Predicting Legal Proceedings Status: an Approach Based on Sequential Texts

Felipe Maia Polo [1] **Itamar Ciochetti** [2] **Emerson Bertolo** [2]

1. Objective and practical importance of this work

The objective of this paper is to develop an interpretable model for the classification of sequences of texts and apply it to classify Brazilian legal proceedings in three possible classes of status: (i) archived proceedings, (ii) active proceedings and (iii) suspended proceedings. Each proceeding is made up of a chronological sequence of short texts written by the courts that we will call "motions", which relate to the current state of proceedings, but not necessarily to their status. Although there are 90 different Courts in Brazil (State, Labour, Federal and others) – plus the Supreme Court –, all legal proceedings in Brazil must be included in one of the three presented classes (Archived, Active, Suspended). In spite of the status of a proceeding being an objective information, sometimes it can be hard for public or private organizations with large portfolios to track it because the information: (i) is non-structured and non-standardized, (ii) can be spread in hundreds of separate individual Courts' web pages and (iii) it can be imprecise, incorrect or outdated. Our work may help big public and private organizations to better handle their portfolios since the status is a fundamental information when there is a need to track legal proceedings in large scale.

2. Data and Model Architecture

2.1. Datasets

Our data is composed by two datasets: a dataset of $3 \cdot 10^6$ unlabelled motions (short texts) and a dataset containing 6449 legal proceedings, each with an individual and variable number of motions, but which have been labeled by law experts. Among the labelled data, 47.14% is classified as Archived (class 1), 45.23% is classified as Active (class 2) and 7.63% is classified as suspended (class 3) and we have splitted it in training set (70%), validation set (10%) and test set (20%).

[1]University of São Paulo, São Paulo, Brazil [2]Tikal Tech, São Paulo, Brazil. Correspondence to: Felipe Maia Polo <felipemaiapolo@gmail.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 108, 2020.

2.2. Embedding learning and representation of texts

After pre-processing the texts[1], we tokenize them. To tokenize the texts, we used a method proposed in the literature (Mikolov et al., 2013b) in order to identify which sets of 2 to 4 words generally appear together and should be considered as unique tokens. After that, we used the model CBOW Word2Vec (size=100, window=5) (Mikolov et al., 2013a) to learn the vector representations for each of the tokens in the vocabulary. Then, we normalized the final representations to have a unitary euclidean norm, which is an important step for the interpretability as we will see. Each text is then represented by a matrix of dimensions $R \times D$ where R is the maximum number of tokens allowed per text and D the size of the embeddings. In our case[2] $D = 100$ and $R = 30$.

2.3. Classifier Architecture

Our experience in the Legal field is that the last motion does not contain enough information for our purpose but it is almost guaranteed that the last 5 motions do. Then, we separated the last five (5) motions/texts from each of the legal proceedings and put them in chronological order. To extract features from each motion we used a convolutional layer (Kim, 2014) with K unidimensional filters that run through each text. By cross validation[3], we set K=12. After extracting the features, they pass through a ReLU activation function and then are selected according to the *max-over-time pooling* procedure (Collobert et al., 2011), that is, we kept only one feature per filter - each motion/text will be represented by only K numbers, that feed the Recurrent Neural Network (RNN) with Long Short-Term Memory LSTM units (Hochreiter & Schmidhuber, 1997) with hidden state size $H = 10$, chose by cross validation[4]. We then use a Softmax function to get probabilities at the bottom of the many-to-one RNN. In order to give an interpretable

[1]More details can be found in the full text, available in the Section 7.

[2]We have noticed that over 90% of the motions have a maximum of 30 tokens and that the important information is almost surely not located in the end of the texts. We chose to work with $D = 100$ because it was big enough in our tests.

[3]More details can be found in the full text, available in the Section 7.

[4]The architecture we used is similar to existing approaches in the literature (Lee & Dernoncourt, 2016).

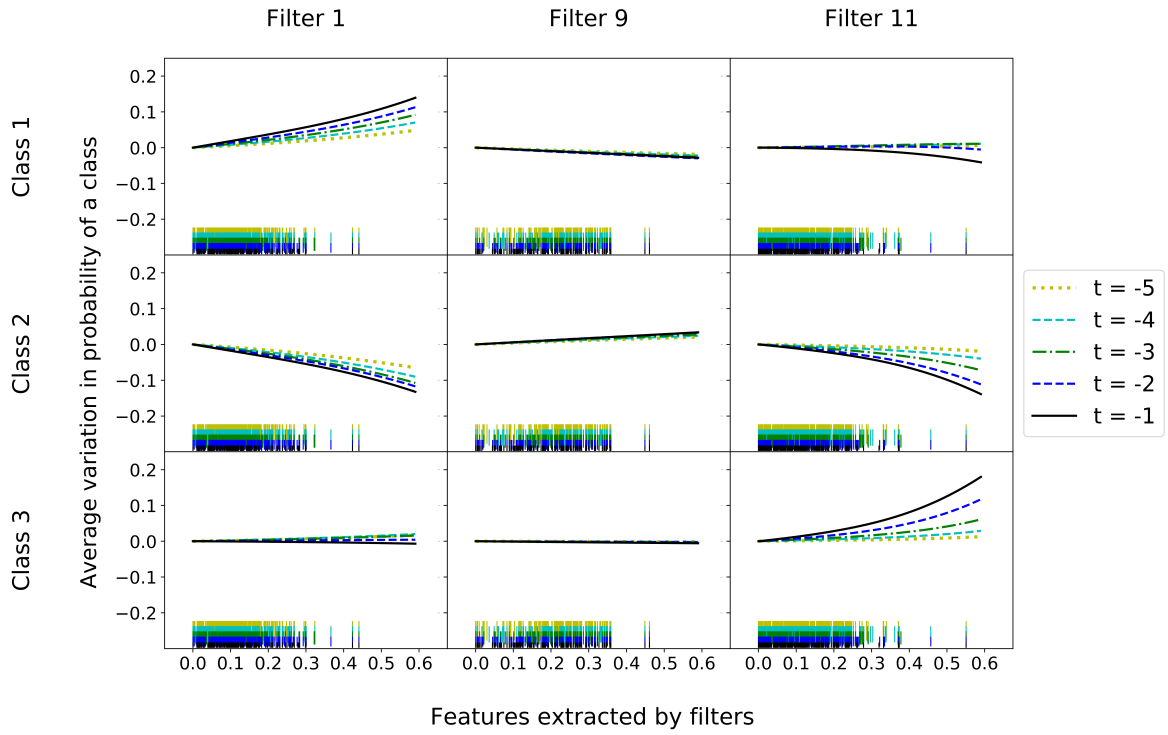

Figure 1. Partial dependence plots

appeal to the solution, as we show ahead, we constrained the euclidean norm of the convolutional filters to be equal one. More details and a scheme are available in the full version of this work (More details in Section 7).

3. Interpretability

Consider $\mathbf{f} \in \mathbb{R}^{100}$ to be a convolutional filter and $\mathbf{x} \in \mathbb{R}^{100}$ a vector representation for a specific tokem. If we constrain the norms of tokens and filters to be unitary, that is $\|\mathbf{f}\| = \|\mathbf{x}\| = 1$, then the feature extracted be filter the $\mathbf{f}$ from the tokem $\mathbf{x}$ is given by $z = \text{ReLU}(\mathbf{f} \cdot \mathbf{x}) = \text{ReLU}[\cos(\theta)]$. In the learning process, the network aligns the filters representations to those representations of the tokens that help the most in the task of classifying legal proceedings. In order to better understand what are the patterns extracted by the convolutional layer of the neural network, let's look at the embeddings representations of tokens in our vocabulary which have the closest representations to the filters according to cosine similarity. As long as we have 12 filters in our model, which is a big quantity, we are going to focus in three specific filters (1, 9 and 11), which bring interesting results. Regarding the *filter 1*, we have[5]: (i) "*final storage of docket*" (0.46), (ii) "*final remittance to origin*" (0.45). Regarding the *filter 9*, we have: (i) "*emitted*" (0.47), (ii) "*certificate*" (0.43). Regarding the *filter 11*, we have: (i) "*temporarily stored docket*" (0.55), (ii) "*docket remain in clerk*" (0.5). It seems filter 1 and 11 are important for us while filter 9 search for patters not directly linked to the classification.

To interpret how each filter relates to the classification task, we will use Partial Dependence Plots (Molnar, 2019). We are interested to see what happens to the predicted probabilities of the three classes when we vary the features extracted by the filters after max pooling, keeping all the other things constant, and considering the possible instants of time - to the most recent to the least recent text. In this paper, we calculated the partial dependence functions according to the test set data and we centered it on zero, so that it is easier to make comparisons between plots. The patterns extracted by filter 1, in Figure 1, explain which legal proceedings are likely to be archived but not suspended or active, which can easily make sense when one sees those expressions linked to filter 1, e.g. 'final storage of docket' and 'final remittance to origin'. Regarding to filter 11, it is possible to notice that the partial dependence functions are decreasing in all plots but the one related to the suspended proceedings. This is understandable because the expressions linked to filter 11 are more common to appear when a proceeding is suspended, e.g. 'temporarily stored docket'. On the other hand, patterns extracted by filter 9, presented in Figure 1, have almost no impact in the decision of the neural network as expected. Also, it seems that more recent information is more important. Overall, the results are very intuitive.

[5]Cosine similarity in parentheses.

Table 1: Aggregate analysis of evaluation metrics

	Macro averaging			Weighted averaging		
Features	F1 Score	Precision	Recall	F1 Score	Precision	Recall
CNN	0.89 ± 0.02	0.92 ± 0.02	0.87 ± 0.03	0.93 ± 0.01	0.93 ± 0.01	0.93 ± 0.01
Doc2Vec	0.81 ± 0.02	0.82 ± 0.03	0.8 ± 0.03	0.84 ± 0.01	0.84 ± 0.02	0.84 ± 0.02
TFIDF	0.88 ± 0.02	0.93 ± 0.02	0.85 ± 0.03	0.92 ± 0.01	0.92 ± 0.01	0.92 ± 0.01
BERT	0.9 ± 0.02	0.92 ± 0.03	0.88 ± 0.03	0.93 ± 0.01	0.94 ± 0.01	0.94 ± 0.01

4. Predictive Performance

In order to present the results and compare them to those obtained by similar alternatives, we will consider three other ways to extract features from the texts (other than convolutional filters), which are applications of the Doc2Vec algorithm(Le & Mikolov, 2014), TFIDF (Salton & McGill, 1986) and BERT-Base (Devlin et al., 2018) (feature-based approach) models[6]. For the Doc2Vec alternative, we kept the specifications for the Word2Vec model that we discussed in Section 2.2. For the TFIDF alternative, we imposed a ceiling of 2000 tokens, keeping the more frequents in the corpus. For both alternatives we applied the processing steps described in Section 2 and trained them using the unlabelled dataset. Regarding the BERT alternative, we fine-tuned a pre-trained portuguese model (Souza et al., 2019) using the Masked Language Model objective on the unlabelled dataset - in this case we applied the same preprocessing steps used by the authors. As one can see in Table[7] 1, we obtained competitive results with our main model. Despite our main proposal achieving similar results to other options, it is in its simplicity[8] and interpretability that this solution stands out.

5. Related work

Despite the efforts made by researchers to create applications in the legal field, we were unable to find in the literature an attempt to solve a problem like ours, perhaps because it is a specific problem in Brazil, which have a less structured legal system. The problems closest to ours we could relate in literature are those related to administrative problems in the legal systems: (i) identifying the parties in legal proceedings (Nguyen et al., 2018), (ii) classification of legal documents according to their administrative labels (Braz et al., 2018; da Silva et al., 2018) or (iii) predicting the area a proceeding belongs to (Sulea et al., 2017). This paper has a different application that can be useful when looking for efficiency in legal systems, especially in developing countries. In addition, we explicitly consider sequences of texts in our model, something that has not yet been observed in the legal literature by us.

Among the interpretable and explainable approaches available, we can obtain (i) those that provide mechanisms for the interpretation/explanation of individual results, referring to a certain data point, (ii) those that allow the interpretation/explanation of the big picture and (iii) those who fulfill both functions. In this sense, our work focus in the big picture interpretation. Some recent works have been developed as applications in the legal area (Chalkidis et al., 2018; Marques et al., 2019; Westermann et al., 2019), but which require a high level of feature engineering, does not provide a big picture interpretation or which are not directly adaptable to sequences of texts. This work contributes to the literature as long as it uses simple tools such as cosine similarity and partial dependence plots for an intuitive interpretation of general results in the classification of text sequences, which can be applied beyond the legal area.

6. Conclusion

We believe that the major contribution of this work is precisely the way we solve an important problem, which is classifying legal proceedings' status, having an interpretable appeal and combining several techniques to analyze sequences of texts in chronological order, which are so common in the legal context.

7. Appendix

You can find the most recent version of the full text ***here*** and the arXiv version ***here***. The code (Jupyter Notebooks) used in this work as well as the datasets can be found in `https://bit.ly/2CnwkKb`. The data can also be found in `https://doi.org/10.6084/m9.figshare.11750061.v1`.

Moreover, we would like to thank *Ana Carolina Domingues Borges*, *Andrews Adriani Angeli* and *Nathália Caroline Juarez Delgado* from Tikal Tech for helping us to obtain the datasets. This work would not be possible without their efforts.

[6] More details can be found in the full text, available in the Section 7.

[7] The 0.95 confidence intervals were calculated using a bootstrap procedure.

[8] Our main model has 2,153 trainable weights while the Doc2Vec benchmark has 15,813, the TFIDF alternative has 243,813 and the BERT one has 163,953. One can see that our main model is much simpler, then less prone to overfitting and easier/faster to train.

References

Braz, F. A., da Silva, N. C., de Campos, T. E., Chaves, F. B. S., Ferreira, M. H., Inazawa, P. H., Coelho, V. H., Sukiennik, B. P., de Almeida, A. P. G. S., Vidal, F. B., et al. Document classification using a bi-lstm to unclog brazil's supreme court. *arXiv preprint arXiv:1811.11569*, 2018.

Chalkidis, I., Androutsopoulos, I., and Michos, A. Obligation and prohibition extraction using hierarchical rnns. *arXiv preprint arXiv:1805.03871*, 2018.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

da Silva, N. C., Braz, F., Gusmão, D., Chaves, F., Mendes, D., Bezerra, D., Ziegler, G., Horinouchi, L., Ferreira, M., Inazawam, P., et al. Document type classification for brazil's supreme court using a convolutional neural network. 2018.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Kim, Y. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

Le, Q. and Mikolov, T. Distributed representations of sentences and documents. In *International conference on machine learning*, pp. 1188–1196, 2014.

Lee, J. Y. and Dernoncourt, F. Sequential short-text classification with recurrent and convolutional neural networks. *arXiv preprint arXiv:1603.03827*, 2016.

Marques, M. R., Bianco, T., Roodnejad, M., Baduel, T., and Berrou, C. Machine learning for explaining and ranking the most influential matters of law. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, pp. 239–243, 2019.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013b.

Molnar, C. *Interpretable Machine Learning*. 2019. https://christophm.github.io/interpretable-ml-book/.

Nguyen, T.-S., Nguyen, L.-M., Tojo, S., Satoh, K., and Shimazu, A. Recurrent neural network-based models for recognizing requisite and effectuation parts in legal texts. *Artificial Intelligence and Law*, 26(2):169–199, 2018.

Salton, G. and McGill, M. J. Introduction to modern information retrieval. 1986.

Souza, F., Nogueira, R., and Lotufo, R. Portuguese named entity recognition using bert-crf. *arXiv preprint arXiv:1909.10649*, 2019. URL http://arxiv.org/abs/1909.10649.

Sulea, O.-M., Zampieri, M., Malmasi, S., Vela, M., Dinu, L. P., and Van Genabith, J. Exploring the use of text classification in the legal domain. *arXiv preprint arXiv:1710.09306*, 2017.

Westermann, H., Walker, V. R., Ashley, K. D., and Benyekhlef, K. Using factors to predict and analyze landlord-tenant decisions to increase access to justice. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, pp. 133–142, 2019.