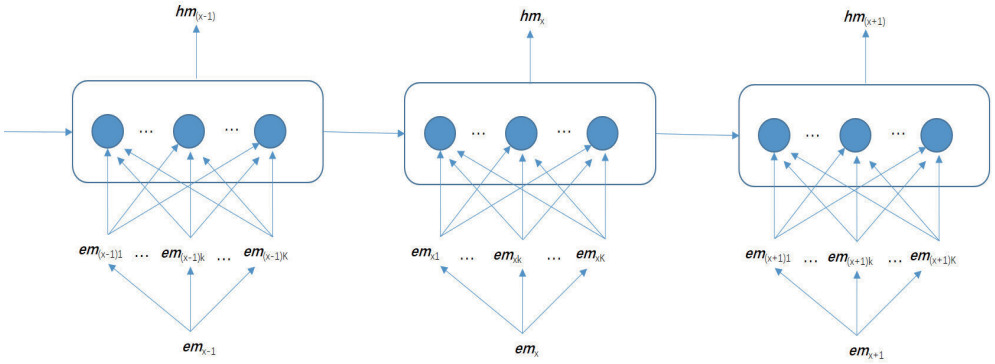

FIGURE 2. Topics distribution embedding into sequence context

Each LSTM cell can handle the vector in the sequence through input, forget and output control gates. The activate function of a control gate is sigmoid. The structure of topic embeddings LSTM is set up, which means that a set of functions is determined. What needs to be done at the next step is to solve a set of unknown parameters by machine learning which iteratively updates parameters with optimization amid training on Chinese legal text corpus.

3.4. **Algorithm.** Our approach can leverage embeddings of topic models to enhance LSTM for long texts. Topic models establish abstract semantic associations based on topics, and LSTM cells are embedded with contextual sequences. As mentioned above, the proposed topic embeddings LSTM consists of three parts:

(1) Topics-based paragraph representations. The document m in the dataset D is split into X_m paragraphs, the original dataset D turns into the new dataset $\overline{D}$, i.e., $\overline{D} = A \times R$, including $\sum_{m=1}^{|D|} X_m$ samples, and $\overline{D}$ would be fed to LDA topic model for unsupervised training. LDA initialization is performed when preprocessing is complete, where the number of topics K is crucial to paragraph embeddings in LSTM. The suitable K could be found by analyzing perplexity and coherence in the K loop.

(2) Alignment fill paragraphs embeddings. The collection $\overline{D}$ is put on the trained LDA model, and the new one $\hat{D}$ is inferred, aiming at paragraph embeddings. The original document m is mapped into $[pm_1, pm_2, \ldots, pm_{x_m}, \ldots, pm_{X_m}, pm_{X_m+1}, \ldots, pm_{X_{max}}]$, where pm_{x_m} is represented by $[em_{x1}, em_{x2}, \ldots, em_{xk}, \ldots, em_{xK}]$. $pm_1, pm_2, \ldots, pm_{x_m}, \ldots, pm_{X_m}$ are paragraphs in the document m, X_{max} is the maximal number of paragraphs in all documents, and zeros fill $pm_{X_m+1}, \ldots, pm_{X_{max}}$ to align LSTM input.

(3) Embeddings data normalization. Z standard is performed on embeddings to avoid optimization difficulties. We can calculate $\mu_k = \frac{1}{X_{max} \times |D|} \sum_{m=1}^{|D|} \sum_{x=1}^{X_{max}} em_{xk}$, and μ_k is the mean of topic No. k in all documents. At the same time, we may construct ${\sigma_k}^2 = \frac{1}{(X_{max}-1)\times(|D|-1)} \sum_{m=1}^{|D|} \sum_{x=1}^{X_{max}} (em_{xk} - \mu_k)^2$, and ${\sigma_k}^2$ is the variance of topic No. k in all documents. Then, we update em_{xk} with $\frac{em_{xk}-\mu_k}{\sigma_k}$.

4. Experiments

4.1. **Data Description.** Chinese legal text corpus comprises 1,103 cases acquired from China Judgment Online written in Chinese. Judgments with the legal area, which the corresponding court labeled, were published on the Internet. Unlike English, Chinese corpus requires performing preprocessing, i.e., word segmentation. When we split an English sentence, it is easy to get tokens of words by a few spaces and punctuations. However, a Chinese sentence is composed of several Chinese characters arranged continuously in a row. A Chinese word token is constituted of a Chinese character or more Chinese characters, and there are no spaces between the words in a Chinese sentence. Thus, the n-gram span of a Chinese word is determined by the contextual meaning, and we utilized Python library Jieba to handle Chinese word segmentation.

4.2. **Experimental setup.** LDA model maps to a set of functions, where parameters theta in document-topics distribution and phi in topic-words distribution are unknown. The theta and phi may be resolved by optimizing Kullback-Leibler divergence in the training sample. A part of the dataset is set aside for the testing, and the rest is divided into three folds, where two folds are used for training, one fold is used for validation, and it is performed alternately three times. Hyper-parameters need to be set up in advance, such as alpha, beta, and K, where alpha is the prior

probability of theta, beta is the prior probability of the phi, and K is the number of topics. Hyper-parameters in LSTM involve the number of cells of the hidden layer, the number of LSTM layers, learning rate, batch size, epochs, dropout, word embedding, and max length of a sentence. Hyper-parameters setup is shown in Table 1.

Table 1. Setup of Hyper-parameters

Hyper-parameters	Description
K	the number of topics in LDA
alpha	priori probability in Dirchlet of document-topics distribution
beta	priori probability in Dirchlet of topic-words distribution
iter	parameters theta and phi update frequently
nhid	the number of cells of the hidden layer, for example 50
nlayers	the number of LSTM layers
lr	learning rate
batch_size	samples of a batch
epochs	run the full dataset more times
dropout	randomly selected neurons are ignored during training
embsize	related to K, for example 100
sen_max_len	related to paragraphs, for example 40

This work implemented the model functionality utilizing Python language and PyTorch, and some third-party libraries are used including torch.nn, torch.optim, nnet.lstm, nnet.blstm, torch.autograd, and so on. The 1080Ti GPU of the cloud server was rented from AutoDL Cloud, and the version of CUDA is 11.6. The 1080Ti is operating at a frequency of 1481 MHz, which can be boosted up to 1582 MHz, and memory is running at 1376 MHz. The total parameters of our model can be calculated according to $4 \times ((\text{embedding\_size}+\text{hidden\_size}) \times \text{hidden\_size} + \text{hidden\_size}) \times \text{max\_len}$, and the system has the memory footprint of 120MB. Each epoch took 32–35 seconds.

5. Results and Discussion

In order to decide the optimum number of topics to be extracted using LDA for topic modeling for Chinese legal texts, topic perplexity and topic coherence scores are always used to measure how well the topics are extracted. Visualization of a topic model can refer to https://gitee.com/cupkzyw/legal-bi-lstm/blob/master/images/visualizationtopicmodel.pdf. The result is shown in Fig. 3. We can infer that a specific number of topics is 100 if the overall score of perplexity and coherence is taken into consideration according to the mentioned situation.

According to the subdivision of Chinese legal practice area, we set the number of topics (num_topics) as 100. The dimensionality of input data fed to the LSTM network is 100. The fine-tuned hyper-parameters contain learning rate (lr), dropout, size of batch (batch_size), size of the input (emb_size), epochs, and so on. The criterion function is cross-entropy, increasing as the predicted probability diverges

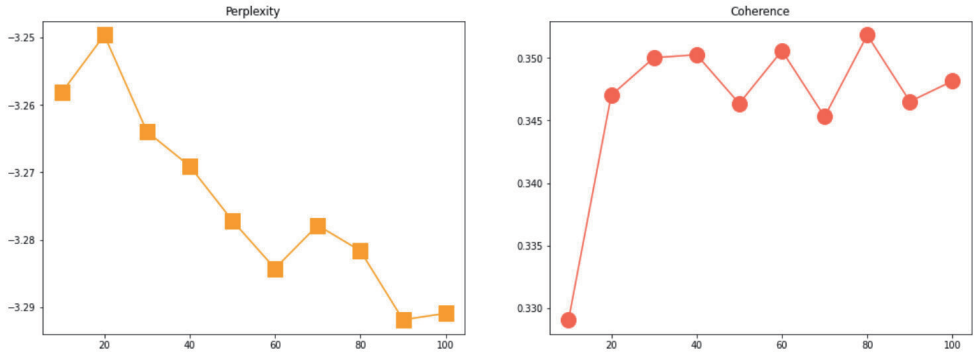

FIGURE 3. Perplexity and coherence under different topics

from the actual label. We select Adam as the optimizer considering the momentum and gradient descent [22]. The result of accuracy on the test dataset follows as Fig. 4.

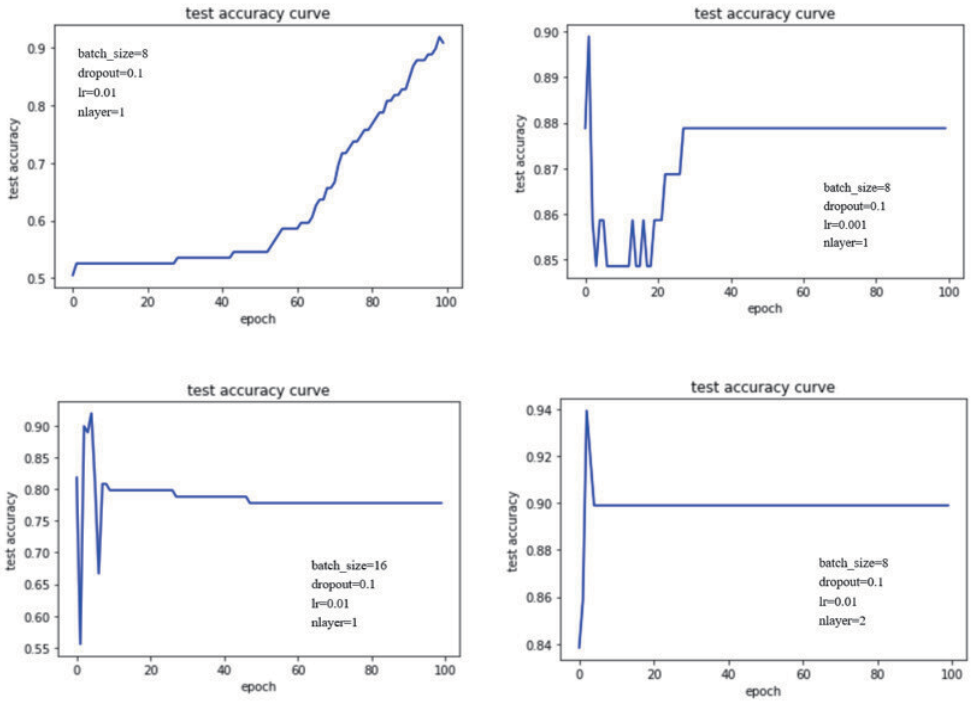

FIGURE 4. Test accuracy with different epochs

The four sets of hyper-parameters are individually shown, i.e., (batch_size = 8, dropout = 0.1, lr = 0.01, nlayer = 1), (batch_size = 8, dropout = 0.1, lr = 0.001, nlayer = 1), (batch_size = 16, dropout = 0.1, lr = 0.01, nlayer = 1), and (batch_size = 8, dropout = 0.1, lr = 0.01, nlayer = 2). The result from the above figure may suggest that the test accuracy curve varies to the different hyper-parameters. The

curve in the left-top gradually ascends when lr is equal to 0.01, whereas the initial ups and downs of the curve in the right-top is followed by flattening at the position where epochs are 28 when lr is equal to 0.001. The fluctuating curve of the left-bottom is flattened at the position where epochs are 8 when batch_size is equal to 16, and the curve located in the right-bottom is rapidly converged where epochs are 3 when nlayer is equal to 2.

Due to the varied hyper-parameters, an explanation for the different performance on the test accuracy could be related to optimization and overfitting. The update step of parameters is too tiny when lr is equal to 0.001, which would lead to the training loss of function trap at a local minimum point. The overfitting could have happened on the training when batch_size is equal to 16, which stopping training early could alleviate. However, the performance is the best when nlayer is set to 2, reflecting that the model structure of two layers may correct model bias better than one layer.

In order to analyze the performance of our model, the methods of SVMs (support-vector machines), KNNs (k-nearest neighbors), DTs (decision trees), RFs (random forests), MLP (multi-layer perceptron), AdaBoost (adaptive boosting), and NB (naive Bayes) are used to compare with topic embeddings LSTM, as is shown in Table 2. The default set of hyper-parameters in our model is (batch_size = 8, dropout = 0.1, lr = 0.01, nlayer = 1).

Table 2. Precision, recall, and F1 of each method

methods	precision	recall	F1
	positive/negative	positive/negative	positive/negative
our model (default)	0.859/**0.976**	**0.98**/0.836	0.916/0.901
our model (lr=0.001)	**0.952**/0.825	0.8/0.959	0.869/0.886
our model (batch_size=16)	0.882/0.896	0.9/ 0.877	0.891/ 0.886
our model (nlayer=2)	0.923/0.957	0.96/ 0.918	**0.941**/**0.938**
SVMs	0.753/0.772	0.754/0.771	0.75/0.772
KNNs	0.698/0.872	0.898/0.639	0.786/0.738
DTs	0.941/0.641	0.41/**0.976**	0.571/0.774
RFs	0.747/0.621	0.428/0.866	0.545/0.723
MLP	0.736/0.751	0.728/0.759	0.732/0.755
AdaBoost	0.793/0.753	0.706/0.83	0.747/0.789
NB	0.597/0.842	0.912/0.431	0.722/0.57

The above table demonstrates that our model outperforms SVMs, KNNs, DTs, RFs, MLP, AdaBoost, and NB no matter what hyper-parameters is set to which of four groups. The maximum positive precision is 0.952 when lr is equal to 0.001 in our model, whereas the maximum negative precision is 0.976 when hyper-parameters are set to default in our model. The best positive precision is DTs with 0.941 amid SVMs, KNNs, DTs, RFs, MLP, AdaBoost, and NB, and the others are all lower than 0.8. However, DTs yields a relatively low negative precision with 0.641. Our model (default) takes the lead in positive recall with 0.98, whereas DTs has

the advantage in negative recall with 0.976. Our model (nlayer=2) exceeds other methods in the positive and negative F1 with 0.941 and 0.938. At the same time, the best positive and negative F1 is individually KNNs and AdaBoost with 0.786 and 0.789 amid SVMs, KNNs, DTs, RFs, MLP, AdaBoost, and NB.

6. Conclusions

In this work, we have proposed the topic-modeling framework of topic embeddings LSTM to discover features of topics distribution embedding into sequence context in Chinese legal texts. Topic embeddings LSTM models each document with sequentially processed segments in topics distribution inferred by LDA training on the Chinese legal corpus. Each segment embedding is a context-varying vector in the embedding space of the document. Using perplexity and coherence metrics, our model can learn the best the number of topics and the size of embeddings. We applied the model to classify the Chinese legal corpus. Experimental results imply that our model outperforms SVMs, KNNs, DTs, RFs, MLP, AdaBoost, and NB in terms of precision, recall, and F1 while providing intuitive interpretation by visualization of a topic model.

Acknowledgments

We are greatly indebted to colleagues at the Haii (Human-AI Interaction laboratory), Department of Computer Science, Durham University, UK. We thank Zhaoxing Li, Yunzhan Zhou, Chenghao Xiao, Ziqi Pan, Jindi Wang, and Jialin Yu for their special suggestions and many interesting discussions. Additionally, we would like to thank Zengwei Gao, Minjie Zhang, Zhiyang Jia, and Haitao Shi from the Department of Computer Science at China University of Petroleum-Beijing at Karamay for their valuable contribution to this research.

References

[1] A. Bibal, M. Lognoul, A. De Streel and B. Frenay, *Legal requirements on explainability in machine learning*, Artificial Intelligence and Law **29** (2021), 149–169.

[2] D. Blei and J. Lafferty, *Correlated topic models*, Advances in Neural Information Processing Systems **18** (2006), 147–154.

[3] I. Chalkidis, I. Androutsopoulos and A. Michos, *Extracting contract elements*, in: Proceedings of the 16th International Conference on Articial Intelligence and Law, ACM Press, 2017, pp.19–28.

[4] K.Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, *Learning phrase representations using RNN encoder-decoder for statistical machine translation*, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2014, pp.1724–1734.

[5] A. B. Dieng, F. J. Ruiz and D. M. Blei, *The dynamic embedded topic model*, arXiv preprint, 2019, `https://arxiv.org/abs/1907.05545`.

[6] I. Glaser, E. Scepankova and F. Matthes, *Classifying semantic types of legal sentences: Portability of machine learning models*, in: Proceedings of International Conference on Legal Knowledge and Information Systems, IOS Press, 2018, pp. 61–70.

[7] S. Hochreiter and J. Schmidhuber, *Long short-term memory*, Neural Computation **9** (1997), 1735–1780.

[8] R. Jozefowicz, W. Zaremba and I. Sutskever, *An empirical exploration of recurrent network architectures*, in: Proceedings of the International Conference on Machine Learning, ML Research Press, 2015, pp, 2332–2340.

[9] R. Keeling, R. Chhatwal, N. Huber-Fliflet, J. Zhang and H. Zhao, *Using machine learning on legal matters: Paying attention to the data behind the curtain*, Hastings Science and Technology Law Journal **11** (2020), 9–20.

[10] P. Krasadakis, E. Sakkopoulos and V. S. Verykios, *A survey on challenges and advances in Natural Language Processing with a focus on legal informatics and low-resource languages*, Electronics **13** (2024): 648.

[11] D. Lehr and P. Ohm , *Playing with the data: what legal scholars should learn about machine learning*, UCDL Rev. **51** (2017), 653–570.

[12] T. Mikolov, K. Chen, G. Corrado and J. Dean. *Efficient estimation of word representations in vector space*, in: Proceedings of the International Conference on Learning Representations, ICLR, 2013, `http://arxiv.org/pdf/1301.3781`.

[13] I. Mokanov, D. Shane and B. Cerat, *Facts2law-using deep learning to provide a legal qualification to a set of facts*, in: Proceedings of the 17th International Conference on Artificial Intelligence and Law, ACM Press, 2019, pp. 268–269.

[14] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee and L. Zettlemoyer, *Deep contextualized word representations*, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, 2018, pp.2227–2237.

[15] M. Sag, *The new legal landscape for text mining and machine learning*, Journal of the Copyright Society of the U.S.A. **61** (2019), 346–350.

[16] R. Sil and A. Roy, *A novel approach on argument based legal prediction model using machine learning*, in: Proceedings of the International Conference on Smart Electronics and Communication, Institute of Electrical and Electronics Engineers Inc., 2020, pp. 487–490.

[17] O. M. Sulea, M. Zampieri, M. Vela and J. Van Genabith, *Predicting the law area and decisions of french supreme court cases*, in: Proceedings of the International Conference Recent Advances in Natural Language Processing, Incoma Ltd, 2017, pp. 716–722.

[18] H. Surden, *Machine learning and law*, Wash. L. Rev.**89** (2014), 87–99.

[19] E. Talley, D. O'Kane, C. Kellner and A. Stremitzer, *The measure of a MAC: A machine-learning protocol for analyzing force majeure clauses in M&A agreements [with comment]*, Journal of Institutional and Theoretical Economics **168** (2012), 181–208.

[20] S. Vanderbeck, J. Bockhorst and C. Oldfather, *A machine learning approach to identifying sections in legal briefs*, in: Proceedings of the 22nd Midwest Artificial Intelligence and Cognitive Science Conference, Indiana University, 2011, pp. 26–33.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, *Attention is all you need*, in: Advances in Neural Information Processing Systems, 2017, pp. 5999–6009.

[22] Q. Zhang, Y. Zhou and S. F. Zou, *Convergence guarantees for RMSProp and adam in generalized-smooth non-convex optimization with affine noise variance*, arXiv preprint, 2019, `https://arxiv.org/abs/2404.01436`.

Manuscript received January 11, 2024
revised September 12, 2024

Yangwu Zhang
School of Information Management for Law, China University of Political Science and Law, 102249, Beijing, China;
Department of Computer Science, China University of Petroleum-Beijing at Karamay, 834000, Karamay, China
E-mail address: yangwuzh@cupl.edu.cn

Guohe Li
Department of Computer Science, China University of Petroleum-Beijing at Karamay, 834000, Karamay, China
E-mail address: guoheli@cupk.edu.cn

Lijie Cui
Department of Computer Science, China University of Petroleum-Beijing at Karamay, 834000, Karamay, China
E-mail address: lijiecui@cupk.edu.cn

Xiao Pu
Department of Computer Science, China University of Petroleum-Beijing at Karamay, 834000, Karamay, China
E-mail address: xiaopu@cupk.edu.cn

Lingyan Bian
Department of Computer Science, China University of Petroleum-Beijing at Karamay, 834000, Karamay, China
E-mail address: lingyanbian@cupk.edu.cn

Lei Shi
Department of Computer Science, Durham University, DH1 3LE, Durham, UK
E-mail address: lei.shi@durham.ac.uk