

# Legal Text Classification and Summarization using Transformers and Joint Text Features

Shaz Furniturewala, Racchit Jain, Vijay Kumari and Yashvardhan Sharma

*Department of Computer Science and Information Systems, Birla Institute of Technology and Science Pilani, Pilani Campus*

## Abstract

This paper presents the approaches undertaken while performing relevance classification on legal documents and thereby making summaries of them using extractive summarization for task 2 of the track 'Artificial Intelligence for Legal Assistance'[1] proposed by the Forum of Information Retrieval Evaluation in 2021[2]. The approaches for relevance classification include fine tuning BERT for the downstream task of relevance classification and then using joint text features to classify relevance.

## Keywords

relevance classification, joint text features, BERT, extractive summarization, AILA

## 1. Introduction

Legal case documents have an extremely domain specific language structure and therefore, any kind of operation/analysis on these documents requires human legal experts that can perform these tasks with accuracy and speed. One of these tasks is the summarization of legal documents. Legal domain often requires these summaries to provide compressed but accurate information about judgements and decisions related to a particular case, however due to the specificity of the domain this is often done by a legal expert. The amount of time and skill required to make these summaries manually prove them to be very expensive. Therefore there's a need for an automated method of summarization of these legal documents. This paper discusses approaches for extractive summarization of such documents. The 'Artificial Intelligence for Legal Assistance' track proposed by FIRE 2021, comprised of two tasks. This paper will discuss task-2 of this track, 'Summarization of Legal Judgements'. Each team was provided with an annotated dataset of 500 Supreme Court legal documents along with a headnote summary for each of them. Every sentence in the document was given one out of the seven labels: Facts, Ruling by Lower Court, Argument, Statute, Precedent, Ratio of the decision, Ruling by Present Court as well as a binary label that defines if a particular sentence is relevant or not. The presented approach achieved 1st rank in task 2a of the conference with a precision of 0.64, a recall of 0.58 and an F1 score of 0.59.

---

*Forum for Information Retrieval Evaluation 2021, December 13–17, 2021, India*

✉ f20200025@pilani.bits-pilani.ac.in (S. Furniturewala); f20190145@pilani.bits-pilani.ac.in (R. Jain);

p20190065@pilani.bits-pilani.ac.in (V. Kumari); yash@pilani.bits-pilani.ac.in (Y. Sharma)

🌐 <https://www.bits-pilani.ac.in/pilani/yash/profile> (Y. Sharma)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).



CEUR Workshop Proceedings (CEUR-WS.org)

## 2. Related Work

Based on the comparative study of legal text summarization algorithms done by Paheli Bhattacharya et al [3], it was found that state of the art legal text summarization is done using legal domain specific extractive summarization algorithms. Another approach was found by Atefeh Farzinder et al. [4] who chose to deconstruct the thematic structure of the legal text and identify various themes to improve summarization. An innovative technique legal text classification was found by Jiaming Gao et al [5]. They created a joint feature vector of the legal text by concatenating the statistical feature vector (obtained using tfidf) and the semantic feature vector (obtained from BERT source code). This was then classified using different classifiers.

## 3. Dataset

The training dataset provided by AILA 2021[6] contained 500 document-summary pairs. Each document was annotated by a legal expert and marked with one of seven rhetorical labels as well as relevance to the summary. The role labels are as follows:

1. **Facts (FAC)**: sentences that describe the events that led to the filing of the case
2. **Ruling by Lower Court(RLC)**: Indian Supreme Court cases are given a preliminary ruling by one of the lower courts such as the Tribunal or the High Court. This role denotes sentences that are a ruling/decision by these lower courts
3. **Argument(ARG)**: sentences that correspond to the arguments made by each of the opposing parties
4. **Statute(STA)**: relevant statute cited
5. **Precedent(PRE)**: relevant precedent cited
6. **Ratio of the decision (Ratio)**: sentences that denote the rationale/reasoning given by the Supreme Court for the final judgement
7. **Ruling by Present Court(RLC)**: sentences that denote the final decision given by the Supreme Court for that case document

The train data contained 72192 sentences as training samples. The test data was 50 headnotes annotated with 7 rhetorical roles. This contained a total of 5066 samples. Task 2a required us to label relevant sentences and task 2b required us to create summaries.

## 4. Proposed Technique

### 4.1. Task2a

For this task we propose two techniques, The first one is fine tuning Legal-BERT[7] a pretrained language model on legal data for the downstream task of relevance classification and the next one is using a join text feature approach where we concatenate the statistical features that is the TF-IDF vectors of the judgements with deep semantic features generated by the Legal-BERT model.

**Table 1**  
Pretraining corpora

Corpus	No. of Documents	Total Size in GB	Repository
EU Legislation	61,826	1.9 (16.5%)	EURLEX (eur-lex.europa.eu)
UK Legislation	19867	1.4(12.2%)	LEGISLATION.GOV.UK
ECJ cases	19867	0.6 ( 5.2%)	EURLEX
ECHR cases	12554	0.5 ( 4.3%)	HUDOC
US court cases	164141	3.2 (27.8%)	CASE LAW ACCESS PROJECT
US contracts	76366	3.9 (34.0%)	SEC-EDGAR

#### 4.1.1. Legal-BERT

This method utilizes Transformers based models for the task of relevance classification, this method is similar to that discussed in [8]. The proposed model uses a modified pretrained BERT[9] encoder called LEGAL-BERT-BASE. It is part of a family of BERT models designed to assist natural language processing tasks for the legal domain.

#### 4.1.2. Pretraining of Legal-BERT

The model was pretrained on 12 GB of legal data of various formats from various public sources. The pretraining corpus was: The model has the same architecture as BERT-BASE. It has 12 layers, 768 hidden units and 12 attention heads. This makes it a total of 110M parameters. LEGAL-BERT is trained for 1M steps, approximately 40 epochs, over all of the corpora. Batches consisted of 256 samples and each sentence had up to 512 tokens.

#### 4.1.3. Fine Tuning of Legal-BERT

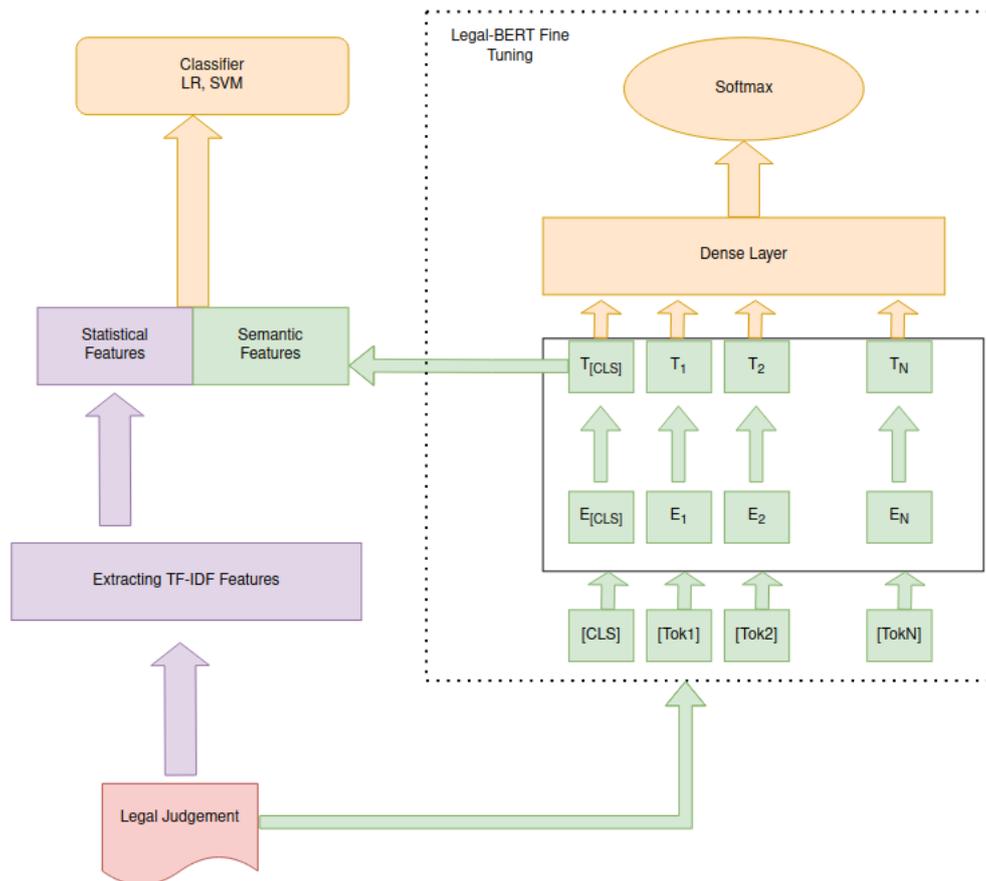
The BERT AutoTokenizer was used to tokenize the inputs. Training data was fed into the model in batches of 32 and trained for 2 epochs. The entire pretrained LEGAL-BERT model was fine tuned for the downstream task of sentence classification into one of two categories (relevant or irrelevant). Adam Optimizer was used while training with a learning rate of  $1e-4$ . The seed value was set to 42 and the model was fine tuned for 2 epochs.

#### 4.1.4. Joint Text Features

Based on the conclusions of Jiaming Gao et al[5]. in their paper on legal text classification we converted the legal text into statistical features and semantic features and combined them for the classification task. The tf-idf vector of the text was used for the statistical features. The vector for train data was acquired through the tfidf vectorizer tool by scikit-learn. The dimensionality of this vector was then reduced to 5000 from 30000 using Latent Semantic Analysis [10] (truncatedSVD) tool of scikit-learn. The semantic feature of the text was acquired from the source code of LEGAL-BERT. From the output results of the last hidden layer the feature vector of the CLS token was extracted. This is a 768-dimensional deep semantic feature of the legal text. The CLS token is also called the Classification token. The reason this token

is used is because it's a fixed embedding that is present at the beginning of every sentence. This indicates that the CLS vector contains BERT's understanding of the sentence because the output of this token is inferred by all the words in the sentence. This means this vector contains all the information which is very useful for a sentence classification task. The final joint text feature was created by concatenating these two feature vectors to form a 5768 dimensional vector. This joint text feature was ultimately classified by a Support Vector Machine and using Logistic Regression.

**Figure 1:** Joint Text Feature model as described in Jiaming Gao et al [5]



## 4.2. Task2b

For the purposes of extractive summarization, we utilized the results of the classification model. Sentences that were labeled relevant by the model were concatenated into one summary. This ensured that the semantic features learnt by the classification model were also used to write an

**Table 2**

Rouge scores on summary by relevant sentences given by fine tuned Legal-BERT model

Rouge Test	Average recall	Average precision	Average F-score
ROUGE-1	0.49168	0.68037	0.53006
ROUGE-2	0.28433	0.39362	0.30703
ROUGE-3	0.19134	0.26491	0.20731
ROUGE-4	0.14920	0.20849	0.16243

extractive summary leading to greater efficiency because a second network did not need to be trained. This approach was chosen based on the results obtained by Paheli Bhattacharya et al.[3] in their paper. They found that legal document specific state of the art extractive summarization produced better results than state of the art classical extractive summarization techniques and neural network based abstractive summarization techniques.

## 5. Results and Evaluation

The submitted model achieved 1st rank based on precision, recall and f-scores for the task of relevance classification. The Legal-BERT approach achieved a precision of 0.64, a recall of 0.58 and an F1-score of 0.59. The joint text feature approach with the deep semantic features from Legal-BERT and statistical features from TF-IDF vectors gave an accuracy of 0.75 with SVM classifier and 0.747 with Logistic Regression classifier. The joint text feature model was trained on 20000 sentences and tested on 5233 sentences. The summaries were evaluated on the basis of their rouge scores. The concatenation of relevant sentences classified by the Legal-BERT model gives the rouge-scores as specified in table 2.

## 6. Conclusion and Future Work

In this paper, we utilized two methods to solve a sentence classification problem. The first method was using LEGAL-BERT, a BERT model that had been pretrained entirely on legal domain data, to classify sentences. This gave us an accuracy of 0.78 (our own evaluation) when trained on 53000 sentences and tested on 18000 sentences. The second method involved creating a joint feature vector of the legal text by combining statistical features, acquired using tf-idf, and semantic text features, extracted from LEGAL-BERT. This joint feature vector was then classified using SVM and Logistic Regression. This gave us an accuracy of 0.75 (our own evaluation) when trained on 20000 sentences and tested on 5000 sentences. Given that this accuracy is comparable to LEGAL-BERT even though it was trained on a much smaller dataset shows that this method has potential to provide much better results. In addition to classification we concatenated the sentences labeled relevant by both models into an extractive summary. These summaries also

gave great results. For future improvements on these models we could increase the training data and connect the BERT CLS vector to a classification network that would allow better learning of semantic features. In addition, we could also take role [11][12] based filtering into account and incorporate that into BERT. For text summarization, the convolutional model implemented by Misha Denil et al. [13] could be utilized after concatenating relevant sentences to improve rouge scores.

## References

- [1] V. Parikh, U. Bhattacharya, P. Mehta, B. Ayan, P. Bhattacharya, K. Ghosh, S. Ghosh, A. Pal, A. Bhattacharya, P. Majumder, Overview of the third shared task on artificial intelligence for legal assistance at fire 2021, in: FIRE (Working Notes), 2021.
- [2] V. Parikh, U. Bhattacharya, P. Mehta, B. Ayan, P. Bhattacharya, K. Ghosh, S. Ghosh, A. Pal, A. Bhattacharya, P. Majumder, Fire 2021 aila track: Artificial intelligence for legal assistance, in: Proceedings of the 13th Forum for Information Retrieval Evaluation, 2021.
- [3] P. Bhattacharya, K. Hiware, S. Rajgaria, N. Pochhi, K. Ghosh, S. Ghosh, A Comparative Study of Summarization Algorithms Applied to Legal Case Judgments, 2019, pp. 413–428. doi:10.1007/978-3-030-15712-8\_27.
- [4] A. Farzindar, G. Lapalme, Letsum, an automatic legal text summarizing system, *Jurix* (2004) 11–18.
- [5] J. Gaoa, H. Ninga, Z. Han, L. Kongb, H. Qib, Legal text classification model based on text statistical features and deep semantic features (2020).
- [6] V. Parikh, V. Mathur, P. Mehta, N. Mittal, P. Majumder, Lawsum: A weakly supervised approach for indian legal document summarization, arXiv preprint arXiv:2110.01188v3 (2021).
- [7] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, I. Androutsopoulos, Legal-bert: The muppets straight out of law school, arXiv preprint arXiv:2010.02559 (2020).
- [8] R. Jain, A. Agarwal, Y. Sharma, Spectre@ aila-fire2020: Supervised rhetorical role labeling for legal judgments using transformers., 2020.
- [9] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: NAACL-HLT, 2019.
- [10] S. T. Dumais, Latent semantic analysis, *Annual review of information science and technology* 38 (2004) 188–230.
- [11] P. Bhattacharya, S. Paul, K. Ghosh, S. Ghosh, A. Wyner, Identification of rhetorical roles of sentences in indian legal judgments, in: Proc. International Conference on Legal Knowledge and Information Systems (JURIX), 2019.
- [12] P. Bhattacharya, P. Mehta, K. Ghosh, S. Ghosh, A. Pal, A. Bhattacharya, P. Majumder, Overview of the FIRE 2020 AILA track: Artificial Intelligence for Legal Assistance, in: Proceedings of FIRE 2020 - Forum for Information Retrieval Evaluation, 2020.
- [13] M. Denil, A. Demiraj, N. Freitas, Extraction of salient sentences from labelled documents (2014).