Rhetorical Labeling for Legal Judgements using fastText

Tebo Leburu-Dingalo, Edwin Thuma, Gontlafetse Mosweunyane and Nkwebi Peace Motlogelwa

Department of Computer Science, University of Botswana

Abstract

This paper describes our participating systems in the FIRE AILA 2021 shared task on predicting rhetorical roles for sentences in a legal judgement document. In particular we propose three multi-class classifiers to predict for each of the sentences a rhetorical role from the following: facts, arguments, ratio of the decision, precedent, statutes, ruling of lower court and ruling of present court. Each of the classifiers uses a supervised fastText model. As input tokens the first classifier uses unigrams, the second one used bigrams and the last one uses trigrams. Our system that uses trigrams attains an F-Score of 0.340 followed closely by the bigram system at 0.338 while the baseline has a score of 0.317.

Keywords

Rhetorical Role, Facts, Arguments, fastText

1. Introduction

Lawyers or law practitioners often have to consult relevant precedent cases and statutes while preparing legal reasoning for a court case. Since court documents are large in number, it will be beneficial to have an automated tool that assists lawyers to retrieve relevant previous cases and statutes [1]. In addition, court documents are generally very long and unstructured, often with no section or paragraph headings . This negatively impacts the readability of the documents as identifying the most important segments such as facts, arguments and precedents tends to be difficult for the user. Hence there is a need to automatically identify and segment the documents into these meaningful parts to ease readability and allow lawyers timely access to the most crucial information when required. The FIRE AILA 2021 Task 1 Rhetorical Role Labelling for Legal Judgements was suggested as a way to mitigate the difficulty of searching in long and unstructured Indian court documents when the user is looking for specific sections in the documents [2, 3]. To accomplish this, the task suggests that identifying semantic function a sentence in the document is associated with has to b, Prasenjit, Prasenjite understood. This is termed rhetorical role labelling [4]. The task considers seven rhetorical labels inherent to legal documents which are facts of the case, ruling by the lower court, argument, statute, precedent,

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

leburut@ub.ac.bw (T. Leburu-Dingalo); thumae@ub.ac.bw (E. Thuma); mosweuny@ub.ac.bw (G. Mosweunyane); motlogel@ub.ac.ub.bw (N. P. Motlogelwa)

https://www.ub.bw/connect/staff/202 (T. Leburu-Dingalo); https://www.ub.bw/connect/staff/1966 (E. Thuma); https://www.ub.bw/connect/staff/1379 (G. Mosweunyane); https://www.ub.bw/connect/staff/830 (N. P. Motlogelwa)

ratio of the decision and ruling by the present court . The task was started as part of FIRE 2020 AILA Track [4]. For the task, a training dataset consisting of 50 documents containing 9, 308 sentences in total with rhetorical labels assigned by law experts was used, while the test dataset had an additional set of 10 case documents. The dataset was provided by [4]. The 21 runs submitted by the 9 teams employed different methods for rhetorical role labelling. The best performing system in terms of F-Score and Recall was by team ju_nlp [5] who experimented with the transformer architecture ROBERTA (state-of-the-art deep learning model) and BiLSTM with different epochs of the model training for the different runs. Scores attained for F-score and Recall were 0.468 and 0.501 respectively. Team heu_gjm [6] deployed TF-IDF features and deep semantic features using BERT, with different classifiers namely Logistic Regression, Linear Kernel SVM and AdaBoost. The BERT model with Logistic Regression gave the best precision for the task at 0.541. Team double_liu [7] used bag-of-words based features with SVM and Adaboost as classifiers. The team also used the BERT model, which outperformed all systems submitted in terms of accuracy at 0.619. Results from the task show that even with the use of complex deep learning methods rhetorical labelling remains a difficult problem to solve as none of the methods proposed achieves optimal performance. In this work we attempt to address the rhetorical role labelling problem through the use of a fastText classifier. FastText is a linear classifier which has been shown to perform on par with deep learning algorithms in text classification while training at faster speeds and utilizing less processing power [8, 9]. In addition our choice of the fastText model is motivated by its capability to support out of dictionary words which can be useful when working with domain specific corpora. Furthermore, the model allows the use of phrases as input tokens to preserve word order, a practice that has proven effective for classification problems [10, 11, 12]. Thus alongside exploring the effectiveness of the fastText classifier in the detection of rhetorical roles we will further investigate the effectiveness of using bigrams and trigrams in improving classification accuracy.

2. Methodology

Rhetorical Labelling (RL) entails segmenting a document into several coherent sentences and assigning rhetorical roles to these sentences. A rhetorical role describes a semantic function that a sentence plays in a document. The task calls for the labelling of sentences into seven roles as follows: Facts referring to the chronology of events that led to filling the case, ruling by lower court, Arguments of contending parties, relevant cited statute, relevant precedent cited, ratio of the decision referring to rationale/reasoning given for the final judgement and ruling by present court referring to the final decision given by the court. For our study, the task will be approached as a classification problem where each role is considered a class, and each instance of a sentence in a document is classified into only one of the classes. In our experiments we deploy a supervised fastText text classifier trained on the provided Task 1 dataset.

Fasttext [1] is an open source toolkit developed for effective learning of text representations and text classification [6]. FastText incorporates the context of words in its embeddings as surrounding words are taken into account when learning a word representation. Furthermore fastText represents each word as a bag of character n-grams in addition to the word itself which

[1] https://fasttext.cc/

is useful for corpora with rare non-dictionary words. Text representations are obtained by averaging word representations. The representations are then fed into a linear classifier and classes determined by deploying a loss function that computes probability distribution over predefined class labels. By default fastText accepts unigrams as input tokens, however this can be varied to for instance bigrams and trigrams. The loss function generally used is the softmax which is can be changed to hierarchical softmax for larger number of classes to speed up training.

3. Experimental Setup

3.1. Dataset

The training dataset consists of 70 documents with variable number of sentences of different lengths. The test data consists of 10 documents also with varying number of sentences. Each of the sentences in the training dataset is annotated with one of the seven classes. It was noted that the data was unbalanced with an unequal distribution among the classes. Measures to be used for evaluation are Precision, Recall and F1 Score.

3.2. Platform

The Python Programming Language and its libraries is used for all experiments. The Fasttext Open Source Library is used for classification.

3.3. Pre-Processing

Training data sentences are converted to lower case, contractions fixed and punctuations removed. The NLTK library is used to remove stop words. The Porter Stemmer is used to stem the words. To conform to fasttext input file requirements, each sentence is rearranged and a prefix " __label__" affixed to the start of each class label. The final format for each sentence is shown in the example below:

__label__Facts none of her children survived her

For training, data is converted into input text files for training and validation using a ratio of 70/30.

4. Runs Description

In our approach we consider the influence of word order in improving performance. We therefore train a classifier with similar parameters while varying the length of word tokens. The submitted runs are for the different models of the classifier obtained for different input tokens. Each test sentence was pre-processed to lower case, fix contractions, remove stop-words and also stem the words. The Porter Stemmer is used to stem the words.

Table 1
UB_BW Results by Role

Run	Measure	Argument	Facts	Precedent	Ratio of the Decision	Ruling by Lower Court	Ruling by Present Court	Statute
UB_BW RUN 1	Precision	0.3878	0.5236	0.191	0.6331	0.0	0.3721	0.0
	Recall	0.4872	0.4644	0.5075	0.4897	0.0	0.6154	0.0
	F-Score	0.4318	0.4922	0.2776	0.5523	0.0	0.4638	0.0
UB_BW RUN 2	Precision	0.4571	0.597	0.1823	0.6212	0.0	0.4857	0.0
	Recall	0.4103	0.5021	0.5224	0.5103	0.0	0.6538	0.0
	F-Score	0.4324	0.5455	0.2703	0.5603	0.0	0.5574	0.0
UB_BW RUN 3	Precision	0.4545	0.585	0.1943	0.6198	0.0	0.500	0.0
	Recall	0.3846	0.4895	0.5075	0.5446	0.0	0.6538	0.0
	F-Score	0.4167	0.533	0.281	0.5798	0.0	0.5667	0.0

4.1. UB_BW RUN 1

For our baseline the fastText classifier model is trained for 25 epochs at a learning rate of 0.5 with WordNgrams set to unigrams.

4.2. UB_BW RUN 2

In an effort to improve performance, in our second run we use bigrams as our input tokens while the model's learning rate and epochs remain at 25 and 0.5 respectively. A slight improvement is noticed over the baseline in terms of both training accuracy and precision.

4.3. UB_BW RUN 3

In our third run the model's parameters are retained as per the two previous runs, however the input tokens are set to trigrams. A negligible improvement is noted in terms of training accuracy and precision over the second run. Training data results based on Precision, and performance accuracy (on the training set) are shown in the Table 2 and Table 1.

5. Results and Analysis

The performance of our runs relative to other teams systems on the test data is shown in Table 2. For our baseline system we used unigrams as input tokens while for the second and third systems bigrams and trigrams were used respectively. It can be observed from the results that the system that the trigrams based system UB_BW RUN 3 performed much better than the baseline that used unigrams UB_BW RUN 1 across all measures. However a negligible difference is noticed between the trigrams and bigrams system UB_BW RUN 2. A category wise analysis of the results extracted from the results as shown in Table 1 shows all systems performed poorly in terms of predicting labels for the classes Ruling by Lower Court and Statute. It can also

Table 2
Results if Task 1: Rhetorical Role Labeling for Legal Judgements

RUN ID	PRECISION	RECALL	F-SCORE
RUSTIC RUN 1	0.548	0.616	0.557
RUSTIC RUN 2	0.528	0.619	0.551
RUSTIC RUN 3	0.511	0.627	0.549
MINITRUE RUN 1	0.485	0.572	0.517
ARGUABLY RUN 1	0.465	0.591	0.505
MINITRUE RUN 3	0.461	0.57	0.503
MINITRUE RUN 2	0.46	0.565	0.501
SSN_NLP RUN 2	0.451	0.571	0.491
ARGUABLY RUN 2	0.45	0.586	0.491
SSN_NLP RUN 3	0.438	0.571	0.475
NITS LEGAL RUN 2	0.453	0.464	0.451
NITS LEGAL RUN 1	0.441	0.434	0.428
SSN_NLP RUN 1	0.411	0.539	0.409
LEGAL AI 2021 RUN 1	0.394	0.361	0.364
UB_BW RUN 3	0.336	0.369	0.340
UB_BW RUN 2	0.335	0.371	0.338
CHANDIGARH CONCORDIA RUN 3	0.317	0.488	0.329
CHANDIGARH CONCORDIA RUN 2	0.317	0.485	0.327
UB_BW RUN 1	0.301	0.366	0.317
CHANDIGARH CONCORDIA RUN 1	0.29	0.476	0.298
LEGAL NLP RUN 3	0.225	0.227	0.22
CEN NLP RUN 2	0.309		0.199
LEGAL NLP RUN 1	0.197	0.217	0.196
LEGAL NLP RUN 2	0.198	0.215	0.192
CEN NLP RUN 1	0.179	0.194	0.179
NIT AGARTALA RUN 1	0.192	0.22	0.179

be noted that for other classes the baseline system and the bigram system outperformed the trigram system. However the trigram system outperforms the other systems in terms of F-score for all classes

6. Discussion and Conclusion

In this paper we explored the effectiveness of using phrases with the fastText classifier to assign rhetorical labels to sentences in a court case document. While our systems did not give good performance overall we believe that with enhancements and more training data the fastText classifier has potential to benefit the rhetorical labelling task. We also observe performance with the introduction of bigrams and trigrams in the model which indicates that phrases can have a positive influence in a classification task. Going forward we aim to further investigate the influence of phrases in improving text classification by performing empirical evaluation with various models.

References

[1] 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: P. Mehta, T. Mandl, P. Majumder, M. Mitra (Eds.), Working Notes of FIRE 2020 - Forum for Information Retrieval Evaluation, Hyderabad, India, December 16-20, 2020, volume 2826 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 1–11.

[2] V. Parikh, U. Bhattacharya, P. Mehta, A. Bandyopadhyay, 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.

[3] V. Parikh, U. Bhattacharya, P. Mehta, A. Bandyopadhyay, 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.

[4] P. Bhattacharya, S. Paul, K. Ghosh, S. Ghosh, A. Wyner, Identification of rhetorical roles of sentences in indian legal judgments, in: M. Araszkiewicz, V. Rodríguez-Doncel (Eds.), Legal Knowledge and Information Systems - JURIX 2019: The Thirty-second Annual Conference, Madrid, Spain, December 11-13, 2019, volume 322 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2019, pp. 3–12.

[5] S. B. Majumder, D. Das, Rhetorical role labelling for legal judgements using ROBERTA, in: P. Mehta, T. Mandl, P. Majumder, M. Mitra (Eds.), Working Notes of FIRE 2020 - Forum for Information Retrieval Evaluation, Hyderabad, India, December 16-20, 2020, volume 2826 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 22–25.

[6] J. Gao, H. Ning, Z. Han, L. Kong, H. Qi, Legal text classification model based on text statistical features and deep semantic features, in: P. Mehta, T. Mandl, P. Majumder, M. Mitra (Eds.), Working Notes of FIRE 2020 - Forum for Information Retrieval Evaluation, Hyderabad, India, December 16-20, 2020, volume 2826 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 35–41.

[7] L. Liu, L. Liu, Z. Han, Query revaluation method for legal information retrieval, in: P. Mehta, T. Mandl, P. Majumder, M. Mitra (Eds.), Working Notes of FIRE 2020 - Forum for Information Retrieval Evaluation, Hyderabad, India, December 16-20, 2020, volume 2826 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 18–21.

[8] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, Bag of tricks for efficient text classification, CoRR abs/1607.01759 (2016). arXiv:1607.01759.

[9] V. Zolotov, D. Kung, Analysis and optimization of fasttext linear text classifier, CoRR abs/1702.05531 (2017). arXiv:1702.05531.

[10] R. Johnson, T. Zhang, Effective use of word order for text categorization with convolutional neural networks, CoRR abs/1412.1058 (2014). arXiv:1412.1058.

[11] C. Chang, M. Masterson, Using word order in political text classification with long short-term memory models, Political Analysis 28 (2020) 395–411.

[12] S. Jameel, W. Lam, L. Bing, Supervised topic models with word order structure for document classification and retrieval learning, Inf. Retr. J. 18 (2015) 283–330.