A Comparative Study of BERT and Legal-BERT for the Prediction of Indian Legal Case Judgements

Anukriti Bansal[1]*, Aman Jain[1]*, Himanshu Goyal[1], and Vikas Bajpai[1]

The LNM Institute of Information Technology, Jaipur, Rajasthan, India
anukriti.bansal@lnmiit.ac.in, amanj3335@gmail.com,
himanshu1234goyal@gmail.com, vikas.bajpai87@gmail.com

Abstract. Transformer-based models have gained immense popularity in the past few years and many state-of-the-art algorithms in natural image processing are based on transformers. BERT(Bidirectional Encoder Representation from Transformer) is also an attention based model which works very well on text classification. In order to use these models in Indian judiciary, incorporating domain knowledge is necessary. Legal-BERT was developed using the domain knowledge of US and EU legal documents. We obtain text embedding from both these models and use them for the prediction of Indian legal case judgements using several machine learning and deep learning algorithms. Both BERT and Legal-BERT have an input size limitation of 512 tokens. In this paper, we propose a solution for the limited token size of BERT models and perform a thorough comparison of BERT and Legal-BERT models to check their effectiveness in the Indian legal system. In all the algorithms, it has been observed that Legal-BERT outperforms basic BERT model and can be used in Indian legal context as well. The source code of the current work is made publicly available at https://github.com/stelios357/Predicting-Indian-Legal-Case-Judgements.

Keywords: BERT · Legal-BERT · Transformer model · Legal Judgement Prediction · Deep Learning

1 Introduction

Prediction of the judgement of legal court cases may help in the understanding of the judicial decision-making process. It can also assist lawyers and judges by suggesting them the possible outcome of any court case. This system will be very useful in countries like India where almost 27 millions cases are pending in courts, many of which are there for more than ten years [26, 29]. One of the reasons for such a huge backlog is very low judges to citizens ratio. In order to speed up the disposal of cases, intelligent and efficient systems for the prediction of outcomes, searching for similar cases from the past, and summarization of legal documents can be very helpful.

* The authors have contributed equally.

Transformer-based language models are currently the state-of-the-art models for several natural language processing (NLP) tasks such as text classification, summarization, translation, question answering, etc [28] [21, 22]. Obtaining the text embedding is one of the most fundamental and significant steps in NLP tasks. BERT (Bidirectional Encoder Representation from Transformer) is very popular and effective context-based embedding model [13]. The embeddings obtained from BERT can be used as input to various machine learning and deep learning-based algorithms for various NLP tasks. The publicly available BERT models have been pre-trained mainly using Wikipedia pages and various books, therefore, they may fail to give good performance when used in specialized domains such as biomedical, clinical, scientific text, or legal [3,8,18,20]. Chalkidis et al [8] introduced a BERT model for legal domain and named it as Legal-BERT. They trained the model from scratch using a different set of hyper-parameters as compared to original BERT proposed by Devlin et al [13], on the EU and US legal text documents. Their model performed better on legal text classification and sequence tagging.

BERT and Legal-BERT both were trained on the datasets which are different from the documents of Indian judicial system. Additionally, both have the input size limitation of 512 tokens, which is very less in case of long legal case documents. To exploit the advantages of already trained language models, in this paper, we provide a unique way to handle the limitation of token size of these models and compare them for the prediction of judgements based on Indian legal documents. There are arguments and pleas in legal documents and the task is to predict the court's decision that whether the plea will be accepted or rejected. In this paper we use bert-base-uncased[1] and legal-bert-uncased[2] models and compare their text-embeddings to predict court's judgement, using various machine learning and deep learning algorithms.

First the complete legal document is tokenized into words and special tokens ([CLS] token at the starting of the text and [SEP] token at the end of each sentence) are added. An important constraint with the BERT and Legal-BERT is that they accept only 512 tokens as input. Malik et al [24] in their work have mentioned that for majority of Indian court files, last 512 tokens are sufficient for the prediction of judgement. We divide the whole legal document in the chunks of 512 tokens and use the last 5 chunks for our experiments. Text embeddings (vector of length 768) of each chunk is obtained as the output of BERT and Legal-BERT. These embeddings are given in different ways to various machine learning and deep learning model to predict the Indian court judgement that whether it will be accepted or rejected. From our experiments, we have observed that although Legal-BERT was trained on EU and US legal documents, but it perform well on Indian legal texts as well. Our research will help in choosing a better model for further research in Indian legal domain for various other NLP tasks as well.

[1] https://huggingface.co/bert-base-uncased
[2] https://huggingface.co/nlpaueb/legal-bert-base-uncased

2 Literature review

Several authors are working on finding solutions for different problems in the legal domain, such as measuring similarity between documents [25], document summarization [4], and legal judgement prediction [5,32]. Here, we review some of the recent work on legal judgement prediction using classical machine learning and deep learning algorithms

2.1 Machine learning based prediction methods

Shaikh et al [27] have proposed a method to predict the outcome of murder related cases as Acquittal or Conviction. They have manually identified certain keywords and used used classical machine learning algorithms for the prediction. Lage-Freitas et al [19] used TF-IDF and machine learning-based algorithm for the prediction of Brazilian court decisions. In another approach, linear SVM was used by [2] to forecast the violations from facts in Human Rights cases in European Courts.

2.2 Deep learning based prediction methods

Chaphalkar et al [9] used multi-layer perceptron to predict the outcome of construction dispute claims. They determined the intrinsic factors for construction disputes related claims through case study approach and used MLP for the prediction of the outcome. Chalkidis et al [7] evaluated a variety of neural models for judgement prediction and proposed hierarchical BERT to surpass the limitation of token length of BERT. On CAIL2018 [30] dataset, Yang et al. [31] made an attempt to predict similar law articles and relevant information by using multi-perspective bi-feedback network. In another work to predict relevant similar law articles and the charges, Luo et al. [23] proposed an attention based model, which was trained and tested on Criminal Law dataset of China. For predicting the prison term, Chen et al. [10] used deep gating network.

2.3 Major contributions

Following are the major contributions of this paper:

1. We propose various novel ways to represent text encoding from the BERT model that also deals with the issue of limited token length of BERT. As observed in the results, the proposed feature representations give better results as compared to using 512 tokens of BERT and Legal-BERT.
2. A thorough comparison of BERT and Legal BERT is presented using various machine learning and deep learning architectures.
3. The comparison results can be used in further research in the use of transformer models in Indian legal domain.

The organization of the paper is as follows. In Section 3 we explain the proposed feature representation methods and various machine learning and deep learning algorithms that are used to predict court judgments. Details of the dataset and experimental evaluations of various algorithms are given in Section 4. Finally we conclude the paper in Section 5.

3 Proposed method for legal case decision prediction

3.1 Overview

In this work, we propose various novel ways to use BERT (and Legal-BERT) text embeddings as an input to various machine learning and deep learning algorithms for the prediction of legal case decision (whether the case will be accepted or rejected). First, we convert the legal text into a sequence of tokens (words), and this process is called tokenisation. After tokenisation, we add special tokens and send it as input to BERT/Legal-BERT. [CLS] token is added at the starting of the input text and [SEP] token is added at the end of each sentence of the text. BERT model then output an embedding vector of size 768 for each of the tokens. We use the embeddings of [CLS] token to represent the complete text.

As mentioned in Section 1, a major constraint with the BERT is that it only accepts 512 tokens as input. Malik et al [24] claimed that in most of the Indian legal case documents, text embedding of the last 512 tokens is sufficient for the prediction of the court's decision. To see the effect of capturing more information from the legal documents in the performance of machine learning and deep learning-based prediction algorithms, we divide the documents in the chunks of 510 tokens. We add [CLS] and [SEP] tokens at the starting and at the end of each chunk respectively. We use last 5 chunks of the document and use each of them as an input to the BERT. At the end, we have five vectors each of length 768 ([CLS] representations). Since length and the quality of input plays an important role in the performance of any machine learning and deep learning algorithm, we propose three ways to represent the information from these five vectors. These are explained in Section 3.2 Feature representation.

Next we explain various machine learning and deep learning models which are used for the prediction using the text embeddings from BERT and Legal-BERT. In addition to these models, we also use BERT/Legal-BERT as classifier.

3.2 Feature representation

We use four different feature representations for our experiments. In the first case we give only the last 512 tokens of legal case document as an input to BERT (and Legal-BERT), which give us the text embeddings in form of a vector of length 768. This vector acts as an input to different ML and DL algorithms. The features are represented as $bert_{512}$ and $legalBert_{512}$. The details of remaining three representations are given as follows:

Sequential representation We obtain text embeddings from the last 5 chunks (512 tokens each) of legal documents which give us five vectors of length 768 each. In this representation, we concatenate these five vectors to create a vector of length 3840. $bert_{seq}$ and $legalBert_{seq}$ are used to represent sequential features.

Averaging The length is the feature vector obtained after sequential representation is very large. Reducing the feature length may give better performance with machine learning algorithms. Inspired from the global average pooling layer of convolutional neural networks, here we take the average value of five vectors, which results in a vector of length 768. We represent these features as $bert_{avg}$ and $legalBert_{avg}$

Maximum In this case, the idea came from the max-pooling layer of convolutional neural networks. The maximum value at the same index of five vectors is chosen, which results in a 768 sized vector. The features are represented as $bert_{max}$ and $legalBert_{max}$

3.3 Classical machine learning models

The four types of feature representations mentioned in the above section are used as input to the various machine learning algorithms: Adaboost [14], gradient boosting [6, 15], logistic regression, random forest classifier [17], and support vector machine (SVM) [12].

3.4 Deep learning models

Classical deep learning models We experiment with two different deep learning models for the prediction of the judgement of Indian legal case documents. The first one is a multi-layer perceptron model consisting of five fully connected layers. Intermediate dropout layers are used to prevent the over-fitting. Another model we use for legal judgement prediction is the classifier that come along with the pre-trained BERT and Legal-BERT.

Sequential deep learning models The classical models mentioned above do not capture the sequence information of these five vectors or chunks ([CLS] representations of length 768 each). To preserve the sequence information, these vectors are used as input to sequence-based models. In this paper we have experimented with LSTM, GRU, Bidirectional LSTM (BiLSTM), and Bidirectional GRU (BiGRU). The network architecture for all these models is the same and consists of three LSTM/BiLSTM/GRU/BiGRU cells of sizes 100, 50, 20 respectively. To prevent over-fitting, we use intermediate dropout layers at the rate of 0.25, 0.20, and 0.1. We trained the models for with early stopping criteria and batch size 16.

4 Experimental results and discussion

All the experimental programs are coded using Keras [11] API of TensorFlow framework [1,16] . The hardware setup includes Nvidia DGX-1 server with Dual 20-core intel Xeon e5-2698 v4 2.2 Ghz processor, supported by 512 GB DDR4 RAM and 256 GB (8x32) GPU memory.

4.1 Dataset description

Our dataset consists of legal case files of Supreme Court of India that were scrapped from the website of indiankanoon.org [3]. Each case file is the entire case preceding of the Supreme Court of India in the English language. The data has been scrapped for the period 1947 to April 2020. These documents are very verbose and have a lot of grammatical and spelling errors since they are typed when a preceding is going on. Hence the pre-processing and the cleaning is done accordingly. Given a case, the appeal can either be accepted or rejected. The final decision of whether the plea has been accepted or rejected is found towards the end of the case preceding. This information about the final decision has been extracted and used to label the data: 1 for accepted and 0 for rejected. In certain case documents, there can be multiple pleas. In such scenario, if there is a single decision for all the pleas, the case file is annotated accordingly, otherwise, even if a single plea is accepted, we label the document as accepted. There are 42409 legal cases files in our dataset. Out of a total of 42409 case proceedings, the plea has been accepted in 17838 cases which makes 42.06% of the dataset, and in 24571 cases, the plea has been rejected, which makes 57.93% of the dataset. The cases in which the decision was neutral or which had no decisions are ignored and are not part of the dataset. To design algorithms for the prediction of court case judgement, i.e., whether the plea will be accepted or rejected, we split the dataset in training, validation, and test in the ratio 70:18:12.

4.2 Evaluation metrics

To evaluate the overall performance of different models in the prediction of legal case judgement, we use binary cross entropy as the loss function and F1 score as the basic evaluation metric.

Binary cross entropy The loss function is given as follows:

$$Log\ loss = \frac{1}{N}\sum_{i=1}^{N} -(y_i * log(p_i) + (1-y_i) * log(1-p_i)) \tag{1}$$

Here, p_i is the probability of class 1, and $(1-p_i)$ is the probability of class 0.

[3] https://indiankanoon.org/

F1 score/F measure This measure helps in determining the correctness of the prediction. It requires the computation of precision and recall metrics. The precision (P) suggests how well the model is able to correctly predict the judgements (positive or negative). Recall (R), on the other hand determines how many predictions we correctly made while penalizing for the incorrect predictions. F1 score is computed as follows:

$$F1 = \frac{2.R.P}{R+P} \tag{2}$$

4.3 Results and Discussion

Prediction results using classical machine learning models As mentioned in Section 3.3, we have experimented with five classical machine learning models. We have four different types of feature representations (Section 3.2) and embeddings from BERT and Legal-BERT. Therefore, for each machine learning model there are eight results as shown in Table 1. In all the cases, Legal-BERT embeddings clearly outperforms the results using BERT-base embeddings. It can also be observed that a few machine learning models performs well for features obtained using last 512 tokens while others give better results on the proposed feature representations that consider more number of tokens (or chunks).

Prediction results using classical deep learning models The prediction results on different feature representations using MLP and BERT and Legal-BERT as classifiers is shown in Table 2. Here as well Legal-BERT performs better than BERT-Base for both deep learning models. Additionally, it can be observed that as we increase the number of tokens (five chunks of 512 tokens), results improve significantly. All three proposed feature representations (Section 3.2) give better results as compared to features obtained using 512 tokens. This show the applicability of our method in various NLP applications.

Prediction results using sequential deep learning models As mentioned in Section 3.4, we have experimented with four sequential deep learning algorithms. As can be seen from the Table 3, in all four cases text-embeddings obtained from Legal-BERT works better in prediction of judgements for Indian legal documents.

5 Conclusion

In this paper we proposed a solution for input token size limitation of BERT/Legal-BERT models and showed that the proposed method helps in capturing more contextual information and therefore gives better prediction accuracy. This can be used for various other NLP applications as well. BERT and Legal-BERT both were trained on the documents that are different from Indian legal texts.

Table 1. Prediction results using classical machine learning models

Feature Representation	Train set				Test set			
	Precision (%)	**Recall (%)**	**F1 (%)**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 (%)**	**Accuracy (%)**
Support Vector Machine								
$bert_{512}$	66	66	65	65	63	63	63	63
$bert_{seq}$	71	72	70	70.37	55	63	59	62.17
$bert_{avg}$	63	63	63	63.14	61	62	61	61.51
$bert_{max}$	65	65	65	64.80	62	62	62	62.04
$\boldsymbol{legalBert_{512}}$	**72**	**72**	**72**	**72.22**	**67**	**68**	**67**	**67.75**
$legalBert_{seq}$	73	73	73	72.7	64	64	64	63.44
$legalBert_{avg}$	67	67	67	67.07	66	66	66	66.25
$legalBert_{max}$	70	71	70	70.55	66	66	66	66.22
Logistic Regression								
$bert_{512}$	63	63	63	63.16	62	62	62	62.26
$bert_{seq}$	68	68	68	67.79	64	64	64	63.83
$bert_{avg}$	63	63	62	62.74	61	61	61	61.18
$bert_{max}$	61	62	61	61.60	61	61	60	60.62
$legalBert_{512}$	67	68	67	67.89	61	61	60	60.62
$\boldsymbol{legalBert_{seq}}$	**74**	**74**	**74**	**74.08**	**68**	**69**	**68**	**68.74**
$legalBert_{avg}$	67	67	67	67.39	66	66	66	66.07
$legalBert_{max}$	64	65	64	64.74	64	64	64	64.06
Random Forest								
$bert_{512}$	66	66	65	65.94	60	60	60	60.54
$bert_{seq}$	62	60	57	57.55	60	59	55	55.88
$bert_{avg}$	64	64	64	63.87	58	59	58	58.35
$bert_{max}$	64	65	64	64.77	59	59	59	59.09
$\boldsymbol{legalBert_{512}}$	**66**	**66**	**66**	**66.60**	**61**	**61**	**61**	**62.35**
$legalBert_{seq}$	62	62	60	59.85	61	60	58	58.07
$legalBert_{avg}$	64	65	64	64.63	60	61	60	60.74
$legalBert_{max}$	65	65	65	65.19	60	61	60	60.62
Adaboost								
$bert_{512}$	99	99	99	99	77	69	69	73.11
$bert_{seq}$	99	99	99	99	74	70	70	72.18
$bert_{avg}$	99	99	99	99	73	69	69	71.62
$bert_{max}$	99	99	99	99	74	69	69	71.80
$\boldsymbol{legalBert_{512}}$	**99**	**99**	**99**	**99**	**79**	**71**	**72**	**75.04**
$legalBert_{seq}$	99	99	99	99	75	71	71	73.35
$legalBert_{avg}$	99	99	99	99	75	70	71	73.12
$legalBert_{max}$	99	99	99	99	75	71	71	73.20
Gradient Boost								
$bert_{512}$	96	96	96	96.22	71	71	71	71.88
$bert_{seq}$	97	97	97	96.91	71	71	71	71.44
$bert_{avg}$	98	96	96	96.31	70	70	70	70.83
$bert_{max}$	97	97	97	97.25	70	70	70	70.35
$legalBert_{512}$	96	97	96	76.56	73	73	73	73.82
$legalBert_{seq}$	98	98	98	97.88	73	73	73	73.33
$\boldsymbol{legalBert_{avg}}$	**97**	**97**	**97**	**96.89**	**74**	**74**	**74**	**74.22**
$legalBert_{max}$	97	97	97	97.31	72	72	72	72.94

Table 2. Prediction results using classical deep learning models

Feature Representation	**Train set**				**Test set**			
	Precision (%)	**Recall (%)**	**F1 (%)**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 (%)**	**Accuracy (%)**
Multilayer Perceptron (MLP)								
$bert_{512}$	60	64.1	62	62.39	55.4	66.1	60	59.96
$bert_{seq}$	70.7	71.3	71	70.59	69.9	62.2	64	63.99
$bert_{avg}$	62.7	63.3	63	63.42	60.3	62.1	61	61.12
$bert_{max}$	66.4	61.7	64	64.48	60.9	64.3	62	61.74
$legalBert_{512}$	64.6	54	57	57.39	59.9	54.9	57	57.35
$\boldsymbol{legalBert_{seq}}$	**95**	**97**	**96**	**96**	**71.3**	**73.3**	**72**	**71.87**
$legalBert_{avg}$	81.1	80.9	81	80.79	78.4	64.7	70	69.6
$legalBert_{max}$	75.7	72.3	74	74.01	67.6	64.6	66	65.99
BERT and Legal-BERT as classifier								
$bert_{512}$	88.8	91	90	89.9	64.9	59.1	62	61.9
$bert_{seq}$	92.5	94.1	93	93.3	74.4	69.9	72	72.1
$bert_{avg}$	88.9	91.3	90	90.1	63.8	66.4	65	65.1
$bert_{max}$	91.9	90.5	91	91.2	66	66.4	66	66.2
$legalBert_{512}$	92.3	95.9	94	94.1	69.2	65.3	67	67.2
$\boldsymbol{legalBert_{seq}}$	**97.7**	**95.5**	**97**	**96.6**	**76.7**	**81**	**79**	**78.8**
$legalBert_{avg}$	96.5	94.7	96	95.6	72.4	74.8	74	73.6
$legalBert_{max}$	97.5	94.1	96	95.8	73.7	74.7	74	74.2

Table 3. Prediction results using sequential deep learning models

Model	Text Embeddings	Train Accuracy (%)	Train Recall (%)	Train Precision (%)	Train F1 (%)	Test Accuracy (%)	Test Recall (%)	Test Precision (%)	Test F1 (%)
GRU	BERT	74.4	71.5	69.9	70.7	64.9	59.4	59.4	59.4
GRU	Legal-BERT	93.9	93.3	92.6	92.9	72.4	67.9	68.0	68.0
BiGRU	BERT	83.7	80.3	81.5	80.9	66.2	59.0	61.2	60.1
BiGRU	**Legal-BERT**	**95.9**	**92.5**	**98.0**	**95.1**	**72.8**	**62.5**	**71.0**	**71.0**
LSTM	BERT	71.8	55.6	72.5	62.9	65.7	47.9	63.6	54.6
LSTM	Legal-BERT	76.9	63.8	78.7		70.4	53.9	70.5	61.1
BiLSTM	BERT	70.6	58.3	68.8	63.1	66.1	62	60.4	61.3
BiLSTM	Legal-BERT	81.4	80.7	77.2	78.9	69.9	57.6	67.7	62.3

There is uncertainty in using Legal-BERT in Indian context by some of the researchers [24]. We presented a thorough comparison of BERT and Legal-BERT for the prediction of Indian court judgments using various machine learning and deep learning algorithms. In all the scenarios Legal-BERT clearly outperformed the basic BERT model. This establishes the applicability of Legal-BERT in Indian context as well. In future, we plan to pre-train and fine-tune transformer model using Indian legal documents. Other transformer-based architectures and complete document embeddings will be explored as a future research direction.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: A system for large-scale machine learning. In: 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16). pp. 265–283 (2016)
2. Aletras, N., Tsarapatsanis, D., Preoţiuc-Pietro, D., Lampos, V.: Predicting judicial decisions of the european court of human rights: A natural language processing perspective. PeerJ Computer Science **2**, e93 (2016)
3. Beltagy, I., Lo, K., Cohan, A.: Scibert: A pretrained language model for scientific text. arXiv preprint arXiv:1903.10676 (2019)
4. Bhattacharya, P., Hiware, K., Rajgaria, S., Pochhi, N., Ghosh, K., Ghosh, S.: A comparative study of summarization algorithms applied to legal case judgments. In: European Conference on Information Retrieval. pp. 413–428. Springer (2019)
5. Bhilare, P., Parab, N., Soni, N., Thakur, B.: Predicting outcome of judicial cases and analysis using machine learning. International Research Journal of Engineering and Technology **6**(3), 326–330 (2019)
6. Breiman, L.: Arcing the edge. Tech. rep., Technical Report 486, Statistics Department, University of California at ... (1997)
7. Chalkidis, I., Androutsopoulos, I., Aletras, N.: Neural legal judgment prediction in english. arXiv preprint arXiv:1906.02059 (2019)
8. Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., Androutsopoulos, I.: Legal-bert: The muppets straight out of law school. arXiv preprint arXiv:2010.02559 (2020)
9. Chaphalkar, N., Iyer, K., Patil, S.K.: Prediction of outcome of construction dispute claims using multilayer perceptron neural network model. International Journal of Project Management **33**(8), 1827–1835 (2015)
10. Chen, H., Cai, D., Dai, W., Dai, Z., Ding, Y.: Charge-based prison term prediction with deep gating network. arXiv preprint arXiv:1908.11521 (2019)
11. Chollet, F., et al.: Keras (2015), `https://github.com/fchollet/keras`
12. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**(3), 273–297 (1995)
13. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
14. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences **55**(1), 119–139 (1997)
15. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Annals of statistics pp. 1189–1232 (2001)
16. Gulli, A., Pal, S.: Deep learning with Keras. Packt Publishing Ltd (2017)
17. Ho, T.K.: Random decision forests. In: Proceedings of 3rd international conference on document analysis and recognition. vol. 1, pp. 278–282. IEEE (1995)
18. Huang, K., Altosaar, J., Ranganath, R.: Clinicalbert: Modeling clinical notes and predicting hospital readmission. arXiv preprint arXiv:1904.05342 (2019)

19. Lage-Freitas, A., Allende-Cid, H., Santana, O., Oliveira-Lage, L.: Predicting brazilian court decisions. PeerJ Computer Science **8**, e904 (2022)
20. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J.: Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics **36**(4), 1234–1240 (2020)
21. Liu, Y.: Fine-tune bert for extractive summarization. arXiv preprint arXiv:1903.10318 (2019)
22. Liu, Y., Lapata, M.: Text summarization with pretrained encoders. arXiv preprint arXiv:1908.08345 (2019)
23. Luo, B., Feng, Y., Xu, J., Zhang, X., Zhao, D.: Learning to predict charges for criminal cases with legal basis. arXiv preprint arXiv:1707.09168 (2017)
24. Malik, V., Sanjay, R., Nigam, S.K., Ghosh, K., Guha, S.K., Bhattacharya, A., Modi, A.: Ildc for cjpe: Indian legal documents corpus for court judgment prediction and explanation. arXiv preprint arXiv:2105.13562 (2021)
25. Mandal, A., Chaki, R., Saha, S., Ghosh, K., Pal, A., Ghosh, S.: Measuring similarity among legal court case documents. In: Proceedings of the 10th annual ACM India compute conference. pp. 1–9 (2017)
26. Prakash, S.B.N.: E judiciary: A step towards modernization in indian legal system. Journal of Education & Social Policy **1**(1), 111–124 (2014)
27. Shaikh, R.A., Sahu, T.P., Anand, V.: Predicting outcomes of legal cases based on legal factors using classifiers. Procedia Computer Science **167**, 2393–2402 (2020)
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
29. Verma, K.: e-courts project: A giant leap by indian judiciary. Journal of Open Access to Law **6**(1) (2018)
30. Xiao, C., Zhong, H., Guo, Z., Tu, C., Liu, Z., Sun, M., Feng, Y., Han, X., Hu, Z., Wang, H., et al.: Cail2018: A large-scale legal dataset for judgment prediction. arXiv preprint arXiv:1807.02478 (2018)
31. Yang, W., Jia, W., Zhou, X., Luo, Y.: Legal judgment prediction via multi-perspective bi-feedback network. arXiv preprint arXiv:1905.03969 (2019)
32. Zhong, H., Wang, Y., Tu, C., Zhang, T., Liu, Z., Sun, M.: Iteratively questioning and answering for interpretable legal judgment prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 1250–1257 (2020)