Predicting Legal Proceedings Status: an Approach Based on Sequential Texts

Felipe Maia Polo[1], Itamar Ciochetti[2] and Emerson Bertolo[2]

[1]University of São Paulo, Brazil

[2]Tikal Tech, Brazil

Objective and practical importance of this work

The objective of this work 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.

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(ii) can be spread in hundreds of separate individual Courts' web pages and (iii) it can be imprecise, incorrect or outdated.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.

Data

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%).

Text representation

After pre-processing the texts, we tokenize them. To tokenize the texts, we used a method proposed in the literature [1] 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) [2] to learn the vector representations for each of the tokens in the vocabulary. Each text is then represented by a matrix of dimensions $\mathrm{R} \times \mathrm{D}$ where R is the maximum number of tokens allowed per text and D the size of the embeddings. In our case $\mathrm{D} = 100$ and $\mathrm{R} = 30$.

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 with K unidimensional filters that run through each text. By cross validation, 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, 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 with hidden state size $\mathrm{H} = 10$, chosen by cross validation. We then use a Softmax function to get probabilities at the bottom of the many-to-one RNN.

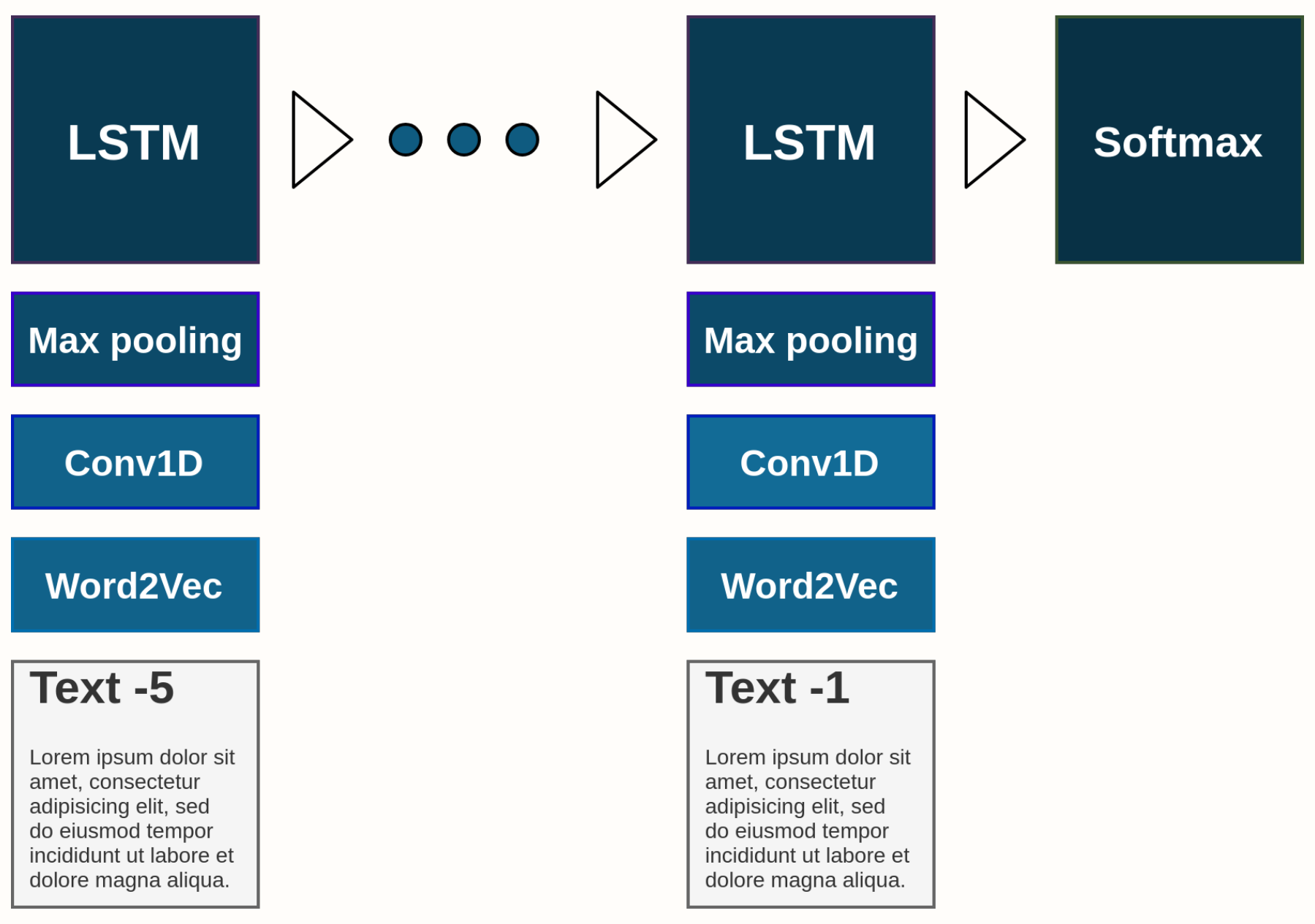
LSTM
LSTM
Softmax
Max pooling
Max pooling
Conv1D
Conv1D
Word2Vec
Word2Vec
Text -5
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Text -1
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Figure 1: Classifier Architecture

Interpretability

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. We have 12 filters in our model, which is a big quantity, then we are going to focus in three specific filters (1, 9 and 11), which bring interesting results. Regarding the *filter 1*, we have[1]: (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.

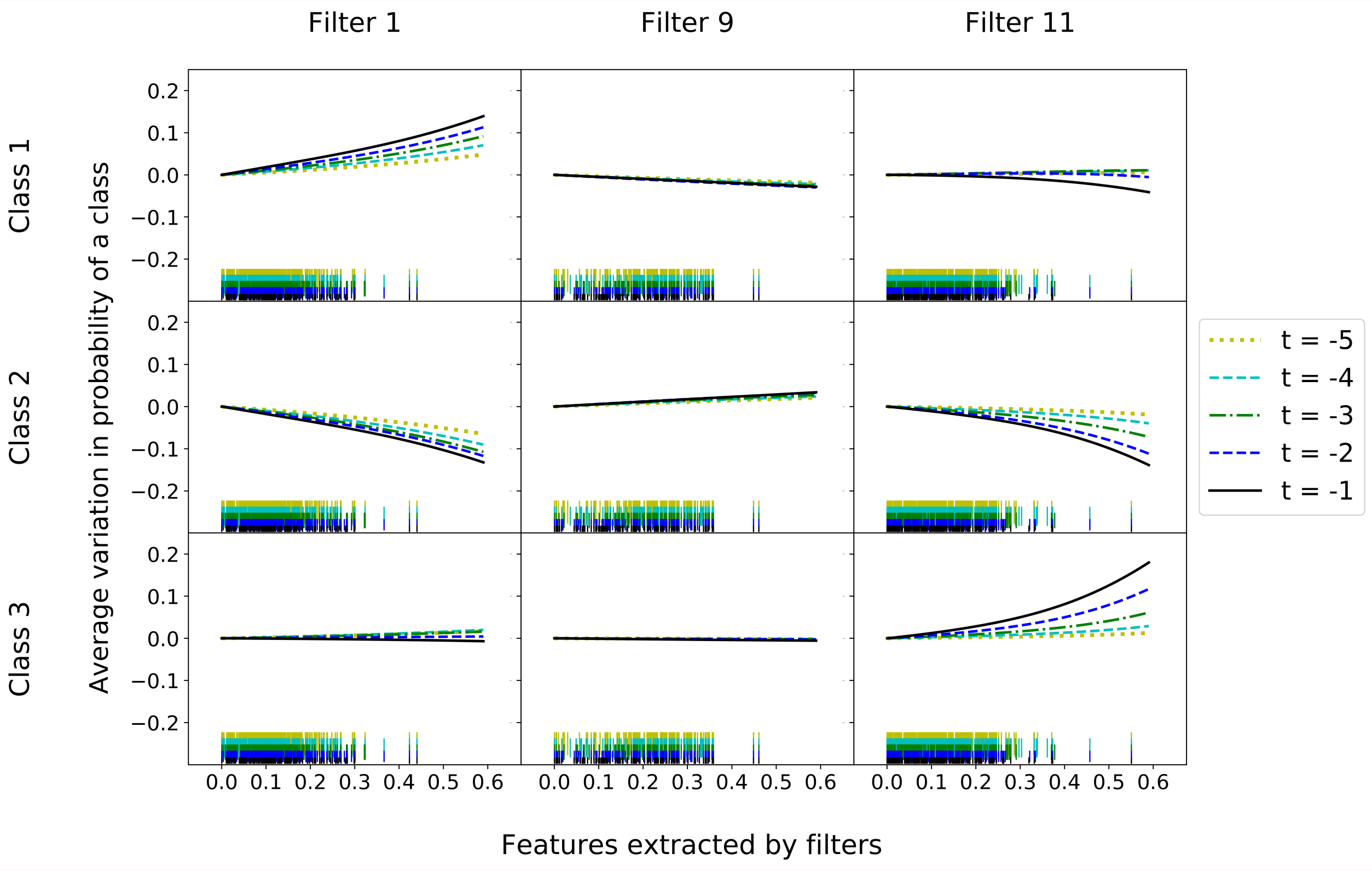

Figure 2: Partial dependence plots

To interpret how each filter relates to the classification task, we will use Partial Dependence Plots. 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. The patterns extracted by filter 1, in Figure 2, 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 2, 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.

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 [3], TFIDF [4] and BERT-Base [5] (feature-based approach) models. For the Doc2Vec alternative, we kept the specifications for the Word2Vec model already we discussed. For the TFIDF alternative, we imposed a ceiling of 2000 tokens, keeping the more frequents in the corpus. For both alternatives we trained them using the unlabelled dataset. Regarding the BERT alternative, we fine-tuned a pre-trained portuguese model [6] using the Masked Language Model objective on the unlabelled dataset. As one can see in Table 1, we obtained competitive results with our main model. Despite our main proposal achieving similar results to other options, it is in its simplicity[2] and interpretability that this solution stands out.

Table 1: Aggregate analysis of evaluation metrics

	Weighted averaging		
Features	F1 Score	Precision	Recall
CNN	0.93 ± 0.01	0.93 ± 0.01	0.93 ± 0.01
Doc2Vec	0.84 ± 0.01	0.84 ± 0.02	0.84 ± 0.02
TFIDF	0.92 ± 0.01	0.92 ± 0.01	0.92 ± 0.01
BERT	0.93 ± 0.01	0.94 ± 0.01	0.94 ± 0.01

References

[1] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean.
Distributed representations of words and phrases and their compositionality.
In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[2] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean.
Efficient estimation of word representations in vector space.
arXiv preprint arXiv:1301.3781, 2013.

[3] Quoc Le and Tomas Mikolov.
Distributed representations of sentences and documents.
In *International conference on machine learning*, pages 1188–1196, 2014.

[4] Gerard Salton and Michael J McGill.
Introduction to modern information retrieval.
1986.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova.
Bert: Pre-training of deep bidirectional transformers for language understanding.
arXiv preprint arXiv:1810.04805, 2018.

[6] Fabio Souza, Rodrigo Nogueira, and Roberto Lotufo.
Portuguese named entity recognition using bert-crf.
arXiv preprint arXiv:1909.10649, 2019.

[1] Cosine similarity in parentheses.

[2] 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.