CEUR-WS.org/Vol-3178/CIRCLE_2022_paper_29.pdf

Exploring SBERT and Mixup Data Augmentation in Rhetorical Role Labeling of Indian Legal Sentences

Alexandre G. de Lima[1,2,3], Mohand Boughanem[3], Eduardo Henrique da S. Aranha[2], Taoufiq Dkaki[3] and Jose G. Moreno[3]

[1]*Federal Institute of Rio Grande do Norte, Natal, Brazil*
[2]*Federal University of Rio Grande do Norte, Natal, Brazil*
[3]*Institut de Recherche en Informatique de Toulouse, UMR 5505 CNRS, F-31000, Toulouse, France*

Abstract

The rise of the Transformer architecture allowed the creation of huge pre-trained language models that led to new state-of-the-art achievements in general-purpose natural language applications. Such models also have the potential to boost domain-specific applications and so this motivates us to evaluate the performance of SBERT, a Transformer architecture-based model, in a case study of rhetorical role labeling of sentences in legal documents. We perform experiments using classification models and compare their performances through lexical features and semantic features generated by SBERT. We also employ the mixup data augmentation method with the semantic features. From the results, we conclude that exploiting the mixup method is beneficial and that the semantic features have a limited enhancing effect on the classification models of our case study.

Keywords

deep learning, natural language processing, sentence classification

1. Introduction

The rhetorical role of a sentence is a kind of label that assigns the semantic function of the sentence, which varies according to the objectives of the application. In the case of legal judgments, the labels identify relevant elements of the lawsuit, such as facts, arguments of the parties, and the court decision. This task is central as it is commonly used to support downstream applications in the legal domain, such as document summarization [1, 2, 3], fact-based case search [4], argument mining [5] and document segmentation [6].

In recent years Deep Learning models have contributed to the achievement of impressive results in Natural Language Processing (NLP) tasks. One of the main reasons is the rise of the Transformer architecture [7], which led to the creation of very effective pre-trained language models. Bidirectional Encoder Representations from Transformers (BERT) [8] is an example of

CIRCLE'22: Joint Conference of the Information Retrieval Communities in Europe, July 04–07, 2022, Samatan, Gers, France

alexandre.lima@ifrn.edu.br (A. G. d. Lima); Mohand.Boughanem@irit.fr (M. Boughanem); eduardoaranha@dimap.ufrn.br (E. H. d. S. Aranha); Taoufiq.Dkaki@irit.fr (T. Dkaki); Jose.Moreno@irit.fr (J. G. Moreno)
0000-0001-7251-0008 (A. G. d. Lima); 0000-0002-8446-638X (E. H. d. S. Aranha); 0000-0002-8852-5797 (J. G. Moreno)

© 2022 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)

a Transformer based model which set new state-of-the-art achievements on some NLP tasks. Such models have the potential to boost general purpose applications, like language translation, as well as domain-specific ones.

In this context, this work presents the utilization of Sentence BERT (SBERT) [9] and machine learning classification models in the task of rhetorical role labeling of sentences from Indian legal judgments. As models' inputs, we experiment with lexical and semantic features, the last being generated by a SBERT model. Regarding the semantic features, we also exploit the mixup data augmentation method [10]. Our objectives are:

- to check if the semantic features, which are generated by a deep learning model, are capable to improve the performance of machine learning classifiers in the task of rhetorical role labeling of sentences from Indian legal judgments;
- to verify the effect of the mixup method on the performance of the classification models based on semantic features.

We exploit nine classification models whose performances vary in function of feature type. On exploiting semantic features, three models perform worse in all metrics, three models perform better in all metrics, and three models improve and degrade in different metrics. The mixup method, which was exploited with two classification models only, improves one model's performance in all metrics and the other model's performance in Recall and F1 score metrics. The best Precision score (0.5597) is achieved by a XGBoost model trained with lexical features, the best Recall score (0.4425) is achieved by a Naïve Bayes model trained with semantic features, and the best F1 score (0.4290) is achieved by a neural network model trained with augmented semantic features. From these results, we conclude that exploiting the mixup method is beneficial and that the semantic features have a limited enhancing effect on the classification models of our case study. Although, we believe it is worth it to perform additional experimentation with features generated by deep learning models.

The remainder of this paper is organized as follows: Section 2 presents the related works; Section 3 presents the experimentation strategy by introducing the mixup method and the forms chosen for sentence representation; Section 4 presents the experimental setup, i.e., the employed dataset, the models and the train and test procedures; Section 5 presents the results and their respective analysis; finally, in Section 6 the work is summarized and future works are discussed.

2. Related Work

Previous works on rhetorical role labeling have employed hand-crafted features altogether machine learning models such as Conditional Random Fields (CRF), Support Vector Machines (SVM) and Naïve Bayes [2, 6, 1].

Several deep learning-based approaches have also been proposed. Yamada et al. [3] compare the performance of CRF models and deep learning models over a dataset comprising 120 Japanese civil judgments, 48,370 sentences, and 7 rhetorical roles. The CRF models adopt hand-crafted features, while the deep learning models adopt word embeddings generated from texts of civil law cases. They compare various models and features and they conclude that deep learning models performed better in most of the considered scenarios.

Tran et al. [11] employ a deep learning model and GloVe word embeddings [12] as features to label sentences from the HOLJ dataset [13]. The model combines a Convolutional Neural Network and a Bidirectional Long-Short Term Memory (BiLSTM) network. Their results are better than the one based on hand-crafted features reported by Hackey and Grover [13] over the same dataset.

Ahmad et al. [14] utilizes a BiLSTM network over the Board Veterans' Claims dataset [5], which consists of 6,135 sentences and six rhetorical roles. For features, they exploit three pre-trained word embeddings, GloVe, FastText [15] and Law2Vec [16], and a word embedding set trained with their dataset. The best performance is achieved by the GloVe based model.

Bhattacharya et al. [17] uses judgments from the Indian Supreme Court to craft a dataset containing 50 documents, with 9,380 sentences labeled by three senior Law students. They compare CRF and BiLSTM models and exploit handcrafted features, randomly initialized word embeddings, and word embeddings trained with Indian court case documents. The deep learning models based on the court case word embeddings perform much better than the other approaches. They also verify that the errors committed by the best model are similar to those committed by human annotators.

In their subsequent work, Bhattacharya et al. [18] extend their experiments by using an additional dataset with judgments from the Supreme Court of the United Kingdom, additional deep learning models, including one based on the Transformer architecture, and additional pre-trained embedding features, including ones from BERT and LegalBERT [19]. Regarding the Indian dataset, the best models are again the deep learning ones based on embeddings trained with Indian court cases. Regarding the United Kingdom dataset, the BERT embeddings provide the best results.

3. Proposed strategy

From an operational perspective, the rhetorical role labeling task boils down to a sentence classification task. Thus, each classification model is trained to produce a unique label (rhetorical role) for an input sentence. The classification models do not work with text inputs, so the sentences have to be converted to a numerical representation. In the following, we describe the sentence representations that we adopt and the mixup data augmentation approach.

3.1. Sentence representation

We exploited two types of features to represent sentences: lexical and semantic.

3.1.1. Lexical features

We chose the TF-IDF scheme [20] to generate lexical features. This scheme works by assigning weights to terms occurring in a document, that is part of a collection of documents. Each document is represented by a vector whose each element is the TF-IDF weight of the respective term. We work in a different setting, adopting sentences instead documents to compute the terms' weights.

3.1.2. Semantic features

BERT is a Transformer architecture-based model which set new state-of-the-art on some NLP tasks, including sentence classification. When fed with a sentence, BERT generates n dense vectors, also known as hidden states, for each token in the sentence. The value of n is the number of BERT's internal layers, i.e., 12 or 24. Some strategies were proposed to derive sentence embeddings from the BERT's hidden states, but according to Reimers and Gurevych [9], these were not fairly evaluated and sometimes they perform worse than averaging static word embeddings. To overcome this, Reimers and Gurevych [9] proposed SBERT as a model capable to generate semantically meaningful sentence embeddings from BERT hidden states in a computationally efficient way. With SBERT, a 768-dimensional dense vector represents a sentence.

3.2. Text Mixup

Mixup is a family of data augmentation method that applies a weighted interpolation of two input vectors to generate a new synthetic one [10]. The interpolation is defined by the following equations:

$$\begin{aligned} \tilde{\mathbf{x}} &= \lambda \mathbf{x}_i + (1-\lambda)\mathbf{x}_j, && \mathbf{x}_i, \mathbf{x}_j \in \mathcal{R}^m \\ \tilde{\mathbf{y}} &= \lambda \mathbf{y}_i + (1-\lambda)\mathbf{y}_j, && \mathbf{y}_i, \mathbf{y}_j \end{aligned}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are raw input vectors, m[1] is the representation size of $\mathbf{x}$, $\mathbf{y}_i$ and $\mathbf{y}_j$ are one-hot label encodings, $(\mathbf{x}_i, \mathbf{y}_i)$ and $(\mathbf{x}_j, \mathbf{y}_j)$ are two examples drawn at random from the training data and λ is a value in the interval $[0, 1]$ drawn at random from a Beta(α, α) distribution, where $\alpha > 0$. The authors of mixup [10] claim that including mixup data in the training of a neural network results in a lesser memorization of corrupt labels and that it works like a regularization strategy. In our strategy, we exploit mixup to augment the semantic feature set only.

4. Experimental setup

4.1. Dataset

The Artificial Intelligence for Legal Assistance (AILA) is a series of computational tasks related to the legal domain [21]. The 2021 edition settled two tasks: legal document summarization; and rhetorical role labeling of judgments from the Indian Supreme Court, whose dataset is the one exploited in this work. The dataset comprises 10,024 sentences distributed among 60 documents, which correspond to the 50 documents from [17] and 10 additional test documents created for AILA 2020. Each sentence receives one of the seven rhetorical roles presented in Table 1.

[1] For lexical features, it could be the vocabulary size, and for semantic features, it is the output size of the used transformer.

Table 1
Rhetorical roles of the AILA dataset.

Role	Number of sentences	Description
Ratio of the decision	3,919	Reasoning given by the Supreme Court for the final judgment.
Facts	2,368	Sentences that denote the chronology of events that led to filing the case.
Precedents	1,523	Citation to relevant prior cases.
Argument	901	The arguments of the contending parties.
Statutes	671	Citation to relevant statutes.
Ruling by Lower Court	341	Preliminary ruling given at the lower courts.
Ruling by Present Court	301	Final decision given by the Supreme Court for the current suit.

4.2. Feature Sets

We exploited six feature sets: a lexical feature set, a semantic feature set, and four augmented semantic feature sets.

The lexical feature set consists of TF-IDF vectors generated from standard n-grams[2] representation. The text prepossessing comprises lower case conversion and removal of symbols and numbers[3]. For implementation, we adopt the `TfidfVectorizer` model from Scikit-learn library[4] [22] with default parameter values, except for the `preprocessor`, `ngram_range` and `min_df` parameters. The implementation is available at the code repository[5]. The TF-IDF model is trained with all sentences in the dataset and this results in a vocabulary of 7,438 terms.

The semantic feature set consists of dense vectors generated by a SBERT model. Since the model is pre-trained, the generation of dense representation vectors consists of just feeding the model with text sentences. For implementation, we adopt the Sentence Transformers library [6,7] and the `sentence-transformers/LaBSE` base model.

Each augmented semantic feature set consists of the union between the semantic feature set and a set of 3,006 synthetic mixup vectors. To generate the mixup vectors, we exploit four α values (1.0, 0.7, 0.3, and 0.1) which result in four synthetic mixup vector sets and, as a consequence, produce four augmented semantic feature sets. During the generation, we randomly select sentences from the dataset, but we take care to not employ two raw vectors from the same class (i.e., $\mathbf{y}_i = \mathbf{y}_j$) when generating a synthetic one.

4.3. Classification Models

We train and evaluate the following classification models: Support Vector Machine (SVM), k-nearest neighbors (KNN), Decision Tree, Random Forest, AdaBoost, Naïve Bayes, XGBoost,

[2] We exploit 1- to 3-grams.
[3] We kept stop words.
[4] 1.0.2 version
[5] https://github.com/alexlimatds/circle-2022
[6] https://www.sbert.net/
[7] 2.2.0 version

Multinomial Logistic Regression (LR), and Multilayer Perceptron (MLP).

For the XGBoost model we adopt its Python implementation[8] [23] configured for a multiclass task. For the `tree_method` parameter we adopt the `hist` value when running the lexical features and the `gpu_hist` value in the case of semantic features. We adopt the default values for the other parameters.

Regarding SVM, KNN, Decision Tree, Random Forest, AdaBoost, and Naïve Bayes models, we adopt the Scikit-learn implementations. When it is the case, we fixed the random generator seed through the `random_state` parameter. In general, we adopt the default values for the models' parameters with the following exceptions: `max_depth=5` and `n_estimators=10` for Random Forest; `max_depth=5` for Decision Tree; `n_neighbors=5` for KNN.

The LR and MLP models are implemented with the PyTorch framework[9] [24]. The MLP model has one hidden layer with 100 units activated by the ReLu function. The weights are initialized by using the Kaiming initialization [25] (hidden units) and the Xavier initialization [26] (output units). As optimization method we adopt the Adam algorithm [27] with the following parameters and values: 10^{-3} for learning rate, $\beta 1 = 0.9$, $\beta 2 = 0.999$, $\epsilon = 10^{-8}$ and 10^{-4} for weight decay. The training follows an early stop approach limited to a maximum of 200 epochs. We adopt a batch size of 64.

For the LR model, the weights are initialized by using the Xavier initialization. As optimization method, we adopt the Stochastic Gradient Descent with momentum algorithm [28] with the following parameters and values: 0.5 for learning rate, 0.9 for momentum, and 10^{-4} for weight decay. The training follows an early stop approach limited to a maximum of 1,000 iterations and we adopt a batch size of 64. An exponential learning rate decay is employed with a decay rate equals to 0.95.

We exploit the lexical and semantic features with all classification models. The semantic augmented features are exploited with the LR and MLP models only. This constraint is due to Scikit-learn and XGBoost libraries, which do not support the target vectors encoded with float point values produced by the mixup method.

The models are trained and evaluated through a 5-fold cross-validation. The data splitting is based on documents instead of sentences. It means that, for each fold, sentences from the same document are used exclusively as train data or test data. Regarding the classification models which exploit the augmented semantic feature sets, each fold exploits all augmented data to train a model but does not apply these data for evaluation. The results are reported through Precision (P), Recall (R), and F1 score metrics.

5. Results and Analysis

Tables 2 and 3 show the scores achieved by the models. The best values for each metric are formatted in bold. The underlined values are the best ones for the respective metric and feature set type. In Table 3, α is the hyperparameter adopted for the Beta distribution.

Regarding the performance of the feature set types, we see there is no prevailing one since each type leads to the best scores in a metric: lexical features for Precision, semantic features

[8] 0.90 version

[9] PyTorch 1.10.0 and CUDA 11.1

Table 2
Performance of the classification models. The values are the average of the test macro averages observed in each fold of the cross-validation procedure.

Model	Lexical features			Semantic features		
	P	R	F1	P	R	F1
SVM	0.4829	0.3902	0.4146	0.4824	0.3906	0.4097
KNN	0.2869	0.2064	0.2115	0.4191	0.3746	0.3804
Decision Tree	0.3630	0.2470	0.2391	0.3409	0.2321	0.2291
Random Forest	0.4334	0.1510	0.0961	0.3584	0.2068	0.1857
AdaBoost	0.4517	0.2968	0.3148	0.3249	0.2673	0.2559
Naïve Bayes	0.2822	0.2917	0.2717	0.3788	**0.4425**	0.3862
XGBoost	**0.5997**	0.3435	0.3823	0.5272	0.3376	0.3640
LR	0.5892	0.3678	0.4048	0.5179	0.3674	0.3934
MLP	0.5392	0.3825	0.4159	0.5000	0.3882	0.4113

Table 3
Performance of the classification models trained with semantic augmented features. The values are the average of the test macro averages observed in each fold of the cross-validation procedure.

Model	α	P	R	F1
Best	Lexical	0.5997	0.3902	0.4159
	Semantic	0.5272	0.4425	0.4113
LR	1.0	0.5047	0.4058	0.4193
	0.7	0.5046	0.3949	0.4103
	0.3	0.5107	0.3965	0.4140
	0.1	0.5088	0.4005	0.4206
MLP	1.0	0.5422	0.3967	0.4189
	0.7	0.5247	0.4045	0.4233
	0.3	0.5431	0.4081	**0.4290**
	0.1	0.5177	0.3989	0.4216

for Recall, and augmented semantic features for F1.

Focusing on the change from lexical features to semantic features, we figure a mixed performance of the models: most of them perform worse in the three metrics (SVM, Decision Tree, XGBoost, LR, and Adaboost); KNN and Naïve Bayes substantially improve in the three metrics; Random Forest performs worse in Precision, but it improves Recall and F1; MLP improves Recall, but it performs worse in Precision and F1.

The exploitation of mixup data is advantageous in general since the performance of the models improves for Recall and F1 when we compare the results between the semantic features and the augmented semantic features. About Precision, there is a varied performance. The LR model performs worse for all α values, while the MLP model performs better. The mixup data also allows the MLP model to achieve the best F1 score.

Table 4 presents the classification scores per label achieved by the classification model with

Table 4
Performance of the MLP model with semantic augmented features ($\alpha = 0.3$) over the individual classes. The values are the test averages observed in each fold of the cross-validation procedure.

Rhetorical role	P	R	F1
Argument	0.3765	0.2010	0.2010
Facts	0.3833	0.4721	0.4721
Precedent	0.3524	0.2131	0.2131
Ratio of the decision	0.3910	0.4820	0.4820
Ruling by Lower Court	0.2286	0.0124	0.0124
Ruling by Present Court	0.6009	0.2769	0.2769
Statute	0.3827	0.3831	0.3831

the best macro F1 score. Regarding the F1 score, the model performs better upon the two most frequent labels (*Ratio of the decision* and *Facts*). Interestingly, the *Statute* label, performs better than *Precedents* and *Argument* labels, despite the fact there are considerably more instances of these two labels than the *Statute* label. The worst performance of the model is related to the *Ruling by Lower Court* label, which is bad when compared to the *Ruling by Present Court* label, even though the former has a slightly higher number of instances. The scores per label achieved by the other models are available in the code repository of this work.

6. Conclusion

This paper tackles the task of rhetorical role label of sentences from suits judged by the Indian Supreme Court. We exploit three feature set types with several machine learning models and each feature set type leads to the best performance on a specific metric. The effect of the semantic features on the models' performance is limited since the semantic feature set is not able to boost a classification model in order to overcome the best F1 score related to the lexical feature set. This just can be achieved when applying the mixup method to the semantic features.

We believe there is room to improve the models' performance. Not just the achieved scores highlight this, but also the results reported by other works. Bhattacharya et al. [17] achieve their best results when they exploit word embeddings trained with judicial texts from the same context of their dataset. Parikh et al. [21] report that their best models are based on LegalBERT. So, for future works, we desire to explore language models based on the legal domain, such as LegalBERT and Law2Vec, as well as fine-tune SBERT. The gains provided by the mixup method also motivate us to better explore data augmentation approaches.

We also intend to improve our experimental framework in order to embody statistical tests and check if the differences among the achieved scores are significant.

Acknowledgments

This work was partially supported by the LawBot project (ANR-20-CE38-0013), granted by ANR the French Agence Nationale de la Recherche.

References

[1] B. Hachey, C. Grover, Extractive summarisation of legal texts, Artif. Intell. Law 14 (2006) 305–345. URL: https://doi.org/10.1007/s10506-007-9039-z. doi:10.1007/s10506-007-9039-z.

[2] M. Saravanan, B. Ravindran, S. Raman, Automatic identification of rhetorical roles using conditional random fields for legal document summarization, in: Third International Joint Conference on Natural Language Processing, IJCNLP 2008, Hyderabad, India, January 7-12, 2008, The Association for Computer Linguistics, 2008, pp. 481–488. URL: https://aclanthology.org/I08-1063/.

[3] H. Yamada, S. Teufel, T. Tokunaga, Neural network based rhetorical status classification for japanese judgment documents, 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. 133–142. URL: https://doi.org/10.3233/FAIA190314. doi:10.3233/FAIA190314.

[4] I. Nejadgholi, R. Bougueng, S. Witherspoon, A semi-supervised training method for semantic search of legal facts in canadian immigration cases, in: A. Z. Wyner, G. Casini (Eds.), Legal Knowledge and Information Systems - JURIX 2017: The Thirtieth Annual Conference, Luxembourg, 13-15 December 2017, volume 302 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2017, pp. 125–134. URL: https://doi.org/10.3233/978-1-61499-838-9-125. doi:10.3233/978-1-61499-838-9-125.

[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: K. D. Ashley, K. Atkinson, L. K. Branting, E. Francesconi, M. Grabmair, B. Waltl, V. R. Walker, A. Z. Wyner (Eds.), Proceedings of the Third Workshop on Automated Semantic Analysis of Information in Legal Texts co-located with the 17th International Conference on Artificial Intelligence and Law (ICAIL 2019), Montreal, QC, Canada, June 21, 2019, volume 2385 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019. URL: http://ceur-ws.org/Vol-2385/paper1.pdf.

[6] J. Savelka, K. D. Ashley, Segmenting U.S. court decisions into functional and issue specific parts, in: M. Palmirani (Ed.), Legal Knowledge and Information Systems - JURIX 2018: The Thirty-first Annual Conference, Groningen, The Netherlands, 12-14 December 2018, volume 313 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2018, pp. 111–120. URL: https://doi.org/10.3233/978-1-61499-935-5-111. doi:10.3233/978-1-61499-935-5-111.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, 2017, pp. 5998–6008. URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[8] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.),

Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 4171–4186. URL: https://doi.org/10.18653/v1/n19-1423. doi:10.18653/v1/n19-1423.

[9] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2019. URL: https://arxiv.org/abs/1908.10084.

[10] H. Zhang, M. Cissé, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, OpenReview.net, 2018. URL: https://openreview.net/forum?id=r1Ddp1-Rb.

[11] V. D. Tran, M. L. Nguyen, K. Shirai, K. Satoh, An approach of rhetorical status recognition for judgments in court documents using deep learning models, in: 11th International Conference on Knowledge and Systems Engineering, KSE 2019, Da Nang, Vietnam, October 24-26, 2019, IEEE, 2019, pp. 1–6. URL: https://doi.org/10.1109/KSE.2019.8919370. doi:10.1109/KSE.2019.8919370.

[12] J. Pennington, R. Socher, C. Manning, GloVe: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1532–1543. URL: https://aclanthology.org/D14-1162. doi:10.3115/v1/D14-1162.

[13] B. Hachey, C. Grover, A rhetorical status classifier for legal text summarisation, in: Text Summarization Branches Out, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 35–42. URL: https://aclanthology.org/W04-1007.

[14] S. R. Ahmad, D. Harris, I. Sahibzada, Understanding legal documents: Classification of rhetorical role of sentences using deep learning and natural language processing, in: IEEE 14th International Conference on Semantic Computing, ICSC 2020, San Diego, CA, USA, February 3-5, 2020, IEEE, 2020, pp. 464–467. URL: https://doi.org/10.1109/ICSC.2020.00089. doi:10.1109/ICSC.2020.00089.

[15] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Trans. Assoc. Comput. Linguistics 5 (2017) 135–146. URL: https://transacl.org/ojs/index.php/tacl/article/view/999.

[16] I. Chalkidis, Law2Vec: Legal word embeddings, 2018. URL: https://archive.org/details/Law2Vec.

[17] 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. URL: https://doi.org/10.3233/FAIA190301. doi:10.3233/FAIA190301.

[18] P. Bhattacharya, S. Paul, K. Ghosh, S. Ghosh, A. Wyner, Deeprhole: deep learning for rhetorical role labeling of sentences in legal case documents, Artificial Intelligence and Law (2021) 1–38.

[19] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, I. Androutsopoulos, LEGAL-

BERT: "preparing the muppets for court'", in: T. Cohn, Y. He, Y. Liu (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020, volume EMNLP 2020 of *Findings of ACL*, Association for Computational Linguistics, 2020, pp. 2898–2904. URL: https://doi.org/10.18653/v1/2020.findings-emnlp.261. doi:10.18653/v1/2020.findings-emnlp.261.

[20] C. D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, Cambridge, UK, 2008.

[21] V. Parikh, U. Bhattacharya, P. Mehta, A. Bandyopadhyay, P. Bhattacharya, K. Ghosh, S. Ghosh, A. Pal, A. Bhattacharya, P. Majumder, AILA 2021: Shared task on artificial intelligence for legal assistance, in: D. Ganguly, S. Gangopadhyay, M. Mitra, P. Majumder (Eds.), FIRE 2021: Forum for Information Retrieval Evaluation, Virtual Event, India, December 13 - 17, 2021, ACM, 2021, pp. 12–15. URL: https://doi.org/10.1145/3503162.3506571. doi:10.1145/3503162.3506571.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[23] T. Chen, C. Guestrin, XGBoost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, ACM, New York, NY, USA, 2016, pp. 785–794. URL: http://doi.acm.org/10.1145/2939672.2939785. doi:10.1145/2939672.2939785.

[24] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[25] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015, IEEE Computer Society, 2015, pp. 1026–1034. URL: https://doi.org/10.1109/ICCV.2015.123. doi:10.1109/ICCV.2015.123.

[26] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Y. W. Teh, M. Titterington (Eds.), Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, volume 9 of *Proceedings of Machine Learning Research*, PMLR, Chia Laguna Resort, Sardinia, Italy, 2010, pp. 249–256. URL: https://proceedings.mlr.press/v9/glorot10a.html.

[27] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015. URL: http://arxiv.org/abs/1412.6980.

[28] I. Sutskever, J. Martens, G. E. Dahl, G. E. Hinton, On the importance of initialization and momentum in deep learning, in: Proceedings of the 30th International Conference

on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013, volume 28 of *JMLR Workshop and Conference Proceedings*, JMLR.org, 2013, pp. 1139–1147. URL: http://proceedings.mlr.press/v28/sutskever13.html.