Automatic Detection of Rhetorical Role Labels using ERNIE2.0 and RoBERTa

Guneet Singh Kohli[1], PrabSimran Kaur[1] and Jatin Bedi[1]

[1]Department of Computer Science and Engineering, Thapar Institute of Information and Technology, Patiala, Punjab, India - 147001.

Abstract

Automatic detection of the rhetorical roles of sentences in a legal case judgment can help in numerous tasks such as summarizing legal decisions, legal search, etc. Thus, making this problem a field of interest for various researches. Legal case documents, however, are not usually well-structured, which makes this task challenging. In this paper, we propose a multi-class text classification for rhetorical role labeling of legal judgments for task 2 of the track 'Artificial Intelligence for Legal Assistance' presented by the Forum of Information Retrieval Evaluation in 2021. We have implemented the following methodology (i) we used ERNIE 2.0 token embedding, which can better capture the lexical, syntactic, and (ii) semantic aspects of information in the training data. The overall F1 score, Precision, and Recall is 0.505, 0.465, and 0.591 respectively, which is third in all the submitted teams. We make our code publicly available at GitHub [1].

Keywords

Text classification, Rhetorical role labeling, ERNIE 2.0, Transformer, AILA

1. Introduction

In the legal context, the phrase "Rhetorical labeling" of a sentence refers to understanding the semantic function associated with it. Legal case documents have a standard structure, such as facts of the case, ruling by the lower court, issues being discussed, arguments of the parties, the final judgment of the present court, and so on. Distinguishing these rhetorical roles of sentences in a legal case document can help improve the document's readability and support in various downstream tasks like semantic similarity, text summarization, case law analysis, etc. However, lack of structure, high specificity, technical vernacular, and multiple themes make it difficult even for human experts to understand the rhetorical roles .Thus rhetorical labeling is an extremely challenging NLP task. Prior attempts in this work focused on automation using hand-crafted features such as the sequential order of labels or linguistic cues to indicate rhetorical roles. Paheli Bhattacharya proposed deep learning neural network models (a Hierarchical BiLSTM model and a Hierarchical BiLSTM-CRF model) that outperform the hand-crafted features approach by automatically learning the features using pre-trained legal embeddings.

The task 2 of 'Artificial Intelligence for Legal Assistance' track proposed by FIRE 2021 [1], focuses on the motivation to classify these roles. In this work, we attempt to give every

[1] https://github.com/guneetsk99/AILA2021

Forum for Information Retrieval Evaluation, December 13-17, 2021

gkohli_be18@thapar.edu (G. S. Kohli); pkaur_be18@thapar.edu (P. Kaur); jatin.bedi@thapar.edu (J. Bedi)

CEUR Workshop Proceedings (CEUR-WS.org)

sentence in the document one of the seven labels: Facts, Ruling by Lower Court, Argument, Statute, Precedent, Ratio of the decision, Ruling by Present Court.Our team made the following contributions to this problem as part of the shared task effort:

- We use ERNIE 2.0 token embeddings, which can better capture the lexical, syntactic, and semantic aspects of information in the training data.;
- We perform single attention learning to capture long-range relations.;

2. Related Work

There have been various attempts towards the development of automatic identification of rhetorical roles. Initial work focused on understanding the rhetorical roles in case documents to summarize these documents. Later the focus shifted to applying techniques on handcrafted features for segmenting a document into functional and issue-specific parts. For instance, [2] used Conditional Random Fields (CRF) to classify the document into seven rhetorical roles to bring out an effective summary. [3] looked into the segmentation of U.S. court documents into functional and issue-specific parts CRF with handcrafted features.

[4] proposed a skip-gram model for identifying factual and non-factual sentences using a classifier model from fastText library. In another line of work, [5] used the Machine Learning approaches to compare the use of rule-based scripts. Unlike all these works that focused on using handcrafted features to identify rhetorical roles in the legal domain automatically, [6] used a Deep learning approach where no handcrafted features were required. [6] used a hierarchical BiLSTM model, which does not require handcrafted features and performs much better at this task.Later, more deep learning approaches were used for the classification of rhetorical roles. [6] produced a fully annotated dataset 53,210 documents that they collected from a (http://www.westlawindia.com). Later more deep learning approches were applied on the data. For instance, [7] used Roberta embeddings and passed the output through a neural network model for classification. [8] used Bert model for the classification purpose.

3. Dataset

The dataset provided by AILA 2021 contained 60 legal case documents. The training sample included 50 annotated legal text documents, including 9380 training data as shown in Table 1. The test included 10 annotated legal text documents, including 1905 test data as shown in Table 1. Every sentence in the documents was assigned one of the 7 rhetorical roles, as explained below.

1. **Facts (abbreviated as FAC)**: These refer to the sentences that contain information that led to the filing of the case and how it evolved in the legal system. (First Information Report at a police station, filing an appeal).
2. **Ruling by Lower Court (abbreviated as RLC)**: Since the cases mentioned in the dataset were from the Supreme court, preliminary ruling by the lower courts (Tribunal, High Court, etc.). These sentences correspond to the verdicts given by the lower courts.

Table 1
Proportion of each label

Label category	Train	Dataset	Test	Dataset
Ratio of the decision	3624	38.64	587	30.81%
Facts	2219	23.66%	403	21.15%
Precedent	1468	15.65%	319	16.75%
Argument	845	9.0%	256	13.44%
Statute	646	6.89%	167	8.77%
Ruling by lower court	316	3.37%	94	4.93%
Ruling by Present court	262	2.79%	79	4.15%
Total	9380	100%	1905	100%

3. **Argument (abbreviated as ARG)**: These sentences contain the court's discussion over the arguments presented by the opposing parties.
4. **Statute (abbreviated as STA)**: Established law cited from various sources.
5. **Precedent (abbreviated as PRE)**: Relevant precedent cited. These are similar to the statute citations.
6. **Ratio of the decision (abbreviated as Ratio)**: These sentences denote the rationale/reasoning given by the Supreme Court for the application of any legal principle to the legal issue (final judgment).
7. **Ruling by Present Court (abbreviated as RLC)**: These sentences denote the final decision given by the Supreme Court for that case document.

4. Methodology

The Legal case documents are usually lengthy and unstructured, full of legal jargons with missing headings thus making them difficult to read. It becomes a tedious task for a reader to understand where the components are located like facts: that led to filing of cases, arguments: arguments presented by the contenders, statutes: relevant citations to previous statutes and many others similar categories related to legal proceedings. Therefore, the task of semantic/thematic segmentation, also known as rhetorical role labelling of sentences, becomes an important task. It not only enhances the readability of the document but also has applications in several downstream tasks like summarization, case law analysis, semantic search and so on thus increasing the use case of the data along with opening various possibilities.

The methodology used, focused on understanding the semantic relation between the tokens of the sentences that implied towards the legal apprehensions that could be mapped with the provided tokens. The overall task was considered as an extension of a sentence classification problem with seven labels to be predicted. We employed the use of dedicated pre-processing techniques which helped in efficient handling of sentences with token length less than ten by integrating the existing knowledge derived from the data and the observation observed during text EDA. The above pre-processing technique combined with efficient removal of stop words and application of the inflectional stemming which helped in increasing the accuracy of the information retrieval system. The processed text was passed through the complete pipeline of a deep learning-based transformers approach to derive the accurate contextual perceptions of the text and efficiently establish them with the corresponding labels in the data.

4.1. Data Preparation

Properly cleaned data is essential for the correct text analysis and removing unwanted noise before feeding it into the model. Thus, we performed simple preprocessing, which includes tokenization, stopword removal, and lemmatization. Initially, the sentences were split into smaller pieces or ”tokens.” The data was further cleaned by removing common words(stopwords) like ”we” and ”are,” which does not help in text classifications. Finally, the words were lemmatized to obtain the lemma, or base form, of the words.

Additionally, a surprising observation was made regarding the pattern of labels in the case of sentences with word lengths less than 10. Such sentences had the same labels as their previous statement’s label, which helped make the prediction easier for the model in case of lower sentence length, thus making our approach robust enough to handle smaller sentences that lacked high semantic knowledge.

4.2. Modelling

RoBERTa: [9] RoBERTa, retrains BERT with an improved methodology, much more data, larger batch size and longer training times. In RoBERTa the training strategy of BERT is modified by removing the NSP objective. Further, RoBERTa uses byte pair encoding (BPE) as a tokenization algorithm instead of Word Piece tokenization in BERT.

BERT: [10]is a bidirectional language model which aims to learn contextual relations between words using the transformer architecture. We use an official release of the pre-trained models, details about the specific hyperparameters are found in Section V-A. The input to BERT is either a single text (a sentence or document), or a text pair. The first token of each sequence is the special classification token [CLS], followed by WordPiece tokens of the first text A, then a separator token [SEP], and (optionally) after that WordPiece tokens for the second text B. In addition to token embeddings, BERT uses positional embeddings to represent the position of tokens in the sequence. For training, BERT applies Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) objectives. In MLM, BERT randomly masks 15

ERNIE2.0 Transformer Encoder: [11] The model uses a multi-layer Transformer [12] as the basic encoder like other pre-training models such as GPT [13], andBERT [10]. The transformer can capture the contextual information for each token in the sequence via self-attention, and generate a sequence of contextual embeddings. Given a sequence, the special classification embedding [CLS] is added to the first place of the sequence. Furthermore, the symbol of [SEP] is added as the separator in the intervals of the segments for the multiple input segment tasks. Task Embedding The model feeds task embedding to represent the characteristics of different tasks. We represent different tasks with an id ranging from 0 to N. Each task id is assigned to one unique task embedding. The corresponding token, segment, position and task embedding are taken as the input of the model. We can use any task id to initialize our model in the fine-tuning process.

In the present research work, we have implemented the ERNIE 2.0 model to carry out the task

Table 2
Submission RUN1 (ERNIE 2.0)

	ERNIE 2.0		
	Precision	**Recall**	**Fscore**
Argument	0.644	0.744	0.691
Facts	0.622	0.565	0.592
Precedent	0.281	0.582	0.379
Ratio_of_the_decision	0.708	0.545	0.616
Ruling_by_Lower_Court	0.024	0.067	0.036
Ruling_by_Present_Court	0.435	0.769	0.556
Statute	0.542	0.867	0.667
Overall	0.465	0.591	0.505

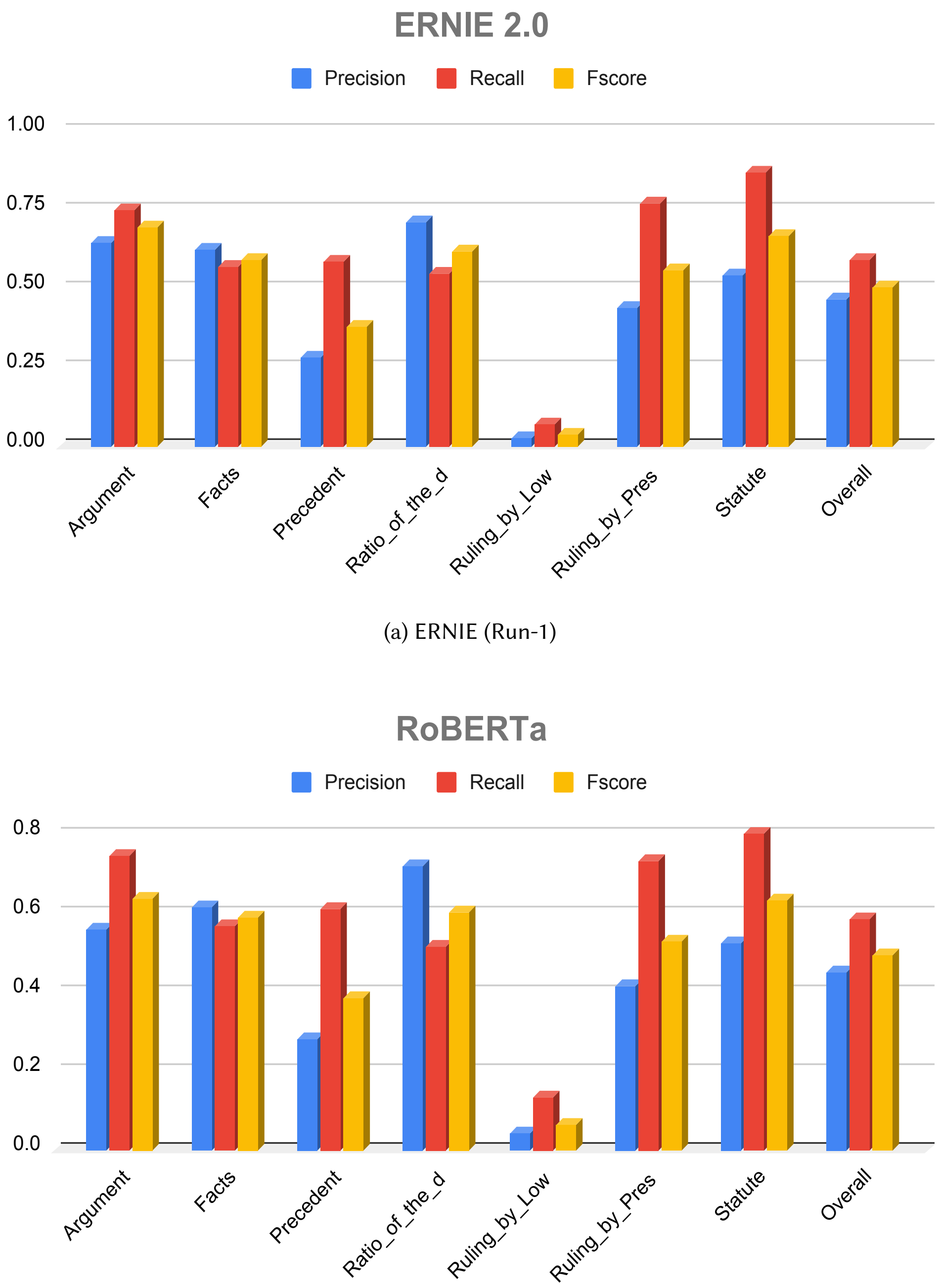
Table 3
Submission RUN2 (RoBERTa)

	RoBERTa		
	Precision	**Recall**	**Fscore**
Argument	0.558	0.744	0.637
Facts	0.616	0.565	0.590
Precedent	0.279	0.612	0.383
Ratio_of_the_decision	0.720	0.517	0.602
Ruling_by_Lower_Court	0.043	0.133	0.065
Ruling_by_Present_Court	0.413	0.731	0.528
Statute	0.522	0.800	0.632
Overall	0.450	0.586	0.491

of Sentence Label Classification which has been established to have outperformed BERT and the recent XLNet in 16 NLP tasks in Chinese and English Language. The base model contains 12 layers, 12 self-attention heads and 768-dimensional of hidden size while the large model contains 24 layers, 16 self-attention heads and 1024-dimensional of hidden size. The model settings of XLNet are the same as BERT. The transformer employed for prediction used ERNIE 2.0 pretrained token embeddings which are known to have better contextual dependencies with each other thus helping us in mitigating the deviation from the text and output label relationship. A single attention mechanism was applied on the token embeddings to derive better understanding of the hidden relationships that could help us in determining the output labels in a more optimized and accurate way. The mentioned pipeline was tested with models like ROBERTA (Base), LawBert and Bert-base-uncased however the performance of the ERNIE 2.0 architecture along with the embeddings generated performed best in our case.

5. Experimentation and Results

The data was tested using the proposed methodology where the two runs submitted are different from each other based on the difference in the base model used which is ERNIE 2.0 in case of Run1 and RoBERTa in case of Run2. After the thorough analysis of the data processed by



(a) ERNIE (Run-1)

(b) RoBERT (Run-2)

Figure 1: Visualization of Comparison Results

our proposed scheme the best run was established to be the use ERNIE 2.0 based sequence classification pipeline built upon the ERNIE 2.0 pre trained token embeddings which are known

Table 4
Hyper parameter tuning of token length

Token Length	Run1 F1 score on validation	Run2 F1 score on validation
200	0.684	0.677
250	0.716	0.704
300	0.653	0.671

Table 5
Validation of proposed methodology

Token Length	Run1 F1 score on validation	Run2 F1 score on validation
Yes	0.512	0.501
No	0.455	0.478

to capture the contextual understanding in the English language better than existing models like BERT, RoBERTa, XLNet. In reference to Table 2 and Table 3, it can be observed that the f1 score for particularly Argument (ARG), Facts (FAC), Ratio of the decision (Ratio) and ruling by present court (RLC) is more than 0.5 which indicates the ability of our methodology to capture the underlying meanings of the said labels. On closely observing the results of submitted runs (shown in Figure 1), we can derive the conclusion of ERNIE 2.0 in establishing itself as a better analyzer of the legal sentences. The use of Roberta gives a better score when compared on the basis of Precedent and Ruling by lower court however in other labels RUN1 establishes itself to be the go-to approach. When looked at each label separately it can be observed that the percentage of improvement in results from Roberta to Ernie or from Run2 to Run1 can be attributed to the fact that Ernie has a more complex architecture with 12 self-attention heads that are more robust to the category of data that the model encountered. The run1 gets the f1 score of 0.505 which is a direct 3% improvement of results. In reference to Table 4, the token length was set at 250 in the case of Run1 and 300 in case of Run2 however the final outcome showed the accuracy of ERNIE with 250 token lengths to be better. Also, Table 5 validates the approach of using the proposed pre-processing as the results were observed to be getting better a value of 0.05 f1 score. The epochs for both the models were set at 15 and the model were trained on Tesla P100-PCIE-16GB. The final result in case of RUN1 was overall precision of 0.465 in comparison to RUN2's precision of 0.450 and the Recall of Run1 was 0.591 in comparison to Run's 2 recall of 0.586.

6. Conclusion

From the overall experiments carried out on the legal corpus it can be concluded that ERNIE 2.0 comes out as the better analyser of Legal Text and has the capability to capture the underlying meaning in the best way. The corpus having 7 labels have contextual overlapping which makes it difficult for various models in giving a higher performance. However the use of better deep learning approaches with advanced embeddings of ERNIE 2.0 makes the above problem easier to solve. For future purposed ensembling of RoBERTa, ERNIE 2.0 and LawBERT can have promising results along with more exploration of preprocessing on the basis of corresponding token lengths as tried in our proposed methodology.

References

[1] 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).

[2] M. Saravanan, B. Ravindran, S. Raman, Automatic identification of rhetorical roles using conditional random fields for legal document summarization, in: Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I, 2008.

[3] J. Savelka, K. D. Ashley, Segmenting us court decisions into functional and issue specific parts., in: JURIX, 2018, pp. 111–120.

[4] I. Nejadgholi, R. Bougueng, S. Witherspoon, A semi-supervised training method for semantic search of legal facts in canadian immigration cases., in: JURIX, 2017, pp. 125–134.

[5] V. R. Walker, K. Pillaipakkamnatt, A. M. Davidson, M. Linares, D. J. Pesce, Automatic classification of rhetorical roles for sentences: Comparing rule-based scripts with machine learning., in: ASAIL@ ICAIL, 2019.

[6] 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.

[7] S. B. Majumder, D. Das, Rhetorical role labelling for legal judgements using roberta., in: FIRE (Working Notes), 2020, pp. 22–25.

[8] Y. Xu, T. Li, Z. Han, The language model for legal retrieval and bert-based model for rhetorical role labeling for legal judgments., in: FIRE (Working Notes), 2020, pp. 71–75.

[9] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).

[10] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[11] Y. Sun, S. Wang, Y. Li, S. Feng, H. Tian, H. Wu, H. Wang, Ernie 2.0: A continual pre-training framework for language understanding, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 8968–8975.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.

[13] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, OpenAI blog 1 (2019) 9.