# An NLP Approach for Detecting a Link Between Text Coherence and the Speed of a Law Project Promulgation

ANDRE MOTTA, Universidade Federal de Minas Gerais, Brasil

One of the best measures against corruption in modern democracies is the usage of transparency tools. The mere fact that the population is able to audit acts done by politicians inhibits and makes it difficult for corruption to occur. The legislative system in Brazil, however, is, albeit transparent in accordance to the law, very keen in obfuscating changes in Law Projects due to Lobbying or attempts to make the project be approved quickly. Through the use of NLP we hope to produce features that can describe the legislative process and it's link to the speed of the Bill's approval.

Key Words and Phrases: natural language processing, brazilian legal system, transparency, word2vec, word mover distance, classification

## 1 INTRODUCTION

Since the promulgation of the Old Republic in Brazil, and even a little earlier, Brazil has had it's legislative process to be a bicameral one. More recently with the promulgation of the 1988 Constitution, right after the end of the 1964's Military Regime, the notion of the workings of a bicameral legislature has been consolidated.

The lawmaking process, is although, by no means a simple one, very well defined. As seen on Figure 7 on Annex A, the legislative process has steps through which any new Bill has to go through until it is Promulgated and Publicated on the Union's Journal. (Being Promulgated is the actual final step that represents the turning of the Bill into a new Law, and the Publication is just an officialization protocol.)

The problem, however, is that some projects may take years or maybe decades until they get through each step on Figure 7. While others may take mere months from the initial idea to the actual publicated Law.

This happens because each house defines a much bigger internal protocol, that will decide on which internal comissions of the house should pre-approve the project or make changes based on the correct form a Law should have. But, sometimes they might make changes based on content as well.

As the project goes through each house, all the changes are publicated on the website of the house (e.g.: Proposal), a privilege we earned in Brazil due to the Law 12.527, that defined criteria of transparency to the whole of the actions taken by politicians.

These changes reflect the tramitation of the project as per the schema in Figure 7 and the internal tramitation of each house. The problem is, unfortunately, the files that are displayed on the website do not reflect the discussions made in order to make small changes, and there are small changes in quantity that alter the meaning of a certain article of the Law Project by a great amount. And substantial changes to projects must be set to a new vote as per the schema.

Given these facts the goal of this particular project is to generate features related to Legislation Projects and it's general coherence in order to try and predict whether the time for the project to be approved will be long or short, given a definition of long and short.

## 2 STRUCTURING THE DATABASE

### 2.1 The Data Scraping

We created a data scraper specific to the websites of the Camara dos Deputados and the Senate. This data scraper collected a database of 3200 Law Projects with their original text and the respective PDF file that was publicated on the Union's Journal.

As our goal was to predict how long it takes for a determined project to be approved, we only collected projects that had been approved and publicated.

### 2.2 The proposed Structuration of Data

Brazilian laws follow the following structure.

1st. Articles

1 - Paragraphs

I. Subsections

a. Items

The problem with this structure when it comes to generating features on about the law, is that the correct way to read each Article is by flattening the tree as defined above and allowing for the combinations. For example:

An article such as:

1st. This law deals with:

1 - The troubles of dealing with excruciatingly:

I. Long contexts

a. in the case of LSTM architecture.
b. in the case of CNN architecture.

II. Long sentences

a. in the case of word2vec architecture.
b. in the case of wmd comparisons.

Author's address: Andre Motta, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil, andre.motta@dcc.ufmg.br.

Has it's context better seen as:

(1) This law deals with: The troubles of dealing with excruciatingly: Long contexts in the case of LSTM architecture.
(2) This law deals with: The troubles of dealing with excruciatingly: Long contexts in the case of CNN architecture.
(3) This law deals with: The troubles of dealing with excruciatingly: Long sentences in the case of word2vec architecture.
(4) This law deals with: The troubles of dealing with excruciatingly: Long sentences in the case of WMD comparisons.

Therefore we developed a parser to flatten the texts into the maximum representation of the law

## 3 METHODOLOGY

### 3.1 Defining Word Embedding

Using the word2vec algorithm, we defined embeddings for every word present in the 3200 files collected.
The word2vec algorithm as seen in [1] comprises on a technique using neural networks to produce embeddings of words that represent the contextual characteristics of a word in the text.
We ran the word2vec algorithm over the flattened text without any uppercase and punctuation in accordance to the proposed structure in order to obtain our own word embedding.

### 3.2 The WMD Algorithm

The WMD (Word Mover's Distance) algorithm as seen in [2] is an algorithm that uses the word embeddings produced by the word2vec in order to numerically compare sentences, or texts.
The algorithm assumes similar words should have similar vectors, so the comparison of words of similar phrases should, as well, render similar values. The algorithm also assumes that words with higher frequency are more important.
So that the algorithm doesn't get mislead, stopwords, punctuation and uppercase are removed from the text. The reasoning of stopwords removal is to remove, articles, prepositions, and other classes of words that have high frequency but usually little semantic meaning on a sentence.

### 3.3 Generation of New Data

*3.3.1 Generation of Features.* In order to generate features we ran the WMD algorithm over each flattened Legislation Project, and compared each sentence to every other sentence.
We saved the value and position of each distance, in order to generate important features. Such as the average distance, the variance and standard deviation, the sum of all distances, and the squared mean distance.
With this we were able to produce relevant internal features to all Legislation Projects.

*3.3.2 The Target.* Our goal was to predict using the features generated by the WMD algorithm, how long would a project take to get to the Promulgation step.
Therefore from the original dataset we extracted the date of the publication of the project from the Original Project File, and the date of the Publication of the Law from the PDF file.

From there, we were able to calculate the time distance between the birth of the project and it's official approval.
To test the algorithm we used three different definitions of what is a long time for approval.

(1) $long\_time > times.mean()$
(2) $long\_time > times.mean() * time.std()$
(3) $long\_time > times.mean() * 2time.std()$

We did not use three standard deviations because there was only a handful of projects in that range.

### 3.4 The Model

We defined then three simple models using the xgboost algorithm as defined by [3]. The models would use the generated features in order to try and predict the defined targets.

## 4 EVALUATING THE MODEL

### 4.1 Metrics

In order to evaluate our models we used the Receiver Operating Characteristic method, that plots the rate of true positives against the rate of false positives. returned the following graphs

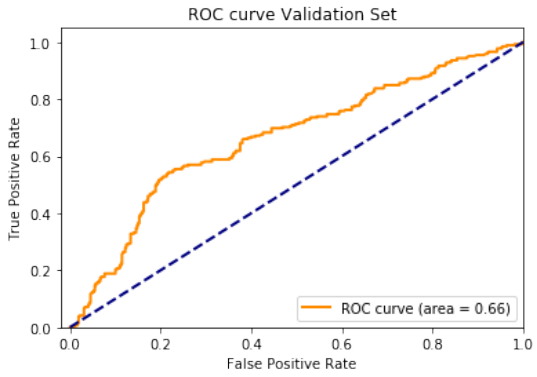


Fig. 1. Long Time 1

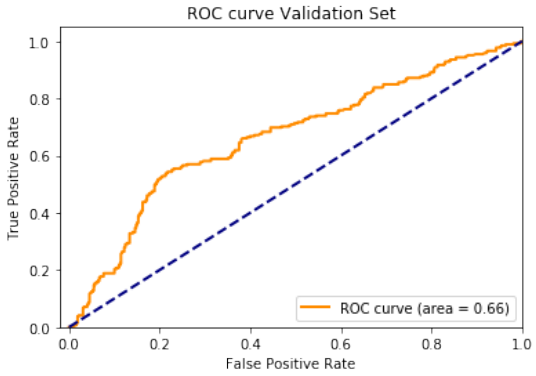


Fig. 2. Long Time 2

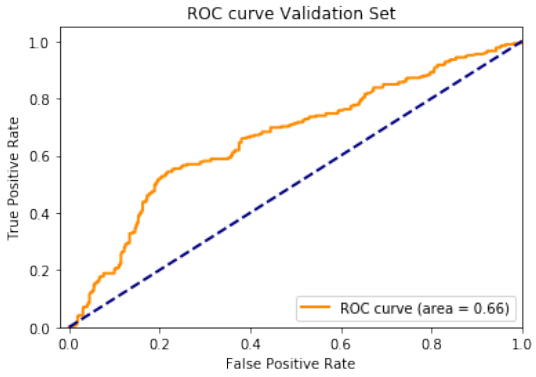


Fig. 3. Long Time 3

## 4.2 Using the SHAP Library

The SHAP package, is a package which is strongly integrated with xgboost and many other models. It's objective is to analyze the effects that one variable has on the outcome of a given predictive model. We used the package in order to try and understand how the features were able to influence the outcome.

As the results of the models were pretty much the same for the three definitions of long time, we will focus on the analysis of the first model, where any project that takes more than the average time of 3 years is considered to take a long time.

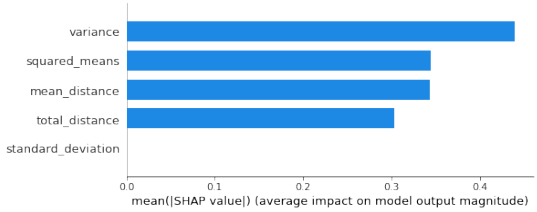


Fig. 4. Magnitude of effect of variables on model decision

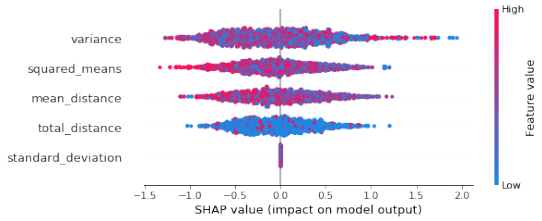


Fig. 5. Effect of variables on model decision

With these two graphics, we are able to see how the variables and its magnitude affect the outcome of the xgboost model in predicting the time it will take for a law project to be approved

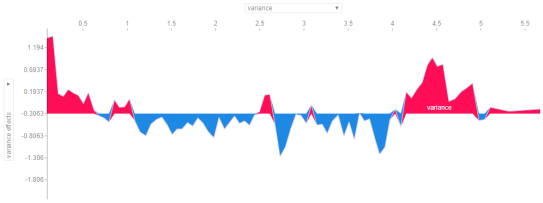

Fig. 6. Variance effect on outcome

Here we can see how the variance affects the outcome of the model. Given that in this context the variance means how coherent the text is. It should be expected that this general coherence should affect the rate in which the project goes through all of the necessary processes to achieve approval or rejection.

## 5 CONCLUSION

The model trained in order to predict how long would it take for a project to be approved was not satisfactory, only being able to indicate the correct class in about 66% of the cases.

These results could indicate that the amount of features produced were not enough on determining how long would it take for any given Law Project to be approved, however, the learning curve and the indications given by using the SHAP package lead us to believe that with a larger dataset and producing more features on the topic of coherence. We might be able to generate better models to this function.

Another thing that hasn't been explored is the possibility that we could have separate embeddings by what sector of society is that project directed for, once there are specifics and the semantics are different depending on a bigger context.

We hope to keep exploring this problem in order to generate better features to solve not only this but many other possible problems in the Brazilian Lawmaking system.

# A FIGURES

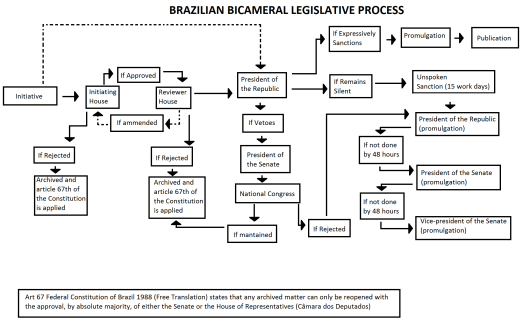


Fig. 7

## ACKNOWLEDGMENTS

The authors would like to thank the entire group from the Agora Digital project that have contributed greatly on the data scraping and on the structuration of data in our weekly discussions.

## REFERENCES

[1] Tomas Mikolov and Kai Chen and Greg Corrado and Jeffrey Dean (2013), "Efficient Estimation of Word Representations in Vector Space," *CoRR* abs/1301.3781, 2013.

[2] Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. (2015), "From word embeddings to document distances," *In Proceedings of the 32nd International Conference on International Conference on Machine Learning* Francis Bach and David Blei (Eds.), Vol. 37. JMLR.org 957-966. 2015.

[3] Tianqi Chen and Carlos Guestrin. (2019), "XGBoost: A Scalable Tree Boosting System," *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* ACM, New York, NY, USA, 785-794. DOI: https://doi.org/10.1145/2939672.2939785