Improving Similar Case Retrieval Ranking Performance By Revisiting RankSVM

Yuqi Liu[1] and Yan Zheng[1*]

[1*]School of Information Management for Law, China University of Political Science and Law, 27 Fuxue Road, Beijing, 102249, China.

*Corresponding author(s). E-mail(s): zhengyan@cupl.edu.cn;
Contributing authors: 211224027@cupl.edu.cn;

Abstract

Given the rapid development of Legal AI, a lot of attention has been paid to one of the most important legal AI tasks–similar case retrieval, especially with language models to use. In our paper, however, we try to improve the ranking performance of current models from the perspective of learning to rank instead of language models. Specifically, we conduct experiments using a pairwise method–RankSVM as the classifier to substitute a fully connected layer, combined with commonly used language models on similar case retrieval datasets LeCaRDv1 and LeCaRDv2. We finally come to the conclusion that RankSVM could generally help improve the retrieval performance on the LeCaRDv1 and LeCaRDv2 datasets compared with original classifiers by optimizing the precise ranking. It could also help mitigate overfitting owing to class imbalance. Our code is available in https://github.com/liuyuqi123study/RankSVM_for_SLR

Keywords: Information Retrieval, Optimization, Natural Language Processing, Legal Intelligence

1 Introduction

In recent years, Legal Artificial Intelligence (AI) has attracted attention from both AI researchers and legal professionals(Greenleaf et al (2018)). Legal AI mainly means applying artificial intelligence technology to help with legal tasks. Benefiting from the rapid development of AI, especially natural language processing (NLP) techniques, Legal AI has had a lot of achievements in real law applications (Zhong et al (2020); Shao et al (2020); Surden (2019)).

Among legal AI tasks, similar case retrieval (SCR) is a representative legal AI application, as the appeal to similar sentences for similar cases plays a pivotal role in promoting judicial fairness(Shao et al (2020)).

As demonstrated in a lot of work, there are mainly two approaches to do information retrieval. 1) Traditional IR models(e.g.BM25 which is a probabilistic retrieval model(Robertson and Walker (1994)) using keywords. 2) More advanced techniques for IR using pre-trained models with deep learning skills, and the latter has achieved promising results in some commonly used benchmark datasets(Ma et al (2021)).When using pre-trained language models, there have been a lot of variations trying to incorporate the structural information in legal documents to help with the retrieval(Hu et al (2022); Shao et al (2020); Chung et al (2014); Ma et al (2023); Zhang et al (2024); Feng et al (2022); Shao et al (2023)) or to utilize LLMs to boost similar case retrieval Zhou et al (2023).

However, compared with attention paid to language models, there is less focus on the classifier used for the final ranking. As it is pointed out in an early paper Cao et al (2006), there are two important factors in document retrieval and one of them is that to have high accuracy on top-ranked documents is crucial for an IR system. So we try to look into better classifiers in the hope of improving ranking performance. As a problem of ranking by criterion of relevance, however, SCR is reduced to a 2-class classification problem following a pointwise path in a lot of work, which means the related classifier is only used to produce one label for a query-candidate pair. It is also mentioned by other papers that Zhu et al (2022)the fine-tuned BERT model does not compare which case candidate is more similar to the query case.

In our paper, we reintroduce pairwise methods to do the final ranking. Under pairwise settings, classifiers care about the relative order between two documents, which is closer to the actual ranking task(Liu (2009)). Among pairwise approaches, we pay extra attention to the Ranking Support Vector Machine(RankSVM) method as a representative method. The advantage of RankSVM is that it aims to minimize the ranking loss and can also mitigate the negative influence of the class-imbalance issue(Cortes and Mohri (2003); Wu and Zhou (2016)) which is common for SCR tasks. It is also mentioned in some papers (Wu et al (2020)) that the setting of binary labels compared with multi labels could help mitigate the accumulated errors from thresholds learning in RankSVM.

So we are inspired to use RankSVM to explore its performance on similar case retrieval tasks and measure the performance using NDCG as our main metric (Normalized Discounted Cumulative Gain).

In our experiments, we test different kinds of retrieval models when combined with RankSVM to examine their retrieval performance on different similar case retrieval datasets LeCaRDv1(Ma et al (2021)) and LeCardv2(Li et al (2023)). Our method is illustrated in Figure 1.

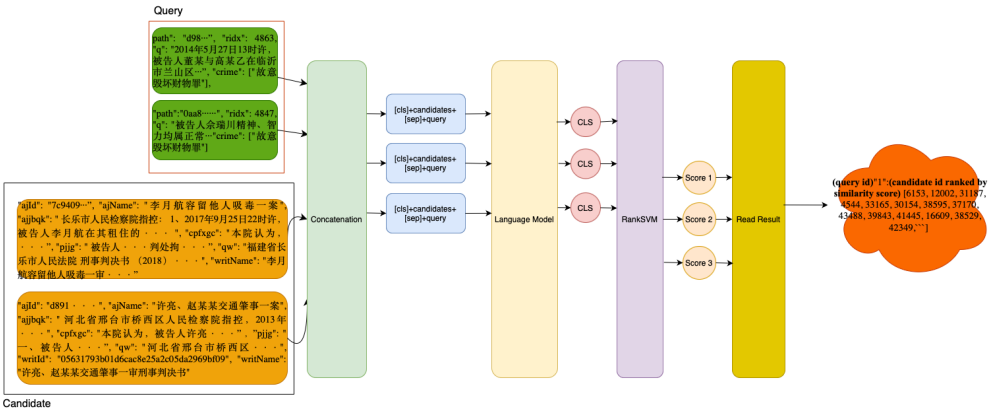


Fig. 1: Illustration of Our Method

2 Materials and Methods

2.1 Language Models for SCR

In this paper, we mainly examine three language models: BERT model, BERT+LEVEN, and Lawformer language model, which respectively represent basic language model, knowledge incorporated model, and language model with long inputs. BERT(Devlin et al (2019)) is a commonly used language model with a maximum window length of 512, and many researchers have done many implementations that have a lot to do with how they understand structural information in legal documents and concatenate them with models that could deal with limited length (Shao et al (2020); Hu et al (2022); Zhu et al (2022)). So in our experiments, we both examine the basic BERT model and the BERT model with structural information extracted from legal documents. We use the code from LEVEN(Yao et al (2022)) as an example where it incorporates event detection.

Lawformer is a representative model that people propose to deal with longer inputs (Xiao et al (2021))which is based on longformer architecture (Beltagy et al (2020)). Lawformer could process long texts up to respectively 509 for query documents and 3072 for candidate documents.

Some work(Xu (2024)) moves on to examine another new pre-trained model Mamba(Gu and Dao (2024)) that could deal with long inputs as well, where MAMBA achieves competitive performance compared with transformer-based models using the same training recipe in the document ranking task. So we also measure its performance on the LeCaRDv1 dataset.

2.2 Datasets and Data Preprocessing

In our experiments, we use two benchmark datasets that are commonly used for similar case retrieval tasks–LeCaRDv1 and LeCaRDv2. Here we mainly examine the difference between these two datasets. LeCaRDv1 dataset(A Chinese Legal Case Retrieval Dataset)(Ma et al (2021)) and LeCaRDv2 dataset(Li et al (2023)) are two datasets

specifically and widely used as benchmarks for similar case retrieval tasks since their release. We first check the difference in the number of query files and candidate files, which is shown in Table 1. While LeCaRDv1 contains 107 queries and 10,700 candidates, LeCaRDv2 contains more queries and candidates. At the same time, it is worth attention that while LeCaRDv1 has a separate candidate pool(folder) of size 100 for each query, there is no subfolder for LeCaRDv2 queries. Moreover, there lies some divergence in their judging criteria. While LeCaRDv1 focuses on the fact part of a case involving key circumstances and key elements, LeCaRDv2 argues that characterization, penalty, and procedure should all be taken into consideration. Even though they say in their paper that there is no explicit mapping function between the Overall Relevance and the sub-relevance, the calculation could be described as follows, see 1.

Table 1: Details about LeCaRDv1 and LeCaRDv2 datasets

datasets	**LeCaRDv1**	**LeCaRDv2**
#candidate cases/query	100	55192
#average relevant cases per query	10.33	20.89
ratio of relevance candidates	0.1033	0.0004

$$relevance_{v2} = relevance_{v1} + relevance_{penalty} + relevance_{procedure.} \tag{1}$$

As it is known, a legal document could usually be partitioned into 3 parts–Parties' Information, Facts, Holding, and Decision, which is illustrated in Figure 2. Here we follow the common practice to only take fact parts of those documents as inputs, see equation 2 and 3.

$$input_{v1} = (query['q'], cand['ajjbqk']) \tag{2}$$

$$input_{v2} = (query['fact'], cand['fact']) \tag{3}$$

Moreover, we examine the fact parts in both datasets to be used as our inputs, see Figure 3 and Figure 4, from which we could tell that many of them are actually longer than the input limit of the BERT model as well as the Lawformer or MAMBA model. At the same time, as we use the same number of bins in the histogram for both LeCaRDv1 and LeCaRDv2 datasets, it could be seen that while the LeCaRDv2 dataset has query files of longer lengths, it has candidate files of shorter lengths. As it is not

With different benchmark datasets, we conduct different data preprocessing. We also conduct visualization to check LeCaRDv2's length of documents. See Figure 4 to check their distribution. It could be told that case documents in LeCaRDv2 dataset are shorter .However, as there is no subfolder for LeCaRDv2 dataset in its original repository https://github.com/THUIR/LeCaRDv2, out of consideration for memory and the potential risk of probable overfitting if we conduct experiments with all negative samples among the corpus, we conduct our experiments trying to follow the practice on LeCaRDv1 dataset by building a subfolder of 130 candidates file for

Court and Case Number : People's Court of XXX Zone, Shandong Province Criminal Judgement (2019) Lu(XXXX) Criminal No.XXX

Parties' Information:

Public Prosecutor : Shandong Province XXX People's Procuratorate

Defendant Background : Defendant XXX, male, born in November 1983 in XXX, of Han ethnicity, with an incomplete junior high school education. ...

Fact:

(Accusation of the Prosecutor) On July 1, 2019, at 7:08 a.m., the defendant XXX was driving a small car with the license plate Lu N×××× along Honghai Road from north to south. While making a left turn at the intersection of Tenglu Street, he collided with a three-wheeled vehicle, also bearing the license plate Lu N×××××, which was being driven by Xu (Person 1) and carrying Xu (Person 2) as a passenger, traveling from south to north along Honghai Road...And obtained the forgiveness of the victim's immediate family members.

(Evidence) Regarding the above-mentioned facts, the prosecution has provided the following evidence:1.Documentary evidence, including the Road Traffic Accident Determination Report and the arrest records.2. Testimony from the witness Xu (Person 2). 3. The defendant's statements and defense arguments. ...

(Arguments of Both Sides) The prosecution believes that the defendant, XXX, violated traffic management regulations, caused a major traffic accident resulting in one death, and bears primary responsibility for the accident. …The defendant, Zhang Baoyu, has no objections to the criminal facts and charges brought by the prosecution and has admitted guilt and accepted punishment. ...

The defense counsel presented the following argument: no objections are raised regarding the charge of traffic accident crime against the defendant Zhang Baoyu as stated in the indictment. ...

(Facts Confirmed by the Court) After trial and investigation, it was found that on July 1, 2019, at 7:08 a.m., the defendant XXX was driving...The above facts are substantiated by the following evidence submitted by the prosecution and verified through cross-examination and authentication in court:1.Documentary evidence:(1) One household registration certificate and one certificate of no criminal record. These confirm that XXX was born on November 23, 1983, and had reached the full age of criminal responsibility at the time of the accident.A query on the Shandong Public Security Cloud Computing Platform, as of August 6, 2019, revealed no prior criminal record for XXX. ...

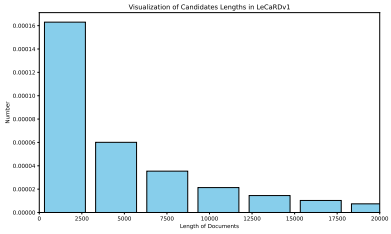
Holding:

This court holds that the defendant, Zhang Baoyu, violated traffic management regulations and caused a major accident...A lighter punishment may be imposed at the court's discretion. ...

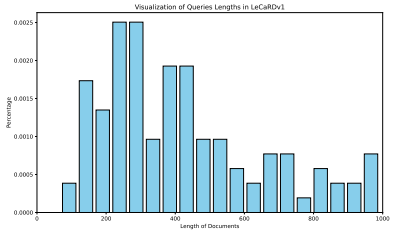
Decision:

The judgment is as follows: Defendant Zhang Baoyu is found guilty of the crime of causing a traffic accident and is sentenced to six months of fixed-term imprisonment. ...

Fig. 2: Illustration of Structure of A Case



(a) Lengths of Candidates in LeCaRDv1 Dataset

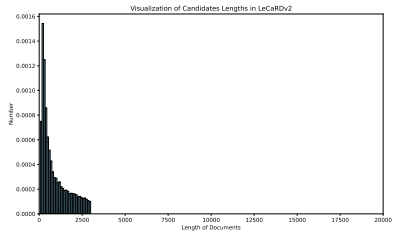


(b) Lengths of Queries in LeCaRDv1 Dataset

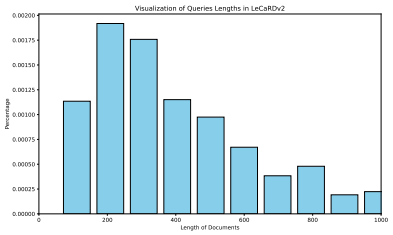
Fig. 3: Length of Documents in LeCaRDv1 Dataset

each query, among which 30 are those that are labeled relevant which is also similar with the practice taken in LeCaRDv2 baseline where they have negative samples with the ratio of positives and negatives at 1:32. It is shown by our result that it is a wise choice. We also follow different train-test splits. For LeCaRDv1, we follow the 5-fold validation while for LeCaRDv2 we took 640 instances as our training dataset and 160 as our test dataset.

At the same time, the data preprocessing does differ from model to model. For Lawformer, we basically follow the practice in its paper to have candidate files of length 3072 and query files of length 509. For LEVEN code, what we do is we detect their queries and candidate files to get their trigger words and relevant event types. According to the original paper, there are As a result, for non-trigger tokens, it just



(a) Lengths of Candidates in LeCaRDv2 Dataset



(b) Lengths of Queries in LeCaRDv2 Dataset

Fig. 4: Length of Documents in LeCaRDv2 Dataset

feeds the sum of token embeddings and position embeddings into the BERT model. For trigger words, we define an event type embedding for each event type and add the corresponding embedding to each input.

$$inputx_{without_event} = func_{tokentoids}('[CLS]' + query + '[SEP]' + cand + '[SEP]') \quad (4)$$

$$inputx_{use_event} = id_{CLS} + id_{query} + id_{sep} + id_{candidate} + id_{sep} \quad (5)$$

$$id_{events} = [0] + id_{query_event} + [0] + id_{cand_event} + [0] \quad (6)$$

2.3 Fine-tuning with Pretrained Model And Feature Extraction

When we do fine-tuning, we basically follow the practice taken in LEVEN code https://github.com/thunlp/LEVEN. For BERT-related models, we add a two-class classification layer after the pooled output of BERT denoted as [CLS] and do fine-tuning with the training set.

For Lawformer, the concrete steps are basically the same as BERT. In terms of hyperparameters, we basically follow what is used in LEVEN code since according to our experiments those are better than the original Lawformer code when applied to a smaller batch size. The hyperparameters used are shown in Table 2.

$$output = func_{twoclassclassification}([CLS]) \quad (7)$$

To extract features, we just use the output of CLS of every query-candidate pair as our features. According to the dimension of CLS, we have 648 features for each query-candidate pair. For further experiments, we only pick the models with the best performance on the validation dataset or test dataset for feature extraction. In the LeCaRDv1 dataset,when training on each fold,we iterate the training dataset for 5 epochs and choose the epoch with the best performance compared with other epochs on the respective validation dataset. After training and validating on each fold, we report the average value for the 5 folds. In the LeCaRDv2 dataset, we iterate the training dataset for 5 epochs and choose the epoch with the best performance on the test dataset to extract features. Our model is only trained from the training data.

The result of RankSVM is used to rank the similarity of candidate-query pairs under the same candidate files.

$$features = [CLS] \tag{8}$$

2.4 RankSVM

Since RankSVM is a relatively classic method developed from ordinal-regression SVM to do information retrieval(Joachims (1998); Jakkula (2006)), we try to apply RankSVM(Joachims (1998); Zheng et al (2019)) for similar case retrieval task in this work. The mathematical formulation for RankSVM is shown below. Given n training queries $q_i{}_{i=1}^{n}$, their associated document pairs $(x_u^{(i)}, x_v^{(i)})$ and the corresponding ground truth label $y_{u,v}^{(i)}$, where a linear scoring function is used without complicated kernel, i.e., $f(x) = w^T x$.

The basic form of RankSVM is shown as follows.

$$min\frac{1}{2}|||w|||^2 + C\sum_{i=1}^{n}\sum_{u,v:y_{u,v}^{(i)}=1}\xi_{u,v}^{(i)} \tag{9}$$

$$s.t.w^T(x_u^{(i)} - x_v^{(i)}) \geq 1 - \xi_{u,v}^{(i)}, if y_{u,v}^{(i)} = 1 \tag{10}$$

,

$$\xi_{u,v}^{(i)} \geq 0, i = 1, \ldots, n. \tag{11}$$

In our experiment, we mainly use the method developed in the paper by Joachims(Joachims (1998)), where the algorithm just has 1 slack variable and it is solved using its dual form. Different algorithms for RankSVM are suitable for different applications. According to our experiment, the result with the algorithm we chose is better than other settings.(e.g., see this work(Joachims (2005)) Also, we use different values of C, and they are 0.001, 0.05, 0.01, 0.02, 0.05, 0.1, 0.5, 1. We report the best result for each model.

It is worth mentioning that here we don't need a threshold and we just get results in scores and do ranking. Under the setting of predicting with a threshold, as it is said, there will be an implicit presumption that when in training and test, its input has the same data distribution and accumulated error(Wu et al (2020)) and may impede model performance.

According to the original LeCaRDv1 and LeCaRDv2 paper, there are 4 classes of relevance. Here we just process it as a two-class classification. When processing labels, as long as two cases are relevant, we view the label of the pair as positive.

3 Result

3.1 Results on LeCaRDv1

To test the effectiveness of our method, we conduct our experiments mainly using BERT, BERT+Event, BERT+Event+RankSVM, BERT+RankSVM, Lawformer, Lawformer+RankSVM. The result is shown in Table 3.

Table 2: Parameters of Different Models

Model	BERT-Based	Lawformer	MAMBA
Algorithm	1-slack algorithm (dual)	1-slack algorithm (dual)	1-slack algorithm (dual)
Norm	l1-norm	l1-norm	l1-norm
Learning Rate	1e-5	2.5e-6(-1)/5e-6(v2)	1e-5
Optimizer	Adamw	Adamw	Adamw
Training batch size	16	2(v1)/4(v2)	16
Evaluating batch size	32	32	32
Step size	1	1	1

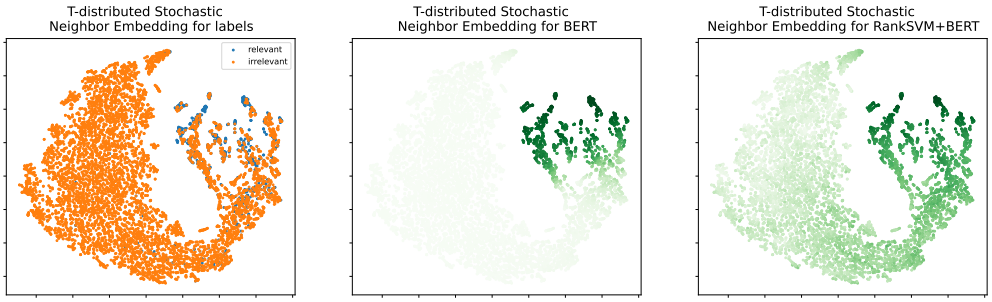
From the result on BERT, we could see that RankSVM helps improve performance especially on NDCG, which proves that our model is better at putting highly relevant cases earlier in a similar case list. To better understand the advantage of RankSVM, we finish visualization based on the training and testing dataset on fold 0 to see the score computed by original BERT and BERT+RankSVM. Specifically, we use features extracted by BERT as our input and compress them into two dimensions using t-SNE(van der Maaten and Hinton (2008)). After that, we color them according to similarity scores computed by relative models or labels. See Figure 5 and 6 for reference.

Table 3: Results on LeCaRDv1 Dataset

Model	NDCG@10	NDCG@20	NDCG@30
BERT	0.7896	0.8389	0.9113
BERT+RankSVM	**0.7963**	**0.8504**	**0.9166**
BERT+LEVEN	0.7881	0.8435	0.9136
BERT+LEVEN+RankSVM	**0.7931**	**0.8452**	**0.9164**
MAMBA	0.7469	0.8013	0.892
Lawformer	0.7592	0.8169	0.8993
Lawformer+RankSVM	**0.7738**	**0.8258**	**0.9073**

From the visualization result, we could see that RankSVM performs better by differentiating the relevance level of each pair continuously, while BERT treats the relevance level more discretely.

What's more, the mechanism of RankSVM is actually maximizing ROC(Receiver Operating Characteristic curve)(Ataman (2005)). So we use the model of BERT comparing the results with and without RankSVM to show how ROC changes. For the BERT model, we use the first fold and the C is 0.01, and the visualization result is shown in Figure 7. It could be seen that RankSVM helps improve the result of BERT on the AUC score, and it does an even better job in the test dataset. It proves our guess before that it could help reduce overfitting. Additionally, to explore SVM's

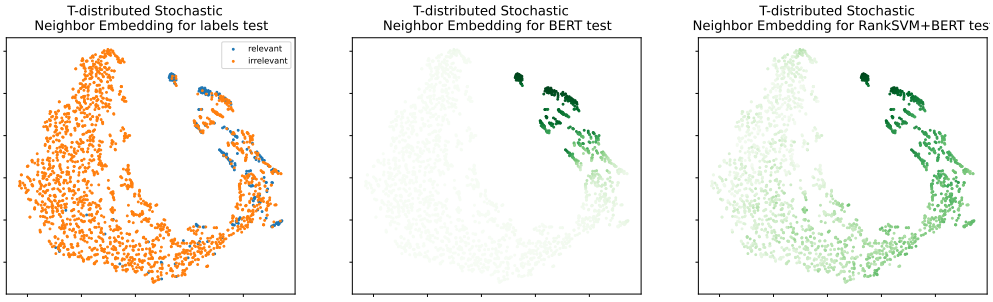


(a) Visualization of Relevance Labels

(b) Visualization of BERT Features

(c) Visualization of RankSVM Scores

Fig. 5: Visualization of Labels/Features/Scores on LeCaRDv1 Training Dataset



(a) Visualization of Relevance Labels

(b) Visualization of BERT Features

(c) Visualization of RankSVM

Fig. 6: Visualization of Labels/Features/Scores on LeCaRDv1 Test Dataset

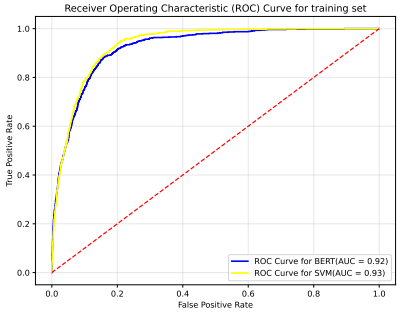
performance on longer text for retrieval, we also benchmark its performance using Lawformer. It could also be seen that all NDCG-related metrics get improved.

We also conduct experiments using MAMBA. As the performance is not competent enough, we don't do further combinations with RankSVM.

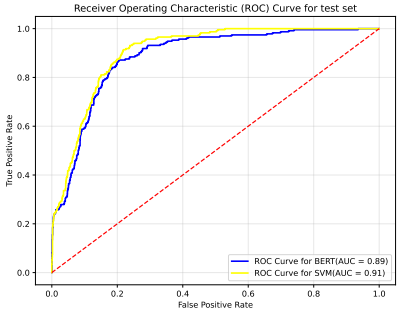
3.2 Results on LeCaRDv2

The result is shown in Table 4.According to the table, we could find that the NDCG related metrics are generally improved similar to LeCaRDv1, except for the BERT model. To understand its performance, here we also do visualization using features extracted from the BERT model following the same steps as the LeCaRDv1 dataset, see Figure 8 and 9.

It is shown by the figure with real labels that BERT features fail to distinguish these two classes explicitly this time as there are a lot of overlaps between the two classes compared with what is extracted from the LeCaRDv1 dataset, while RankSVM tries to learn from those features that are not distinguished between these two classes



(a) ROC Curve for LeCaRDv1 Training Dataset



(b) ROC Curve for LeCaRDv1 Test Dataset

Fig. 7: Three simple graphs

Table 4: Results on LeCaRDv2 Dataset

Model	NDCG@10	NDCG@20	NDCG@30
BERT	**0.8351**	**0.8764**	**0.9348**
BERT+LEVEN	0.8117	0.8455	0.9184
BERT+RankSVM	0.793	0.8582	0.9247
BERT+LEVEN+RankSVM	**0.8078**	0.8548	0.9227
Lawformer	0.7968	0.8461	0.919
Lawformer+RankSVM	**0.812**	0.8574	0.9247

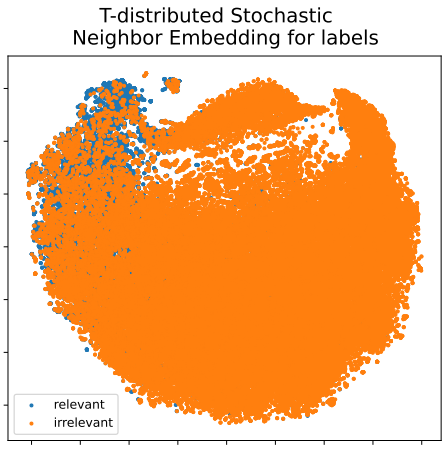
continuously. The failure of BERT may be owing to the changed standards of relevance in LeCaRDv2 and the shorter document length of the LeCaRDv2 dataset. It explains why RankSVM added to BERT fails to help improve NDCG metrics here.

4 Discussion

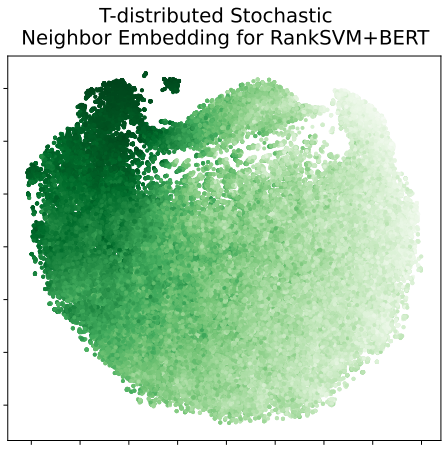
4.1 Impact of Length of Text

Even after we changed the length of the input data from 512 to 800 using the MAMBA model and Lawformer model, we could see that the interesting thing here is that there is no apparent increase in metrics compared with BERT, which is also mentioned in other works(Deng et al (2024)).

But we could also argue that there lies some space for hyperparameter tuning in the future.

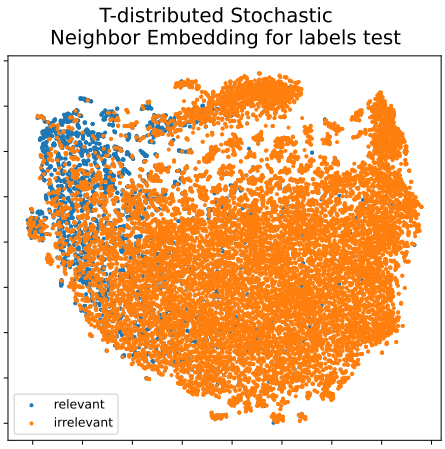


(a) Visualization of Relevance Labels

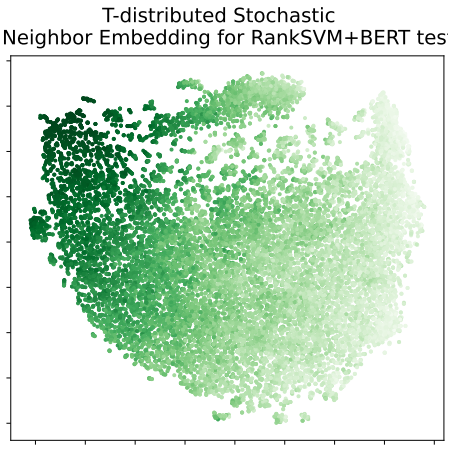


(b) Visualization of RankSVM Scores

Fig. 8: Visualization of Labels/Scores on LeCaRDv2 Training Dataset



(a) Visualization of Relevance Labels



(b) Visualization of RankSVM Scores

Fig. 9: Visualization of Labels/Score on LeCaRDv2 Test Dataset

4.2 Error Analysis

To explore further why RankSVM fails to improve the performance on BERT-based models regarding the LeCaRDv2 dataset, we conduct experiments using multi-class BERT classification to extract features out of our speculation that RankSVM on LeCaRDv2 needs language models that provide more information and end up getting all NDCG-related metrics improved again, see table 5. So our speculation holds as RankSVM still works for BERT on the LeCaRDv2 dataset under this setting.

Recently, there have been more research on improving RankSVM, and we leave that to explore in the future.

Table 5: Result on LeCaRDv2 when we do multi-class classification

Model	NDCG@10	NDCG@20	NDCG@30
BERT Multiclass	0.791	0.8443	0.9181
BERT Multiclass+RankSVM	0.8053	0.8352	0.9222

5 Conclusion

In our paper, we examine the performance of RankSVM when combined with other language models using binary labels to serve a similar case retrieval task on the LeCardv1 dataset and the LeCardv2 dataset. We come to the conclusion that RankSVM with binary labels could help improve the performance of models by improving their concrete ranks, and we also try to give some explanation from the feature perspective. Also, the RankSVM could help fight overfitting, which results from an imbalance class common in this task. Our findings point to the potential of improving SCR task performance from an optimization perspective. There still lies some work that needs to be done to explore the appropriate model to improve performance on the recall metric on the LeCaRDv2 dataset and how the performance of RankSVM will change when we extract features with larger batch sizes. We also leave the investigation regarding RankSVM combined with other models and datasets for future work.

References

Ataman K (2005) Optimizing area under the roc curve using ranking svms. URL https://api.semanticscholar.org/CorpusID:18159506

Beltagy I, Peters ME, Cohan A (2020) Longformer: The long-document transformer. CoRR abs/2004.05150. URL https://arxiv.org/abs/2004.05150, 2004.05150

Cao Y, Xu J, Liu TY, et al (2006) Adapting ranking svm to document retrieval. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Association for Computing Machinery, New York, NY, USA, SIGIR '06, p 186–193, https://doi.org/10.1145/1148170.1148205, URL https://doi.org/10.1145/1148170.1148205

Chung J, Gülçehre Ç, Cho K, et al (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. CoRR abs/1412.3555. URL http://arxiv.org/abs/1412.3555, 1412.3555

Cortes C, Mohri M (2003) Auc optimization vs. error rate minimization. In: Proceedings of the 16th International Conference on Neural Information Processing Systems. MIT Press, Cambridge, MA, USA, NIPS'03, p 313–320

Deng C, Mao K, Dou Z (2024) Learning interpretable legal case retrieval via knowledge-guided case reformulation. arXiv preprint arXiv:240619760

Devlin J, Chang MW, Lee K, et al (2019) Bert: Pre-training of deep bidirectional transformers for language understanding. URL https://arxiv.org/abs/1810.04805, 1810.04805

Feng Y, Li C, Ng V (2022) Legal judgment prediction via event extraction with constraints. In: Muresan S, Nakov P, Villavicencio A (eds) Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Dublin, Ireland, pp 648–664, https://doi.org/10.18653/v1/2022.acl-long.48, URL https://aclanthology.org/2022.acl-long.48

Greenleaf G, Mowbray A, Chung P (2018) Building sustainable free legal advisory systems: Experiences from the history of ai & law. Computer Law & Security Review 34(2):314–326

Gu A, Dao T (2024) Mamba: Linear-time sequence modeling with selective state spaces. URL https://arxiv.org/abs/2312.00752, 2312.00752

Hu W, Zhao S, Zhao Q, et al (2022) Bert_lf: A similar case retrieval method based on legal facts. Wireless Communications and Mobile Computing 2022(1):2511147

Jakkula V (2006) Tutorial on support vector machine (svm). School of EECS, Washington State University 37(2.5):3

Joachims T (1998) Making large scale svm learning practical. Technical reports URL https://api.semanticscholar.org/CorpusID:61116019

Joachims T (2005) A support vector method for multivariate performance measures. In: Proceedings of the 22nd International Conference on Machine Learning. Association for Computing Machinery, New York, NY, USA, ICML '05, p 377–384, https://doi.org/10.1145/1102351.1102399, URL https://doi.org/10.1145/1102351.1102399

Li H, Shao Y, Wu Y, et al (2023) Lecardv2: A large-scale chinese legal case retrieval dataset. URL https://arxiv.org/abs/2310.17609, 2310.17609

Liu TY (2009) Learning to rank for information retrieval. Found Trends Inf Retr 3(3):225–331. https://doi.org/10.1561/1500000016, URL https://doi.org/10.1561/1500000016

Ma Y, Shao Y, Wu Y, et al (2021) Lecard: A legal case retrieval dataset for chinese law system. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 2342–2348

Ma Y, Wu Y, Ai Q, et al (2023) Incorporating structural information into legal case retrieval. ACM Trans Inf Syst 42(2). https://doi.org/10.1145/3609796, URL https://doi.org/10.1145/3609796

van der Maaten L, Hinton G (2008) Visualizing data using t-sne. Journal of Machine Learning Research 9(86):2579–2605. URL http://jmlr.org/papers/v9/vandermaaten08a.html

Robertson SE, Walker S (1994) Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In: Croft BW, van Rijsbergen CJ (eds) SIGIR '94. Springer London, London, pp 232–241

Shao Y, Mao J, Liu Y, et al (2020) Bert-pli: Modeling paragraph-level interactions for legal case retrieval. In: International Joint Conference on Artificial Intelligence,

URL https://api.semanticscholar.org/CorpusID:267909336
Shao Y, Wu Y, Liu Y, et al (2023) Understanding relevance judgments in legal case retrieval. ACM Trans Inf Syst 41(3). https://doi.org/10.1145/3569929, URL https://doi.org/10.1145/3569929
Surden H (2019) Artificial intelligence and law: An overview. Georgia State University Law Review 35(4)
Wu G, Zheng R, Tian Y, et al (2020) Joint ranking svm and binary relevance with robust low-rank learning for multi-label classification. Neural Networks 122:24–39. https://doi.org/https://doi.org/10.1016/j.neunet.2019.10.002, URL https://www.sciencedirect.com/science/article/pii/S0893608019303247
Wu X, Zhou Z (2016) A unified view of multi-label performance measures. CoRR abs/1609.00288. URL http://arxiv.org/abs/1609.00288, 1609.00288
Xiao C, Hu X, Liu Z, et al (2021) Lawformer: A pre-trained language model for chinese legal long documents. CoRR abs/2105.03887. URL https://arxiv.org/abs/2105.03887, 2105.03887
Xu Z (2024) Rankmamba: Benchmarking mamba's document ranking performance in the era of transformers. URL https://arxiv.org/abs/2403.18276, 2403.18276
Yao F, Xiao C, Wang X, et al (2022) LEVEN: A large-scale Chinese legal event detection dataset. In: Findings of the Association for Computational Linguistics: ACL 2022, pp 183–201, https://doi.org/10.18653/v1/2022.findings-acl.17, URL https://aclanthology.org/2022.findings-acl.17
Zhang Y, Zhai S, Meng Y, et al (2024) Event is more valuable than you think: Improving the similar legal case retrieval via event knowledge. Inf Process Manage 61(4). https://doi.org/10.1016/j.ipm.2024.103729, URL https://doi.org/10.1016/j.ipm.2024.103729
Zheng Y, Zheng Y, Ohyama W, et al (2019) Ranksvm for offline signature verification. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp 928–933, https://doi.org/10.1109/ICDAR.2019.00153
Zhong H, Xiao C, Tu C, et al (2020) How does NLP benefit legal system: A summary of legal artificial intelligence. In: Jurafsky D, Chai J, Schluter N, et al (eds) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Online, pp 5218–5230, https://doi.org/10.18653/v1/2020.acl-main.466, URL https://aclanthology.org/2020.acl-main.466
Zhou Y, Huang H, Wu Z (2023) Boosting legal case retrieval by query content selection with large language models. URL https://arxiv.org/abs/2312.03494, 2312.03494
Zhu J, Luo X, Wu J (2022) A bert-based two-stage ranking method for legal case retrieval. In: Memmi G, Yang B, Kong L, et al (eds) Knowledge Science, Engineering and Management. Springer International Publishing, Cham, pp 534–546