# DistilRoBERTa Based Sentence Embedding for Rhetorical Role Labelling of Legal Case Documents

Deepthi Sudharsan[1], Asmitha U[1], Premjith B[1] and Soman K.P[1]

[1]*Center for Computational Engineering and Networking (CEN), Amrita School of Engineering, Coimbatore Amrita Vishwa Vidyapeetham, India*

**Abstract**

In a country like India with a very dense and growing population, every year the number of legal judgements filed keep increasing. With increasing number of legal case documents, a systematic and structured organization of the files are essential for the smooth running of the legal system. As a part of AILA 2021, assigning rhetorical roles of legal documents was given as a shared task to automate the process. Deep Learning and Machine Learning models help achieve this task with ease and minimal error. For efficient information retrieval and classification, preprocessing and word embedding techniques such as sentence transformation have been discussed in the paper. Artificial Neural Networks performed the best and consequently, it was used to further evaluate and improve the prediction of the rhetorical roles. In comparison to other Machine Learning and Deep learning models trained for the task, a basic Artificial Neural Network with one hidden layer and $1024 \times 2$ neurons gave the maximum validation accuracy of 85.18% and testing precision of 30.9%.

**Keywords**
Rhetorical Role labelling, distilroberta-base, Artificial Neural Networks

## 1. Introduction

For the efficient working and the smooth administration of the court of law, an organized and efficient structure of storing the legal case documents is obligatory. Manual examination of legal judgments provided by higher courts or legal officials for the acquisition of cardinal information can be a cumbersome and error-prone process. As a result, automatic information retrieval from legal court case transcripts and employing deep learning techniques to classify those judgments would provide several advantages to individuals working in the legal services industry [1]. To ensure the easy readability of the legal judgments and classifying the documents based on their common thematic rhetorical roles such as "Facts of the Case" , "Issues being discussed", "Arguments given by the parties" etc. [2], deep learning networks prove to be efficient. The Artificial Intelligence for Legal Assistance (AILA 2021) [3, 4] came up with a task that aims to classify the rhetorical roles of sentences from legal case documents given the seven predefined roles that it can be classified under.

To come up with an efficient and a less error-prone solution for the task, various machine learning and deep learning models were trained with and without hyperparameter tuning using GridSearchCV, although it later proved to inefficient for this task. Among the machine learning models that were trained using K - Nearest Neighbor (KNN), Decision Tree, Random Forest, Naive Bayes, Multi Layer Perceptron (MLP) and Support Vector Machine (SVM), SVM proved to be more accurate in predicting the roles with an accuracy of 53%. All the deep learning models that were trained (Long Short-Term Memory (LSTM) Networks), Artificial Neural Network (ANN) and Convolutional Neural Network (CNN), performed significantly better than the machine learning models, with ANN performing better than all the models that were trained for the task with a validation accuracy of about 85.18 % on the training dataset. The single layer ANN model was further evaluated for two different runs on the testing dataset, and the performance was analyzed.

The paper is broadly divided into the following sections: Section 2 introduces related research in the field of legal document retrieval; Section 3 provides the dataset information; Section 4 explains the methodology proposed for the task; Section 5 discusses the evaluation outcomes. Finally, Section 6 finishes the work with some suggestions for further improvements for better outcomes.

## 2. Related Works

In paper [5], GloVe, Doc2Vec and Term Frequency-Inverse Document Frequency (TFIDF) based methods were used to perform the labelling of Rhetorical Roles for Legal Judgements given in AILA 2020 [6]. Manual annotation is significantly used in the automatic labelling of the rhetorical role of sentences. Other works deal with the process of annotation – producing a set of rules for annotation, inter - annotator research, and so on – whereas papers that aim to automate the task of semantic labelling also perform an annotation analysis [7]. Classifier models such as fastText have also been proposed as an approach for searching through legal facts from case documents as discussed in [8]. In [9], BM25 ranking algorithm was used for identifying relevant prior cases for a given situation based on best matches. Similar work had been done in [10] as a part of AILA 2019 [11] while additionally using cosine similarity and jaccord similarity. With the rise of Deep learning applications for the purpose of legal information retrieval, a high demand for Neural Network based classification is reflected in many works[12].

## 3. Data Description

For the given task, the provided training data set consists of over 60 case documents and the rhetorical roles for the sentences in each document. The predefined rhetorical roles that were to be predicted are mentioned in Table 1.

**Table 1**
Rhetorical roles and its corresponding description

| S.No | Rhetorical Roles | Description | Class Occurrence |
|---|---|---|---|
| 1 | Facts | Chronology of events that led to filing the case | 2776 |
| 2 | Ruling by Lower Court | Decision by lower courts | 508 |
| 3 | Ruling by Present Court | Final decision given by the Supreme Court | 381 |
| 4 | Statute | Relevant statute cited | 931 |
| 5 | Precedent | Relevant precedent | 1866 |
| 6 | Argument | Arguments of the contending parties | 995 |
| 7 | Ratio of the decision | Rationale by the Supreme Court | 4525 |

# 4. Methodology

A model that can successfully predict the rhetorical roles with minimal errors needed to be designed in legal documents. After successful retrieval of sentences from all the documents and preprocessing, embedding using pre-trained model available in hugging-face transformer[1] was performed on the data set. A variety of machine learning and deep learning models were trained and tested to find the optimal model that could be used to perform the task. The proposed methodology is shown in Figure 1.
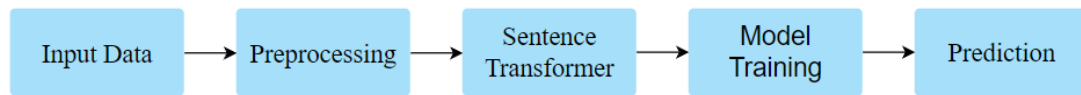


**Figure 1:** Flowchart Depicting Proposed Methodology

## 4.1. Preprocessing

In the data cleaning process, numerical characters, punctuation and extra white spaces were removed using regular expression package. The sentences were modified into lowercase in order to maintain uniformity and using NLTK library[2], stop words were further removed. During exploratory analysis, it was found that 69 sentences were unassigned roles and hence dropped. The non numerical labels were encoded to numerical attributes using label encoder.

## 4.2. Embedding

Sentence embedding was accomplished using pretrained sentence transformer distilroberta - base [13] [14]. Distilroberta - base is a technique from Hugging face library that uses contextual relations between the words to yield contextualized word vector embedding.

---

[1]https://huggingface.co/transformers/
[2]https://www.nltk.org/

### 4.3. Model Training

Initially, Machine learning models like KNN, Decision Tree, Random Forest, Naive Bayes, MLP were trained and SVM classifier was found to have the maximum accuracy of 53 %. To improve the classification accuracy, deep learning models such as LSTM (Long Short Term Memory), [15] ANN (Artificial Neural Networks) and CNN (Convolutional Neural Networks) [16] were trained. Out of the three deep learning models, ANN had comparatively higher accuracy of about 99%. While training, to address the imbalance in the data set 1, class weights [17] were generated and passed as input parameter to the models. For improving the overall performance of the models, GridSearchCV[3] was used on the models to get the best parameters, but there was no improvement in the accuracy when the best parameters yielded by GridSearchCV was used. Hence, ANN model was chosen to perform the classification task. The structure of the selected ANN model is depicted in Figure 2.
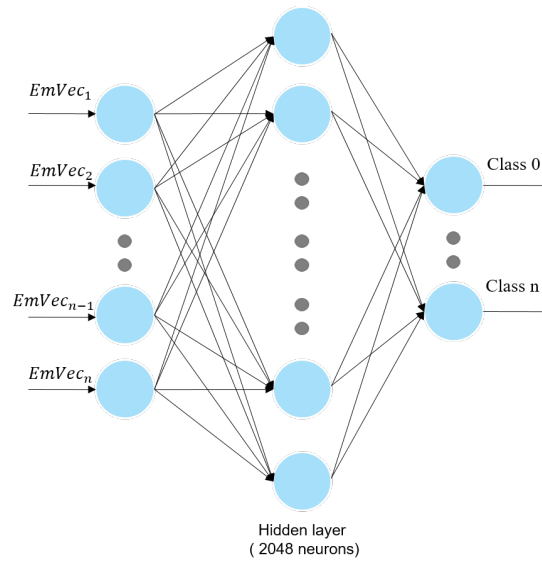


**Figure 2:** Structure of the ANN model

The artificial neural network used has one hidden layer which is connected to the input and output layers. Embedded vectors after passing through the input layer and neurons in the hidden layer is then decoded using inverse transform of the label encoder.

## 5. Results and Discussions

The recorded accuracies for different number of layers and neurons after running for 32 epochs are compared in the Table 2.

---

[3]https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

**Table 2**

Accuracies for different number of layers and neurons

| No. of hidden layers | Epochs | Neurons | Training Accuracy | Validation Accuracy |
|---|---|---|---|---|
| 1 | 32 | 1024 ×2 | 0.9971 | 0.8518 |
| 1 | 32 | 1024 | 0.9965 | 0.9763 |
| 2+1 dropout | 32 | 1024×2 | 0.9927 | 0.8219 |
| 2+1 dropout | 32 | 1024 | 0.9908 | 0.821 |

As observed, for one layer without dropout, the training and validation accuracy seems to be substantially high but it is noticed that for 1024 neurons, the model was over fitting and hence 1024 × 2 neurons was used. The ANN model was run using these optimal parameters for two cases.

In the first run, both the train and test data were embedded together whereas in the second run they were embedded separately.

**Table 3**

Precision, Recall and F1-Score for two runs

| Runs | Precision | Recall | F-Score |
|---|---|---|---|
| Run 1 | 0.179 | 0.194 | 0.179 |
| Run 2 | 0.309 | 0.19 | 0.199 |

From Table 3, it is observed that run two performed better with a precision of 30.9 % than the first run which gave a precision of just 17.9 %. Hence, the training and testing data were embedded separately to achieve better results.

**Table 4**

Categorical comparison of Precision, recall and Fscore for the two runs

| | | Argument | Facts | Precedent | Ratio of the decision | Ruling by Lower Court | Ruling by Present Court | Statute | Overall |
|---|---|---|---|---|---|---|---|---|---|
| Run 1 | Precision | 0.066 | 0.384 | 0.140 | 0.536 | 0.000 | 0.130 | 0.000 | 0.179 |
| | Recall | 0.103 | 0.435 | 0.313 | 0.391 | 0.000 | 0.115 | 0.000 | 0.194 |
| | Fscore | 0.080 | 0.408 | 0.194 | 0.452 | 0.000 | 0.122 | 0.000 | 0.179 |
| Run 2 | Precision | 0.081 | 0.414 | 0.118 | 0.550 | 0.000 | 1.000 | 0.000 | 0.309 |
| | Recall | 0.077 | 0.414 | 0.134 | 0.590 | 0.000 | 0.115 | 0.000 | 0.190 |
| | Fscore | 0.079 | 0.414 | 0.126 | 0.570 | 0.000 | 0.207 | 0.000 | 0.199 |

Table 4 shows the category - wise comparison of the precision, recall and Fscore metrics for both the runs. The single layered ANN architecture that was used predicted the "Ruling by lower court" and "Statute" labels incorrectly for both the runs. Run 2 was able to predict

"Argument", "Facts", "Ratio of the decision" better than Run 1. It is also observed that Run 2 was able to correctly predict "Ruling by Present Court". The Rhetorical role "Precedent" was predicted better by Run 1 in comparison to run 2 unlike the trend shown by the other labels.

## 6. Conclusion

This paper talks about the systematic approach undertaken to successfully predict the rhetorical roles of legal documents using multiple machine learning and deep learning techniques. Basic single layered ANN trained using word embedding from pre - trained sentence transformer, distilroberta - base can help achieve high precision of 30.9 %. The paper can be further expanded by using alternate methods like BM25 ranking algorithm and other methods of embedding like TFIDF or fastText to improve the overall prediction accuracy.

## References

[1] G. Rathnayake, T. Rupasinghe, N. de Silva, M. Warushavithana, V. Gamage, M. Perera, A. Perera, Classifying sentences in court case transcripts using discourse and argumentative properties, International Journal on Advances in ICT for Emerging Regions (ICTer) 12 (2019) 1. doi:10.4038/icter.v12i1.7200.

[2] S. Ghosh, A. Wyner, Identification of rhetorical roles of sentences in indian legal judgments, in: Legal Knowledge and Information Systems: JURIX 2019: The Thirty-second Annual Conference, volume 322, IOS Press, 2019, p. 3.

[3] 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.

[4] 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.

[5] I. Almuslim, D. Inkpen, Document level embeddings for identifying similar legal cases and laws, in: FIRE, 2020.

[6] 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: FIRE (working notes), 2020.

[7] J. Šavelka, K. D. Ashley, Segmenting U.S. court decisions into functional and issue specific parts, in: JURIX, 2018.

[8] I. Nejadgholi, R. Bougueng Tchemeube, S. Witherspoon, A semi-supervised training method for semantic search of legal facts in canadian immigration cases, 2017. doi:10.3233/978-1-61499-838-9-125.

[9] B. Gain, D. Bandyopadhyay, A. De, T. Saikh, A. Ekbal, Iitp at aila 2019: System report for artificial intelligence for legal assistance shared task, 2021.

[10] S. Kayalvizhi, D. Thenmozhi, C. Aravindan, Legal assistance using word embeddings, in: FIRE, 2019.

[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] S. Mandal, S. D. Das, Unsupervised identification of relevant cases & statutes using word embeddings, in: FIRE, 2019.

[13] J. Du, E. Grave, B. Gunel, V. Chaudhary, O. Çelebi, M. Auli, V. Stoyanov, A. Conneau, Self-training improves pre-training for natural language understanding, in: NAACL, 2021.

[14] A. Barua, S. Thara, B. Premjith, K. Soman, Analysis of contextual and non-contextual word embedding models for hindi ner with web application for data collection, in: International Advanced Computing Conference, Springer, 2020, pp. 183–202.

[15] B. Premjith, K. Soman, Deep learning approach for the morphological synthesis in malayalam and tamil at the character level, Transactions on Asian and Low-Resource Language Information Processing 20 (2021) 1–17.

[16] T. T. Sasidhar, B. Premjith, K. Soman, Emotion detection in hinglish (hindi+ english) code-mixed social media text, Procedia Computer Science 171 (2020) 1346–1352.

[17] B. Premjith, K. P. Soman, P. Poornachandran, Amrita_cen@ fact: Factuality identification in spanish text., in: IberLEF@ SEPLN, 2019, pp. 111–118.