UNIVERSIDADE FEDERAL FLUMINENSE

KID YONATAN VALERIANO VALDEZ

POSTURE: a Framework for Unsupervised Semantic Analysis of Political Speeches

NITERÓI

2019

UNIVERSIDADE FEDERAL FLUMINENSE

KID YONATAN VALERIANO VALDEZ

POSTURE: a Framework for Unsupervised Semantic Analysis of Political Speeches

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Engenharia de Sistemas e Informação

Orientador:
Prof. Dr. Daniel Cardoso Moraes de Oliveira

Co-orientador:
Prof. Dra. Aline Marins Paes Carvalho

NITERÓI

2019

Ficha catalográfica automática - SDC/BEE
Gerada com informações fornecidas pelo autor

V144p Valdez, Kid Yonatan Valeriano
POSTURE: a Framework for Unsupervised Semantic Analysis of Political Speeches / Kid Yonatan Valeriano Valdez ; Daniel Cardoso Moraes de Oliveira, orientador ; Aline Marins Paes Carvalho, coorientadora. Niterói, 2019.
88 f. : il.

Dissertação (mestrado)-Universidade Federal Fluminense, Niterói, 2019.

DOI: http://dx.doi.org/10.22409/PGC.2019.m.06416013797

1. Processamento de linguagem natural. 2. Aprendizado de máquina . 3. Inteligência artificial . 4. Produção intelectual. I. Cardoso Moraes de Oliveira, Daniel, orientador. II. Marins Paes Carvalho, Aline, coorientadora. III. Universidade Federal Fluminense. Instituto de Computação. IV. Título.

CDD -

Bibliotecária responsável: Fabiana Menezes Santos da Silva - CRB7/5274

KID YONATAN VALERIANO VALDEZ

POStURE: a Framework for Unsupervised Semantic Analysis of Political Speeches

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Engenharia de Sistemas e Informação

Aprovada em 28 de Junho de 2019.

BANCA EXAMINADORA

Prof. Dr. Daniel Cardoso Moraes de Oliveira - Orientador, UFF

Prof. Dra. Aline Marins Paes Carvalho - Co-orientadora, UFF

Prof. Dra. Flávia Cristina Bernardini, UFF

Prof. Dra. Jonice de Oliveira Sampaio, UFRJ

Niterói

2019

Acknowledgements

First and foremost, I would like to thank my family, mainly my mother, for all the support she always gives me.

I would also like to thank my two advisors, Prof. Aline Paes, and Prof. Daniel de Oliveira, for teachings, and academic orientation.

I am very grateful with the UFF computer postgraduate program, for the knowledge provided, and CAPES for the award of a research scholarship.

Resumo

Tem sido cada vez mais comum que os candidatos à cargos governamentais apresentem a população suas plataformas de campanha antes do período oficial, usando mecanismos de mídia informais, para estarem mais próximos dos possíveis eleitores. Para decidir o seu voto, entre outros aspectos, os eleitores podem considerar o fluxo dos temas abordados, a coerência e consistência dos discursos dos candidatos, as diferenças entre as posições políticas, e também como eles se comparam em relação à interesses em comum. No entanto, capturar e analisar todas essas questões a partir de discursos é uma tarefa difícil para o eleitor, dado o volume de informações oferecidas por vários meios de comunicação e o viés político de alguns deles. Nesta dissertação, propomos um framework chamado POStURE (POlitical Speech analysis with lingUistic REpresentations) para capturar, analisar e comparar automaticamente os discursos políticos disponibilizados em mídias sociais, apoiando-se em técnicas de Processamento de Linguagem Natural e Aprendizado de Máquina. As métricas propostas para abordar os problemas de descoberta de coerência, consistência e similaridade política, estão centradas principalmente em medidas de similaridade. O POStURE aborda dois tipos de medidas de similaridade, a semelhança geométrica que é baseada na representação do contexto textual com *embeddings* e a semelhança não geométrica, que atua com um algoritmo de alinhamento de seqüências genéticas para encontrar alinhames de tópicos discutidos pelos candidatos. Apresentamos os resultados obtidos com o POSTURE a partir dos discursos dos candidatos durante o processo de eleição presidencial do Brasil em 2018, que permite fazer observações objetivas de como os candidatos se comportam em termos de seus discursos e dos discursos de seus concorrentes.

Palavras-chave: Análise de Discursos, Processamento de Linguagem Natural, Embedding, LDA, Aprendizado de Máquina, Smith-Waterman.

Abstract

It has become increasingly usual that candidates for a government elected position to carry out their campaign in platforms before the official period throughout informal media mechanisms, to be closer to the electors. To decide their vote, among other aspects, the electors may consider the flow of the covered topics, the coherence, and consistency of the candidates' speeches, the differences among the candidates' political positions, and also how they compare to each other regarding common interests. However, capturing and analyzing all of those issues from informal discourses is a difficult task for the elector, given the volume of information offered by various media, and the political bias of some of them. In this dissertation, we propose a framework named POStURE (POlitical Speech analysis with lingUistic REpresentations) to automatically capture, analyze, and compare political speeches supported by Natural Language Processing and Machine Learning techniques. The proposed metrics are centering mainly on the similarity measure. POStURE addresses two types of similarity measures, the geometric similarity that is based in representing textual context with embeddings. The non-geometric similarity represented by Topic Sequence Alignment that is based on a genetic sequences alignment algorithm. We present the results obtained with POStURE from the speeches of the candidates for the presidential election of Brazil in 2018, allow to objective observations of how the candidates behave in terms of their speeches and the speeches of their competitors.

Keywords: Discourse Analysis, Natural Language Processing, Embedding, LDA, Machine Learning, Smith-Waterman.

List of Figures

List of Tables

List of Acronyms and Abbreviations

LDA: Latent Dirichlet Allocation;
NMF: Non-negative matrix factorization;
POStURE: POlitical Speech analysis with lingUistic REpresentations;
GloVe: Global Vectors for Word Representation;
Word2vec: Words Embedding;
Doc2vec: Documents Embedding;
TF-IDF: Term Frequency-Inverse Document Frequency;
BOW: Bag of Words;
PV-DM: Distributed Memory version of Paragraph Vector;
CBOW: Continuous Bag of Words;
PV-DBOW: Distributed Bag of Words version of Paragraph Vector;
NLP: Natural Language Processing;
ML: Machine Learning;
RST: Re-structured text format:
NLTK: Natural Language Toolkit;
PCA: Principal Component Analysis;
TSA: Topic Sequence Alignment;
VSM: Vector Space Model;

Contents

Chapter 1

Introduction

Free and fair elections are crucial to the functioning of democratic institutions since such institutions are made of the people, by the people and for the good of the people. In free and fair elections, candidates for government positions are expected to present their government plans (as campaign promises) for the citizens during an official campaign period. In many countries, the official campaign period is limited to a few months before Election Day. In the Brazilian 2018 general elections that period was of 45 days. As of that date, Campaign materials printed or broadcasted on television or radio are regulated under Republic Act No. 23.551[1].

Traditionally, during the campaign, candidates that are running for positions disseminate campaign promises using television, radio, newspapers, weeklies, magazines, monthlies, banners & graphics, posters and other forms of printed material, *i.e.,* printed media. The contribution of printed and broadcasting media in providing information and transfer of knowledge is remarkable. In fact, one of the most extensive forms of propagating political campaign in Brazil is still the free-of-charge broadcast on open TV and radio stations. However, the increasing importance of social media [22], such as Facebook, YouTube, and Twitter, for propagating information, including political issues [32], is a game changer.

Thus, in the last years, the traditional TV and radio-based campaigns have lost ground to social media, including social networks. As those are all non-official media, candidates start to disseminate their ideas and campaign promises *before* the official period of the electoral campaign. Even during the campaign period using social media is advantageous since candidates have no limit of time for speeches – in the Brazilian elections, the television time depends on the coalitions established between the parties, which means that

[1] `http://www.tse.jus.br/legislacao-tse/res/2017/RES235512017.html`

some candidates have a lot of TV time, while others only get only a few seconds.

Following the same behavior, the press also starts to interview and publish content related to the so-called pre-candidates before the official period[2]. By using the information shared in the social media, the citizen may initiate his voting decision process as soon as possible, taking into account interviews, stories, and videos posted by the candidates and the press.

1.1 Motivation

The availability of the information from campaign activities of candidates may be useful in providing information for the voter or provide more time for the citizen to make a conscious and well-founded decision. But, most voters, for different reasons tend to pay more attention at the end of the campaign, that's why candidates tend to highly increase their efforts and use their creativity to reach the voter and define their vote.

In any of the cases, the volume of information can be so huge that one will have difficulty to collect and understand all data. For instance, if a specific candidate proposes new legislation that will invest more tax dollars in public schools, and later one he mentions that educational fundings will be reduced to use the money to another purpose, the citizen may consider this speech incoherent. However, it may be difficult for the citizen to relate both these promises from a huge number of speeches of all the candidates.

Motivated by that large and difficult-to-analyze material, previous work [27, 11, 13] has made use of computational linguistic language techniques to analyze political content. These include the extraction of themes to discover recurrent topics using non-negative matrix factorization [27, 13] together with a sequencing algorithm in its standard form [11].

1.2 Research Objectives

In this dissertation, we go a step further, by, instead of focusing only on the thematic analysis, as in the related work, we also address aspects at the document level contributing by developing a framework called POSTURE (analysis of political discourse with linguistic representations) to assist the citizen's decision process. This framework should be based on updated concepts in natural language processing so that the analysis can be made as

[2]As this usually happens several months before the election, not all of the pre-candidates turns out to be officially enrolled for the government position.

automatically as possible.

This general objective can be broken down into four more specific purposes that together achieve the overall goal of this dissertation as follows:

- **Investigate NLP techniques and analysis tools to address political discourses** - Although there are models that make possible the analysis of political discourses, it is expected to consider other aspects besides the thematic level.
- **Gathering political discourses from social media** - To analyze the discourses, in this dissertation we have to implement mechanisms for collecting speeches from audio videos that are next converted to text. It is convenient to use these data, since the *written press* uses these interviews as a reference source, to then show fractions or interpretations that may be affected by some political bias.
- **Identify relevant information contained in the speeches** - Extract relevant information from the speeches made by the candidates, for example, the similarity of the discourses, the topics discussed, the difference between candidates, and other information. This information must be calculated automatically.
- **Validate the component models used in the proposed framework** - To ensure that the proposed framework provides the correct information, there must be some way to validate the component models that are used.

1.3 Contributions

POSTURE offers ways of analyzing not only the thematic evolution throughout the campaign but also the coherence of speeches that the candidate is discussing, and the constant speeches. Furthermore, POSTURE provides ways to compare candidates, regarding their political position (left, center-left, center, center-right and right), the similarity between the subjects they talk about, and the thematic comparison of the specific topic.

As expected, the analysis tools that we rely on to build POSTURE are based on the textual content automatically extracted from spoken speeches. Thus, to automatically extract the textual patterns, visualize them, and automatically compare their similarity aspects, we make use of text vectorization techniques such as TF-IDF [43] and Doc2Vec [26]. After embedding the texts into a vector space, to find the topics included within the speeches, we adopt the Latent Dirichlet Allocation (LDA) [5] topic modeling method.

Furthermore, Topic sequence alignment [29] was used as another way to measure the similarity between documents that consider the thematic order of the speeches, which is composed of the topic modeling method (LDA), words vectors representation (Word2Vec) [33], and a sequence alignment algorithm (Smith-Waterman) [44].

In summary, the significant contributions of this work are:

- Mechanisms for analyzing political discourses at the documents level in addition to the thematic level.
- Metrics proposed for the analysis of discourses that allows extracting the necessary information, for example, if a candidate is constant, the coherence of candidates, the topics discussed, etc.
- The data collected from the Brazilian presidential elections of 2018 was made available.
- An algorithm was proposed for the segmentation of thematic sequences for the application of the topic-sequences alignment in political speeches in plain text.
- Proposal of the POSTURE framework for the analysis of political discourses.

1.4 Dissertation Outline

Besides this introduction, the dissertation is organized as follows. Chapter 2 introduces concepts that are used by POSTURE framework and related works. Chapter 3 the POSTURE architecture and formulation have been described. Chapter 4 experimental configuration is explained. Chapter 5 presents results of POSTURE functionalities. Chapter 6 concludes the work and present future work.

Chapter 2

Background

In this Chapter the main concepts used throughout this dissertation are addressed. Section 2.1 present two types of measures to calculate the similarity between documents. Section 2.2 presents the ways of representing words and documents, representing lexical, semantic, and probabilistic content. Section 2.3 presents the sequence alignment algorithm. Finally Section 2.5, outlines some of the major research work in analysis of political speeches.

2.1 Similarity Measures

The increase in the amount of text data from various sources, such as social networks, news, magazines, among others, has been growing year after year, motivating researchers to analyze the content and measure the similarities between the documents. Currently, similarities are used to make queries to find which documents are similar to others, mainly to group documents according to their content. A short time ago, the similarity between documents was more focused on the lexical similarity. Nowadays, we also have tools to estimate the semantic similarity. In this dissertation, we explore two types of similarity measures of documents: based on vector representation and not-based in vector representation. Based on vector representation, the most popular method to measure the similarity between documents, which is based in represents documents as vectors in space using different criteria. The non-vector representation, makes use of methodologies or heuristics to calculate the similarity between documents. The results of some studies [42, 36] show the measure based on vector representations are more robust compared with the other group.

2.1.1 Similarity Based on Vector Representation

This type of measure is based on the Vector Space Model (VSM), which is an algebraic model widely used in data mining and information retrieval. The model uses NLP techniques to represent each document/text as a set of scalar multiplication and sum vectors introduced in [42]. Each vector describes an object, in this case documents and the corpus of texts using vectors of n-dimensions, usually, each one representing the frequency of a certain term in a document. The two most commonly used measures are Euclidean distance that calculates the difference between two points in n-dimensional space based on their coordinate, and the cosine similarity is recommended by the state-of-the-art literature to compare two vector of documents [45, 15]. To compute it, we calculate the angle between the vectors, *i.e.,* the orientation and not the magnitude. If the angle between the vectors is 0^0 the similarity will be 1, and if they form an angle of 90^0, the similarity is 0. The cosine similarity is used particularly in the positive space where the results will be in the range of [0,1]. Equation 2.1 shows how to compute the cosine similarity between two elements A and B.

$$Similarity(A, B) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{2.1}$$

The similarity matrix is a square symmetric matrix, which is equal to its transposition ($A = A^T$ and $A_{i,j} = A_{j,i}$). This matrix is calculated using the cosine similarity function, and is used to speed up the response of the queries regardless of the similarity measure used between documents. It can be used both for based on vector representation, as for non-based on vector representation. When high-dimensional vectors represent the documents, the response time is longer. In the same way, non-based on vector representation usually have more processing time in the calculation of similarity.

2.1.2 Similarity not-based in Vector Representation

There is a variety of algorithms that measure the similarity between a pair of documents using heuristic-based methods, and some significant models will be briefly described. The Pearson correlation coefficient [23] is a statistical model to measure the strength of a linear relationship between the desired data, in data mining it is used to calculate the similarity between two variables (documents or keywords) bounded to -1 and +1. Jaccard Coefficient [20] divides the intersection of objects by their unions. If two documents are

equal, the coefficient is 1 which means that they share exactly the same keywords with the same frequency of each one, and if it is 0 there is no similarity between them. As a recent example of a semantic similarity measure, we have the Word Mover Distance(MVD) [24] that measures the minimum amount of distance that the embedded words (Word2Vec) need to travel to reach the embedded words of the other document. This measure is based on the Distance of Earth Movement (transportation problem).

2.2 Text Representation

Natural Language Processing (NLP) has as goal building systems that can understand human language so that they can communicate with us. Since the '90s, several NLP tasks have been addressed by Machine Learning techniques [7, 30]. Machine Learning algorithms operate on an attribute-value setting, where, in most of the cases, these attributes are associated with a numerical value. Thus, when putting together ML and NLP, it has become a standard practice to represent the symbolic elements of the language, namely, the words, sentences, or even entire documents, as numbers [33, 43, 26].

One of the first ways to represent text as numeric vectors were by employing the BoW (Bag of Words) representation. Nowadays, it has become common to train vectors to represent the words, or even whole documents, relying on neural networks [35]. Also, we briefly explain how the topics associated with the speeches can be automatically discovered using the generative probabilistic model LDA (Latent Dirichlet Allocation) [6]. In this Section, we briefly describe the vectorization techniques we relied on in this dissertation to build the proposed framework.

2.2.1 Bag of Words and Term Frequency-Inverse Document Frequency

The simplest way of vectorizing a dataset composed of textual elements is to transform the documents into a set of tokens *e.g.*, the words and consider each one of them as an attribute. The Bag-of-Words (BOW) approach [16] considers this technique. Next, it is necessary to establish a value associated with each attribute, so that an example (*i.e.*, a sentence or a document) becomes a vector of such values to feed a machine learning task. Such values can be simply Boolean variables, indicating the presence or the absence of a word in the example, or they can be numeric, computed from a frequency measure of the words.

Particularly, the vectorization of documents using the Term Frequency-Inverse Document Frequency (TF-IDF) [43] measures the importance of the words in a document by computing the frequency that a word appears in it (Term-Frequency (TF), where the word is the term), but taking into account the existence of very frequent words in the documents (*e.g.*, words such as 'and', 'so', *etc*) to reduce the weight of them. Thus, the relevance of a word increases proportionally to the number of times it in the document (the frequency), but it is offset by the frequency of the word in the entire corpus (IDF – the Inverse Document Frequency). The resulting TF-IDF value is computed from the product of these two measures, as showed in the Equation 2.2, where the final value is normalized between 0 and 1, t is the term, d is the document, and N total number of documents.

$$tfidf_{t,d} = tf_{t,d} \times \log\left(\frac{N}{df_t}\right) \tag{2.2}$$

where

$$tf_{t,d} = \frac{Number\ of\ times\ t\ appears\ in\ a\ document\ d}{Total\ number\ of\ terms\ in\ d}$$

and

$$df_t = Number\ of\ documents\ with\ term\ t\ in\ it.$$

2.2.2 Words and Documents Embedding

Until a few years ago, *Bag-of-words* technique and *Bag-of-n-grams* [16, 48] were the most used ones to transform texts into a set of attributes. However, because BOW relies on frequency measures it may fail in situations where the goal involves capturing semantic aspects of the texts.

One of the causes of such failures is that a word may have many semantic aspects that are difficult to be manually defined. Thus, assuming, for example, that the values associated with an attribute can range from 0 to 1, the "king" and "queen" tokens in a royalty context should have values close to each other, and also close to 1. On the other hand, considering the gender context, the values associated with those same two words should be distant from each other, since they are related to different genres. Regarding the food context, these words should have shallow values, although close to each other. Thus, defining a set of attributes that generalize over several contexts, and assigning appropriate values to such attributes, is a difficult task to perform manually and error-prone, mainly due to subjectivity.

To work around this problem, it has become a standard practice to use a numeric vector to represent the tokens extracted from texts. Such representations are known as *embeddings* [10], and are usually defined as d-dimensional vectors, learned automatically from several texts. Thus, each dimension of the vector may reflect a distinct context, and the value associated with the dimension is learned accordingly. At the end of the learning process, it is expected that the words with the closest semantics will be mapped to close positions in the vector space.

The most commonly used implementations of such techniques are *Word2vec* [33] (which implements the Skip-gram [14] and CBOW [34] algorithms) and GloVe [38], all of them using neural networks with a hidden layer to obtain the learned representations. The vectors referring to the attributes of words are extracted from the weights of the hidden layer, making this form of learning to receive the name of neural language models [3]. The purpose of the neural model learning is to maximize the value of:

$$\frac{1}{T}\sum_{t=k}^{T-k}\log p\left(w_t|w_{t-k},\ldots,w_{t+k}\right) \tag{2.3}$$

Where w_i represents a word in a sequence of words $w_1, w_2, \ldots, w_T$, and $w_{t-k}, \ldots, w_{t+k}$ represents a window of words of size t, where $w_{t-k}, w_k, w_{t+k} \subset w_1, w_2, \ldots, w_T$. Each prediction task is usually defined as a *softmax* classifier, as follows:

$$p\left(w_t|w_{t-k},\ldots,w_{t+k}\right) = \frac{e^{y_{wt}}}{\sum_i e^{y_i}} \tag{2.4}$$

Where y_i is the non-normalized logarithm of the word probability i be the output of the model, calculated as:

$$y = b + Uh\left(w_{t-k},\ldots,wt+k;W\right) \tag{2.5}$$

Where U and b are the weights of the classifier and h is either the concatenation or the mean of the word vectors in W. The neural language models are trained with the gradient descent optimization method, where the gradient is obtained from the Backpropagation algorithm [41].

Since one of the purposes of this dissertation is to detect similarity between entire speeches, the ideal is that they can be arranged directly in a vector space, in the same way as the Word2vec method does with words. Thus, it becomes possible to tackle the speeches as documents and check the ones that are close to each other, according to a distance metric applied to vectors. For that, we benefit from the *Doc2Vec* model [26], whose main function is to create vector representations for fragments of texts, regardless of their size.

This method is based on the same learning models of vector word representations, but, in addition to the word vector matrix W, an array of D vectors is also trained. Thus, Equation 2.6 is rewritten as follows:

$$y = b + Uh\left(w_{t-k}, \ldots, wt + k; W, D\right) \tag{2.6}$$

There are two implementations of Doc2Vec, $PV - DM$ (Figure 2.1) derived from the $CBOW$ method, and $PV - DBOW$ (Figure 2.2), derived from $Skip - gram$ of Word2Vec methods. The $PV - DBOW$ model, in particular, receives the document matrix as input and returns words that are associated with the document as output. In this case, the vector $d_i \in D$, where D is the set of documents used for the training, associated with the document can be seen as a new word, which will be shared between all the contexts from the same document, but not from all the documents. The matrix of vector words W, on the other hand, is shared among all documents.

To use the tool, in the presence of a new document $d_k \notin D$, it is necessary to execute the descending gradient method to obtain the representative d_k vector. To do so, a new column is added to D, and the vectors in D are adjusted following the gradient, but keeping U, W and b fixed. In order to check if two documents are similar to each other, a distance measure between their respective vectors must be computed.

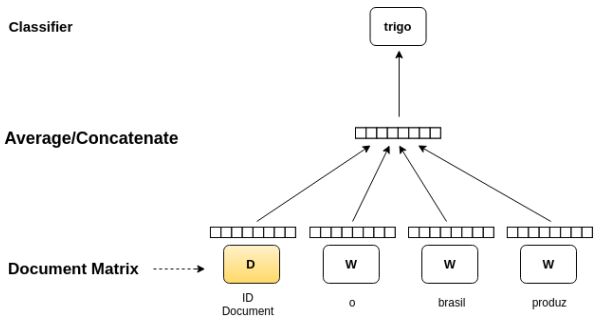


Figure 2.1: Distributed Memory (PV-DM)

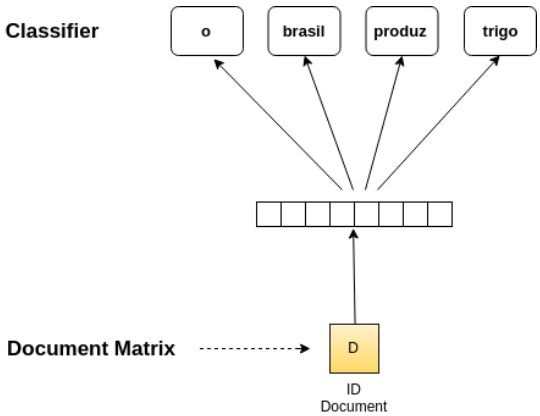


Figure 2.2: Distributed Bag of Words (PV-DBOW)

2.2.3 Topic Modeling with Latent Dirichlet Allocation (LDA)

Topic modeling computes extrapolations from a collection of documents to infer the topics that could have generated the documents [46]. Latent Dirichlet Allocation (LDA) [6] is

a technique for topic modeling that identifies the underlying topics by assuming they are latent variables included in a collection of documents.

In LDA, each document is assumed as a mixture of several topics, where the topic can be modeled as a collection of words that have different probabilities of appearing in the parts of the document that are related to the topic. In this case, the words are assumed as random variables independent from each other but conditioned on the topic. It is also assumed that the order of the words that represent a document does not matter as long as the main topics that "generate" the appearance of words are known. LDA assumes that each word in each document comes from a theme and the topic is selected from a distribution per document on topics, yielding the two matrices as computed in Equations 2.7 and 2.8.

The probability distribution of topics in documents can be calculated as:

$$\Theta td = P(t/d) \tag{2.7}$$

The probability distribution of words in topics can be calculated as:

$$\Phi wt = P(w/t) \tag{2.8}$$

where the probability of a word given document is defined according to Equation 2.9.

$$P(w/d) = \sum_{t \epsilon T} p(w/t, d)p(t/d) \tag{2.9}$$

.

Where T is the total number of topics, and let's assume that there is w number of words in our vocabulary for all the documents. If we assume conditional independence, we can say that $P(w|t, d) = P(w|t)$ and hence $P(w|d)$ is defined as Equation 2.10:

$$P(w|d) = \sum_{t \epsilon T} p(w|t)p(t|d) \tag{2.10}$$

In the Equation 2.10, $p(w|d)$ is the sum of $p(w|t)p(t|d)$, which in turn are computed in the Equations 2.7 and 2.8. Such a product is computed as $\Theta td \cdot \Phi wt$, if we assume that the decomposition matrix of word probability distribution in the document comes from two matrices that consist of the distribution of the topics in a document and the distribution of the words in a topics, respectively.

To obtain the correct weights of such matrices, the Gibbs Sampling [18] method is

used, which is an algorithm for successively sampling conditional distributions of variables whose distribution over the states converges to the true long-term distribution. It is assumed that the matrices Θ and Φ matrices are known, and the topic assignment is defined word by word, until the probability of the data is maximized. Equation 2.11 shows how to compute the conditional probability distribution of the assignment of topics as a single word, conditioned to the rest of the assignments of topics:

$$p(z_{d,n} = k/\bar{z}_{-d,n}, \bar{w}, \alpha, \beta) = \frac{n_{d,k} + \alpha_k}{\sum_i^k n_{d,i} + \alpha_i} \frac{v_{k,w_{d,n}} + \beta_{w_{d,n}}}{\sum_i v_{k,i} + \beta_i} \tag{2.11}$$

where:

$n_{d,k}$ is the number of times that the document 'd' uses the topic 'k'.
$v_{k,w}$ is the number of times that topic 'k' uses the given word.
αk is the Dirichlet parameter per document to topic distribution.
βw is the Dirichlet parameter per topic to word distribution.

There are two parts two this equation. The first one tells us how much of each topic is present in a document. The second part tells how much each topic is close to a word. Note that, for each word, we get a vector of probabilities that explains how likely this word belongs to each one of the topics. In the last Equation, the Dirichlet parameters also act as smoothing parameters when $n_{d,k}$ or $v_{k,w}$ is zero.

2.3 Smith-Waterman

The Smith-Waterman algorithm [44] was initially used in bioinformatics to calculate the *local alignment* of two nucleotide sequences or two protein sequences (DNA), i.e. to determine similar regions between two sequence chains. In this dissertation, this algorithm is used to align text sequences to calculate the semantic similarity and the thematic order of a text. This algorithm is based on dynamic programming, the operation of the algorithm consists of two phases: (1) calculate the dynamic programming matrix, and (2) obtain the final alignment. The first phase receives the two input sequences S_0 and S_1, with size $|S_0|$ = m, $|S_1|$ = n. The dynamic programming matrix is represented as $H^{m+1,n+1}$, where $H_{i,j}$ contains the score between the prefixes S_0[1..i] and S_1[1..j]. In the first phase, the first row and the first column are filled with zeros and the remaining elements of H are obtained with the equation 2.3. Besides, each cell $H_{i,j}$ contains information about the cell that was used to produce the value obtained, the highest value in $H_{i,j}$ is the

optimal score. During the alignment, there are three operations (1) insertion, (2) deletion and (3) substitution (match/mismatch), named together Penalty Gap; these operations occur when a discrepancy is found in the segment. In the second phase (traceability), the optimal local alignment is obtained, using the outputs of the first phase with respect to the scoring system that is being used during the editing of a sequence. The calculation starts from the cell that has the highest value in H, following the route that produced the optimal score until the zero value is reached.

$$H(i,j) = max \begin{cases} 0 & \\ H(i-1,j-1) + s(x_i, y_j) & Match/Mismatch \\ H(i-1,j) - s(x_i, -) & Deletion \\ H(i,j-1) - s(-, y_j) & Insertion \end{cases} . \tag{2.12}$$

where H(i, j) is the maximum similarity score between the suffix of $x[1...i]$ and the suffix of $y[1...j]$, and $s(c,d)$, c, d $\in \sum \cup \{'-'\}$, $'-'$ is the score scheme for gaps.

2.4 Topic Sequence Alignment (TSA)

As seen in the previous sections, the similarity calculation is based on vector-based representations of the documents. Although BoW based topic modeling manages to capture the topics of documents, they do not manage to represent a thematic flow (order of the words). The BoW approaches give a hint of incorrect similarity, for example, two sentences can illustrate this, "John loves dogs, but is scared of the cat." and "The cat loves John, but is scared of dogs." Although both the sentences express the relationship between John and pet animals, yet they are not semantically similar. Thinking about this inconvenience, we based on work [29], which propose the importance of the thematic flow when calculating the semantic similarity of documents.

For calculate the similarity metric related to this algorithm, the step of applying lemmatization is added to our existing pre-processing defined in Section 3.1, because without this step the thematic flow of similar sentences might seem different even if they have the same semantic content. Topic sequence alignment makes use of different models, such as topics modeling, word embedding, and the local alignment algorithm (Smith-Waterman). The workflow is explained in four main components. i) Topic-Sequence Inference level, which is responsible for transforming the words flows in thematic flows

and split sequences into sub-sequences segments, where we propose a heuristic for the segmentation discourses without signs of punctuation. ii) Token level, which calculates the similarity between topics. iii) Sentence level, the similarity between sequence segments is calculated, making use of the similarities of the topic. iv) Document-level, the similarity of the documents is calculated using mainly the Sentence level.

A document similarity measure for TSA is defined as $\sigma : \bar{D} \times \bar{D} \rightarrow [a, b]$ where $\bar{D}$ represents the document space: $a \in \mathbb{R}$ is the lower limit score of similarity, and $b \in \mathbb{R}$ is the upper limit score of similarity.

2.4.1 Inference Level

This component is responsible for inferencing the topics distribution of the two input documents using the trained LDA model. Each document is represented as $D_i = [p_1, p_2, p_3, ..., p_n]$, where p_j is a topic returned by the topics distribution. Next the words sequence from a document is converted into a thematic flow; that is, each word is assigned to the most likely topic. For this operation, the inverted topic word distribution index is used, which maps each word of the vocabulary to the topics, together with the probability of that word in the corresponding topic.

Additionally, this module divides a topic sequence into topic-sequence segments, where a segment represents a sentence, initially each document is just a sequence $D = \langle S_{j;1} \rangle$, where $S_{j;i} = \langle \hat{t}_{x;j;i} | \hat{t}_{x;j;i} \in \{t_1, t_2, ...t_n\} \rangle$ is the topic-sequence segment corresponding to the j^{th} sentence in D_i. The sentence segmentation is important to capture the discourse-level locality of a semantic similarity match. Besides, in long topic-sequences without sentence segmentation, early penalty due to sentence mismatches propagates cumulatively. Hence, the sentence segmentation are better than have longer topic-sequences. For segmenting texts, the original work used the `Spacy`[1] library, which for informal texts calculates the limits of sentences based on the analysis of dependency syntax and use the punctuation marks.

2.4.2 Token Level

This component calculates compensation whenever a topic-to-topic mismatch occurs while computing the alignment score between two sub-sequence. To calculate the similarity between topics it is necessary to have a words vector representation in the space for each

[1] `https://spacy.io/api`

topic. The topic model as we know generates a list of most likely terms that represent each topic. For example, if the top-4 words of three topics t_1, t_2, t_3 are ["lion", "cub", "flesh", "wild"], ["insect", "ants", "forest", "ferns"] , and ["kindergarten", "toddler", "alphabets", "cubs"]; the score for the (t_1, t_2) pair should be higher than that of the (t_1, t_3) pair. In this work, the first 15 top terms of each topic were taken, and their representation vectors of each word of the Word2Vec model were obtained. For obtaining the topics vectors, the mean between values vectors belonging to the topic is calculated as shown in the equation 2.13. After obtaining the vector representation of each topic, the similarity of topic-to-topic is calculated. To not do the same calculation for each query, a similarity matrix of topics is created, using the cosine similarity (Equation 2.1). This matrix is used in the sentence level, specifically in the calculation of sequences alignment in charge of the Smith-Waterman algorithm.

$$Vec\_topic(t_i) = \frac{1}{n}\sum_{j=0}^{n} Enc\_word(w_{ij}, m\_w2v) \tag{2.13}$$

Where n are top-terms of topic, w_{ij} is a word to mapping in the Word2Vec matrix m_w2v, Let Enc_word: $(w_{ij}, w2v) \rightarrow \vec{w}_{ij}$, where $\vec{w}_{ij} \in$ m_w2v is an encoding function, mapping word tokens to their vector representations for i^{th} topic, where i $\in [0, n]$.

2.4.3 Sentence Level

In this component, after segmenting document in topic-sequence segments, the similarity between two topics sequences is calculated, this calculation is based on an adaptation of the Smith-Waterman algorithm. As output, we obtain a similarity value between two sentences in a range [0, 1], this component uses the topics similarity matrix, calculated in Token Level. The sequence could have a coincidence or a mismatch between the tokens of the compared sequence, and each coincidence accumulates a reward in the final score. If there is a mismatch, there are three types of edition: Insertion, Deletion, and Substitution. Each of these edits comes with a gap penalty (i.e. cost of edit) belonging to the original Smith-Waterman algorithm. In addition to the gap penalty is added the result of the comparison of each topic-sequence token. The sequence alignment algorithm is defined by the following Bellman equations:

$$V(x,y) = \begin{cases} 0 & if \quad x = 0 \; or \; y = 0 \\ max \begin{cases} 0, & V\left(x-1,y-1\right) + M \quad if \quad \acute{t}_{x;i;a} = \acute{t}_{y;j;b} \\ max \begin{cases} 0 \\ V(x-1,y) + S(x,y,Del) & if \quad \acute{t}_{x;i;a} \neq \acute{t}_{y;j;b} \\ V(x,y-1) + S(x,y,Ins) & if \quad \acute{t}_{x;i;a} \neq \acute{t}_{y;j;b} \\ V(x-1,y-1) + S(x,y,Sub) & if \quad \acute{t}_{x;i;a} \neq \acute{t}_{y;j;b} \end{cases} \end{cases} \end{cases} \tag{2.14}$$

After to define the Bellman equation, we need to define: $s_{i;a}$ is the i^{th} topic-sequence segment of document D_a, and $\hat{t}_{x;i;a}$ is the token on x^{th} position in $s_{i;a}$, and $\hat{t}_{x;i;a} \in \{t_1, t_1, t_1, ...t_n\}$. The $value(\hat{t}_{x;i;a}, \hat{t}_{y;j;b})$ is the cumulative alignment score assigned to the topic sequence segments till x^{th} token in $s_{i;a}$ sequence, and x^{th} token in $s_{j;b}$ sequence. Besides, $x \in [0, m_i]$ and $y \in [0, n_j]$, m_i and n_j are lengths of $s_{i;a}$ and $s_{j;b}$ respectively, and M is the Match Gain (i.e. reward for a match). For better readability, we will hereupon refer to the function $value(\hat{t}_{x;i;a}, \hat{t}_{y;j;b})$ as $V(x,y)$, and $Score(\hat{t}_{x;i;a}, \hat{t}_{y;j;b}, op)$ as $S(x,y,op)$.

The *Score* is assigned by the sequence alignment algorithm when comparing two tokens of the sequence:

$$Score(\hat{t}_{x;i;a}, \hat{t}_{y;j;b}, op) = G_{op} + (f \times Topic\_similarity(t_i, t_j)) \tag{2.15}$$

Where f is a discount factor for the similarity score that balances the effect of the gap penalties and topic-topic similarity $f \in [0, 1]$. G_{op} is Gap Penalty fo an edit, and $op \in$ [Ins, Sub, Del]. Finally, the sentence similarity is defined:

$$Sentence\_similarity(s_{i;a}, s_{j;b}) = V(m_i, n_j)/(M \times max\{m_i, n_j\}) \tag{2.16}$$

In this application case, the parameters were defined based on preliminary, the following parameters for compensation factor were used: match gain = 1.75, insert penalty = -0.4, delete penalty = -1.2, and substitute penalty = -0.4.

2.4.4 Document Level

In this last component, the similarity for the alignment between documents is calculated, is presented in different ways to do, but always using the Sentence level. For example, the alignment between documents segments using the sequence segmentation proposed in

section 2.4.1, and the alignment between documents without segmentation.

2.4.4.1 One Sequence and Average Between Sequences Segments

The alignment of the whole topic sequence of a document is calculated, that is, the single sequence is not segmented and is considered as a single sentence, for this reason, the alignment of the sentence level is used.

For average between sequences, each sub-sequence of the D_a is aligned with all the sub-sequences of the D_b using the Sentence Level. For each iteration, a score that is added, and at the end of the iteration, it is divided for $a \times b$, to obtain the similarity of the document.

2.4.4.2 Root Mean Square Deviation (RMSD)

The first step is to calculate the highest alignment scores for each D_a sequences segments with the D_b sequences segments, after the maximum scores or best matches list obtained, the RMSD distance is applied, shown in the equation 2.17.

$$RMSD_{maxs} = \sqrt{\frac{\sum_{i=1}^{N} \delta_i^2}{N}} \tag{2.17}$$

Where δ is the max distance between sub-sequence i and either a reference structure of the N equivalent sequence segment and N is the sub-sequence numbers.

2.4.4.3 Smith-Waterman Alignment in Documents

The speeches usually contain more than one sentence in a document. For this reason, the same Smith-Waterman sequence alignment algorithm is applied, defined earlier in the Subsection 2.3. But now, this is done over a sequence of topic-sequence segments, i.e., between the sentences of the two documents. During the process of sentences, alignment is used the topic-to-topic matrix similarity calculated in the token level. Unlike the sentence level, it is almost impossible to find documents with the same sentences. Therefore, it is called directly to the function that calculates the sentences alignment. Similar to the sentence level, the Bellman equation is defined.

$$V_d(i,j) = \begin{cases} 0 & if \quad x = 0 \; or \; y = 0 \\ max \begin{cases} 0, & V_d\big(_{i-1,j-1}\big) + M & if \quad s_{i;a} = s_{j;b} \\ max \begin{cases} 0 & \\ V_d(i-1,j) + S_d(i,j) & if \quad s_{i;a} \neq s_{j;b} \\ V_d(i,j-1) + S_d(i,j) & if \quad s_{i;a} \neq s_{j;b} \\ V_d(i-1,j-1) + S_d(i,j) & if \quad s_{i;a} \neq s_{j;b} \end{cases} \end{cases} \end{cases} \tag{2.18}$$

This Bellman equation, it has some modification, D_a is the topic sequence representative of the document's text $D_a = \langle s_{1;a}, s_{2;a}, ..., s_{m;a} \rangle$, where m is the number of sentences in D_a. In addition $x \in [0, m_i]$ and $y \in [0, n_j]$, m and n are lengths of D_a and D_b respectively, and M is the Match Gain. The score punctuation is defined as:

$$Score_{Doc}(s_{i;a}, s_{j;b}) = sentence\_similarity(s_{i;a}, s_{j;b}) \tag{2.19}$$

The gap penalty was excluded, since it is highly unlikely that sentences across two documents would have the exact topic-sequence segment, the gap penalties would be disproportionately high, thereby adversely affecting the score. The $value_{Doc}(s_{i;a}, s_{j;b})$ is the cumulative alignment score assigned to D_a counted till the i^{th} sentence, and D_b counted till the j^{th} sentence. For better readability, we will hereupon refer to the function $Score_{Doc}(s_{i;a}, s_{j;b})$ as $S_d(x, y)$, and $value_{Doc}(s_{i;a}, s_{j;b})$ as $V_d(x, y)$. The final document similarity is calculated as follows, :

$$Document\_similarity(D_a, D_b) = V_d(m, n)/(M \times max\{m, n\}) \tag{2.20}$$

Where, $(M \times max\{m_i, n_j\})$ is used for linear normalization of the score.

2.5 Related Work

The majority of previous work tailored towards automatically analyzing political discourses has focused on supervised classification of citizens and candidates political positions. To that, one may start from social network content such as tweets, and, from them, to build classifiers shaped to solve sentiment analysis tasks [31, 2]. The only work using data that contains information related to presidential elections in Brazil is [9],

which focuses mainly on the automatic detection of the ideological positioning of tweeter users. There are also previous attempts of automatically detecting ideological political positions from political speeches, both using the classical Bag-of-words method and more widely-used recently Long-Short-Term-Memory networks [12, 19]. However, they rely on annotated datasets of already-elected politicians' speeches. Different from those works, the POSTURE framework developed here aims at automatically analyzing the semantic political content of *candidates* to political positions' speeches using *unsupervised* techniques.

More recently, a supervised method based on a convolutional neural network was employed to classify political speeches accordingly to a proposed taxonomy. Although the method was also applied to evaluate the discourses made during Spanish general elections, they employed a previously induced classifier built from annotated data. In this case, the method has focused on tweets posted by the political parties, while here we gather spoken speeches of varying size.

Previous works have also built methods to extract knowledge from political speeches in using unsupervised approaches. For instance, in [13] the authors proposed a dynamic topic modeling method to extract recurrent political themes from the European Parliament's political agenda. To that, they extract the topics with a two layers non-negative matrix factorization (NMF) approach. Similarly, in [11], the authors aimed at extracting the recurrent topics discussed by the candidates to the presidency of the United States in the 2016 election. Different from the aforementioned work, they made use of a hybrid method combining NMF with a sequencing algorithm (Signature model) based on linear programming. POSTURE also includes topic modeling components, yet using LDA, which, as a probabilistic method, accounts for the uncertainty and noise inherent to the spoken discourses handled in this dissertation. Moreover, we do not stop at the extraction of topics, but use them to determine coherence and similarity among the speeches.

In [1] the authors had the goal of investigating how semantic shifts in discourses appear in temporal and social dimensions. To that, they have also employed distributional semantics of words to insert the discourses into different vector spaces which are later compared using linear and graph-based mappings across those vector spaces. POSTURE assumes that all the speeches are within the same dimension which allows for computing their similarity with classical distance metrics, such as the cosine distance, without the need for mapping across different vector spaces.

Chapter 3

POSTURE Framework for Political Speeches Analysis

We show that the proposed framework can extract important political aspects from the speeches, making them more citizen-friendly and amenable to their evaluation. POSTURE can be used in future elections, besides it is not only restricted to the Portuguese language, but it can also be instantiated in different languages. The Figure 3.1 represents the processes followed by POSTURE framework that we devised in this dissertation and also is showed that POSTURE framework has four core components: (i) text pre-processing, (ii) Data vectorization, (iii) Topic Sequence Alignment, and (iv) POSTURE functionalities.

The process starts with text pre-processing, the process follows to build the vector space of words/document representations and probabilistic topic. Almost simultaneously, topic sequence alignment is calculated, using the word vectors and the probabilistic topic model. POSTURE can compare their semantics aspects using distance functions and extract the topics that the candidates are most interested in it.

The next step is to explore data statistics, such as their distribution along with the time and size of the vocabulary, and also build visualizations directly from the textual content, such as word-clouds and dispersion charts. Such visual representations may help the citizen to acquire some knowledge about the subjects that the candidates are talking about it.

Finally, based on the produced vector space representations and topic sequence alignment, the process goes to the analytical part where POSTURE offers ways of discovering the topics latent to the speeches, computing and visualizing the consistency and coherency of a candidate discourse along the time, and how the speeches of one candidate are com-

pared to the others. In the following subsections, we detail the components mentioned above together with the explanation of the methods used within each component.

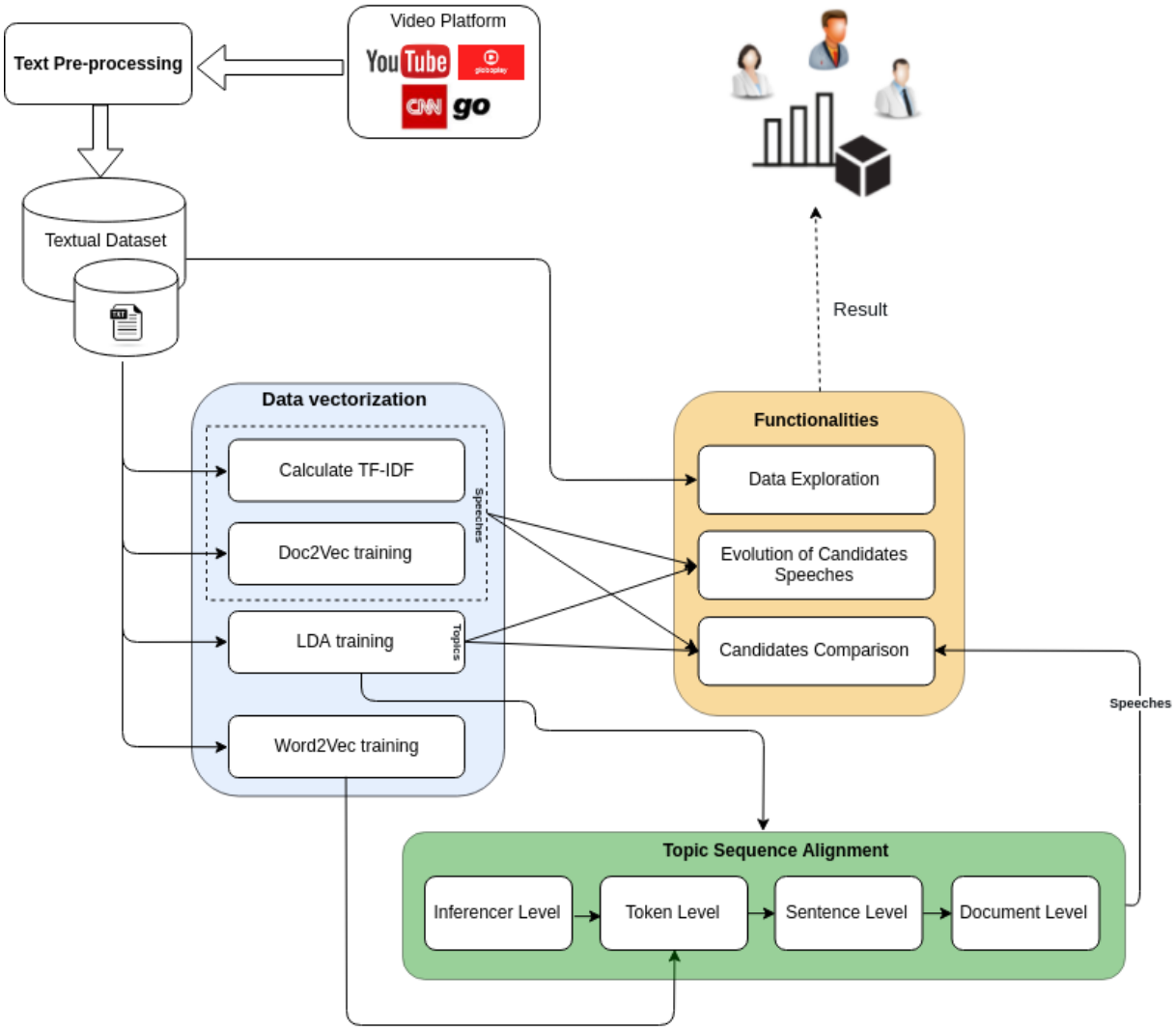


Figure 3.1: POSTURE Conceptual Architecture

3.1 Text Pre-processing

After acquiring the textual content of the speeches, this component processes the dataset to generate a normalized dataset. To that, we employ the steps as follows: (i) Word extraction from the texts (tokenization), (ii) Conversion to lowercase letters, (iii) Removal of words lacking semantic meaning (removal of stop-words, including numbers), (iv) Creation of bi-grams and three-grams to give meaning to compound words, and (v) Lemmatization. These steps were performed using the NLTK library [28].

3.2 Data Vectorization and Calculating Topic Sequence Alignment

Here, we detail how POSTURE proceeds to yield the vector space representations from the texts. It is intended to extract all kinds of information and behavior contained in the speeches, for that each used representation manages to obtain different information, such as Word2Vec the semantic relationship between words, TF-IDF manages to extract lexical relationships between speeches, Doc2vec semantic relations between speeches, and LDA extracts the latent topics conferred in the speeches. Recall from Section 2.2 where TF-IDF, which depends on the number of terms in the corpus and has as parameters the maximum and minimum frequency of words per document; Word2Vec, whose parameters are detailed in Section 4.4; Doc2Vec, also parameters are detailed in Section4.5; and LDA, which represents a document like a distribution of topics in a vectors space. The NLTK library was used for the first vectorization, and the last three vectorization types were performed using the *Gensim* [39] library. On the other hand Topic Sequence Alignment is calculated simultaneously, considering the semantic similarity and the thematic order obtained by the topics alignment using `Spacy`[1] library.

3.3 Evolution of Candidate Speeches

It is quite common that a candidate changes or maintain his/her discourse and campaign promises during the campaign to cope with political factors, to answer to other candidates attacks, and, naturally, to gain support among voters. The component described in this subsection aims at capturing such speech modifications by pointing out the ones that hold uniformity and coherency aspects, and how these issues cope with the evolution of the relevant topics throughout the campaign period.

3.3.1 Constant Discourse Analysis

To analyze if the candidate' speeches are constant, *i.e.*, there is some continuity on the main subjects during the campaign, the use of sequential time periods assumes a vital meaning. The idea is to check which candidates keep similar discourses during several months throughout the campaign period.

To discover if a speech s_i has abidance afterward, POSTURE starts with a s_i rep-

[1] `https://spacy.io/api`

resented in a vector space and finds the speeches ahead of s_i in time that are similar to it, according to a similarity threshold computed over the speeches' vectors with the cosine distance. POSTURE not only requires that there are future speeches close to s_i but, instead, it demands that there is a chained sequence of them, where each element in the chain is a speech uttered in the month that immediately follows the month of the previous element in the chain. POSTURE adopts a monthly index because the speeches within the same month are already expected to be very close, while we would like to observe if the similarity is maintained in the next months. Thus, we may have a sequence of size n represented as $s_i^{m_k} \rightarrow s_i^{m_{k+1}} \rightarrow \cdots \rightarrow s_i^{m_{k+n}}$, where m_k is the first considered month, and the $\rightarrow$ only indicates that m_{k+1} is the month immediately after m_k, *i.e.*, the next speech in the sequence is required to be made in the next month. To classify the candidate accordingly, POSTURE requires that such a sequence has a minimum size. It is also important to notice that POSTURE finds such a chain by selecting the forthcoming speeches that are similar to the first one in the sequence. For instance, suppose the chain $s_i^{m_1} \rightarrow s_i^{m_2} \rightarrow s_i^{m_3}$. It is required that both $s_i^{m_2}$ and $s_i^{m_3}$ are within a certain distance from $s_i^{m_1}$ but they are not necessarily within the same distance from each other. This is because the similarity metric is not a transitive function since assuming transitivity would likely mistake small changes over time as a constant discourse.

Algorithm 1 is responsible for finding all the chains of similar speeches within a period of time so that from them it becomes possible to select the candidates holding the largest number of sequences and the largest sequences, compared to the others during the same period of time. As the speeches are all distributed in a vector space, we leverage the cosine similarity previously discussed to find the speeches that are closest to s_i. For each month m in the selected period (see line 4 in Algorithm 1), the algorithm iteratively visits each speech s within m and, from each one of them, it tries to find the sequence of similar speeches at each month after m (lines 7–21). If the set of similar speeches within a month is empty, then the sequences starting with speech s have reached their end (line 22). The speeches collected at that iteration are only kept if they have more elements than the minimum required size (lines 24,25).

3.3.2 Coherence Balance of Speeches by Period of Time

This metric is inspired by the coherence-of-topics [8], which defines the degree of semantic interpretability of the terms used to describe a topic. One of the ways to calculate the coherence found in the literature is using the words vector representations [37]. That is,

Algorithm 1 Finding all the sequences of speeches through time from a candidate

Input: S_i, a list of vector representations of the speeches issued by the candidate i, $M = (m_p, m_q)$ the initial and final month to be verified, min_sim, a minimum similarity threshold to indicate that two speeches are similar, k the number of top-similar speeches, min_seq_size, the minimum size of a sequence to consider

Output: a set of sequence of speeches $seq\_speeches$

```
1: function CONSTANT_SPEECH(S_i, (m_p, m_q, k), min_sim, min_seq_size)
2:     seq_speeches' ← ∅
3:     m_in ← m_p
4:     m_out ← m_q − (min_seq_size − 1)
5:     while m_in ≤ m_out do
6:         S_im ← get_speeches_month(S_i, m_in)  ▷ returns the speeches discoursed in month m_in
7:         sequences' ← ∅
8:         for each speech s ∈ S_im do
9:             x ← m_in
10:            sequences' ← s
11:            size ← 1
12:            while x ≤ m_q do
13:                S_i(m+x) ← get_speeches_month(S_i, x + 1)  ▷ returns the speeches discoursed in the next month
14:                similar ← most_similar(s, S_i(m+x), k, min_sim)  ▷ Calling Algorithm 2
15:                if similar is not empty then
16:                    temp_seq ← ∅
17:                    for each speech f in similar do
18:                        for each seq ∈ sequences' do
19:                            temp_seq ← seq ∪ f
20:                    sequences' ← temp_seq
21:                    size ← size + 1
22:                else
23:                    x ← m_q  ▷ forces abandoning while
24:                x ← x + 1
25:            if size ≥ min_seq_size then
26:                seq_speeches ← seq_speeches ∪ sequences'
27:            sequences' ← ∅
28:        m_in ← m_in + 1
       return seq_speeches
```

Algorithm 2 Selecting similar speeches

Input: s, a speech representation as a vector, *speeches*, a set of speeches, k, a maximum number of similar speeches to be selected, *min_sim*, a similarity threshold to assume that two speeches are similar to each other

Output: *similar*, a list of speech(es) that are marked as similar to s

```
1: function MOST_SIMILAR(s, speeches, k, min_sim)
2:     similar ← ∅
3:     speeches ← speeches − s                ▷ avoiding consider the speech itself
4:     for each s' ∈ speeches do
5:         if cosine_similarity(s, s') ≥ min_sim then
6:             similar ← similar ∪ s'
7:     similar ← sort(similar)                ▷ Sort from the most similar to the less similar
8: return the first k elements in similar
```

if the words that represent a topic (k-top terms) are close in their semantic vector space, the coherence of topic is higher than when are dispersed. In this section, we propose the analysis of coherence, but at the level of documents. We define political balance coherence as to how a candidate maintains his/her campaign promises and discourses in each month without abrupt variations of coherence throughout the entire campaign period. This way, it is possible to discover through coherence the variability of the speeches in each time period, compared with the other months.

It is important to note that coherence depends on the thematic semantic variation that a candidate presents. A less coherence in some period means that the candidate does not maintain his/her ideals and campaign promises, being more changeable in their speeches. On the other hand, it is interesting to note that a very high coherence is not considered useful in this analysis, since the candidate may be limited in the content of his/her speeches, being somewhat repetitive and shallow. For these two observations, it is considered optimal that a candidate should not have a very high coherence and such a low coherence, considering optimal to have a balance between the two observations. The coherence of a candidate in a period t is calculated, taking all the discourses speeches in this period, to later calculate the average similarity between all the speeches, shown in the Equation 3.1. This metric is based on the cosine distance, specifically on the average distance.

$$Coh_t = \frac{1}{N > 0} \sum_{j=2}^{N} \sum_{i=1}^{j-1} Similarity(s_j, s_i) \tag{3.1}$$

where N is the number of speeches of a specific candidate in the period of time t and s_k is the representation of a speech as a vector.

3.3.3 Topics Evolution during the Political Campaign

Throughout the campaign, there are some hypes and tendencies - that we call topic flow - of topics debated by the candidates. Each candidate discusses the topic based on his/her own experience. These topic-flows allow us to analyze the behavior of all candidates in general, in order to discover the evolution and behavior of politicians during the presidential campaign. POSTURE provides such analysis by taking the topics distribution as a graphic time series of political issues. For this process is necessary to use the vectors trained by the LDA model obtained in Subsection 4.6 that are ordered chronologically. In this case, we analyze all the vector representations of the discourses of all the candidates. Here, each document is represented with a distribution vector of topics $S_n^t = [p_1, p_2, p_3, ..., p_m]$, for example, to analyze the evolution of topic x, it can access its probability $S_n^t[x]$. In a general way, we use concepts from time series, known as rolling means or moving averages. Generally used to soften short-term fluctuations and highlighting long-term trends, a simple average was used. We calculated new values for each point using a window of "w" past speeches, to finally build a sequence of time series, detailed in the Algorithm 3.

Algorithm 3 Finding similarity for speeches as time series using topic probabilities

Input:

1: R_i^k, is a list of latent vector representations, $R_i^k = [S_1^k, S_2^{k+1}, S_3^{k+2}, ..., S_n^{k+t}]$, where k is the date where the speech S_n^t was made (list sorted by date), each speech is represented as $S_n^t = [p_1, p_2, p_3, ..., p_m]$, where m is the number of topics. w is the window taken to smooth the data.

Output: Time series list

2: **function** GET_TIME_SERIES(m, R_i^k, w)
3: *topic_probabilities* $\leftarrow \emptyset$
4: **for each** vector v in R_i^k **do**
5: *topic_probabilities* $\leftarrow v_m$
6: pos $\leftarrow$ w - 1
7: series $\leftarrow$ [zero]*n
8: **for** $i \leftarrow 0$ to n **do**
9: **if** i $\geq$ pos **then**
10: mean $\leftarrow$ Sum(topic_probabilties$_{(i-pos:i-pos+w)}$)/w ▷ Rolling mean
11: $series_i$ = mean
12: **return** *series*

3.4 Candidates Comparison

The differences between candidates play an essential role at the time a citizen decides to vote in a candidate. This component contains several metrics to show some differences between candidates speeches, such as political orientation, and the evolution of the relevant topics by candidate throughout the campaign period.

3.4.1 Similarity Between Candidates

This metric is responsible for pointing out how similar or different the candidates are to each other, according to the content of their speeches throughout the campaign period. Such similarity may emerge from them talking about similar topics, presenting similar campaign promises, expressing similar feelings towards some event, belonging to related political positions, *etc.* Hence, the similarity between the candidates is calculated by pairs of sets of speeches of the candidates, where each set of speeches per candidate is defined as $C^k = \{d_i^k, d_{i+1}^k, ...d_p^k\}$, where p represents the total number of speeches that the candidate C^k has. The similarity of speeches of a candidate with all the other candidates is calculated using the Algorithm 4, the candidate that has higher similarity to the one is chosen, as shown in the Table 5.2. One may note that calculating the average of the cosine similarity would be enough. However, as we noted in the data exploration section, the data per candidate is unbalanced, affecting the using of a simple average. For this reason, the procedure first calculates the similarity of the two pairs discourses of each candidate (see line 6 in Algorithm 4). Subsequently, the similarity is inserted in increasing order (line 8-14), this type of sorting is based on the *insertion sort* algorithm. Once the list of ordered similarities is calculated, we extract a sub-sample of size n, which will represent the most similar discourses that those candidates have, seeking to maximize the similarity between the candidates. Finally, we calculate the average similarity between the extracted sub-sample (line 15). This procedure is fairer because it has the same number of pairs of compared discourses among the different candidates, and also because of the maximization of the similarity between candidates.

3.4.2 Intuiting Political Positions

For the intuition of the positioning of the candidates, POSTURE starts assuming that the political self-positioning of the candidates is true, then it must be fulfilled that the candidates must be close to the others with the same political positioning. There are

Algorithm 4 Calculating the similarity between two candidates

Input:

S_i^a, S_j^b are lists of vector representations of the speeches from a and b candidadates, where i and j are numbers of speeches. n is a size of sub-sample most similar between candidates, where $n < i \times j$.

Output: similarity between the candidate C^a and C^b

function SIMILARITY_TWO_CANDIDATES(S_i^a, S_j^b, n)
    $list\_similarities \leftarrow \emptyset$
    for each speech s^a in S_i^a **do**
        for each speech s^b in S_j^b **do**
            similar=$Similarity(s^a, s^b)$ ▷ cosine distance (Equation 2.1)
            list_similarities ← similar
            i ← size(list_similarities)
            j ← i - 1
            while j>0 and $list\_similarities_j < list\_similarities_{j-1}$ **do** ▷ Insertion sort
                temp_speech ← list_similarities$_j$
                list_similarities$_j$ ← list_similarities$_{j-1}$
                list_similarities$_{j-1}$ ← temp_speech
                j ← j-1
    similarity ← $sum(list\_similarities_{(n:i\times j)})/n$ ▷ Average similarity
    return *similarity*

different political positions around the world that are always composed of extremes and centers. For example according to [4], the Brazilian political positioning is composed of three main groups $\{L, C, R\}$ where the combination of them form others position. In total there are five political positions composed for $Left = \{L\}$, Moderate-left = $\{ML\}$, Center = $\{C\}$, Moderate-right = $\{MR\}$, and Right = $\{R\}$. Also is defined: $L \cap R$ = Center, $C \cap R$ = Moderate-Right, $C \cap L$ = Moderate-Left. For calculate the positioning according to the similar candidates found for both the Tf-IDF and Doc2Vec vectors, POSTURE considers that the final result depends on *a priori* positions, both of the candidate to be calculated and the two similar candidates. Equation 3.2 it is composed of two parts, the positions of the similar candidate(s) affect the result in the same proportion with the political position of the candidate to be calculated. For example, to calculate the political position of Guilherme Boulos.

$$Result\_position = Priori\_position \cap (position\_doc2vec \cup posititon\_tfidf) \quad (3.2)$$

Guilherme Boulos $= Left \cap (Moderate - Left \cup Moderate - Left)$
Guilherme Boulos $= \{L\} \cap (\{ML\} \cup \{ML\})$
Guilherme Boulos $= \{L\} \cap \{ML\}$
Guilherme Boulos $= \{L\} \cap \{C \cap L\}$
Guilherme Boulos $= \{L \cap C\}$
Guilherme Boulos $=$ *Moderate-Left*.

3.4.3 Speeches with Strong Similarity Relationships

Besides finding the constant speeches focusing on each candidate during the entire political campaign (Subsection 3.3.1), POSTURE can also discover the candidates who have delivered speeches with very similar contents compared to the other candidate, during the same or different periods of time. POSTURE not only shows if speeches from distinct candidates have the same thematic content but also points out if a candidate has been completely replicated. In other words, some discourses may have the same thematic content, using the same words or using similar words (synonyms) in different or similar topics order. The thematic order in the two discourses can help to have better precision to know the degree of semantic similarity between speeches of both candidates. Consequently, the calculation consists of two steps. In the first step, the similarity calculation is performed using the vectorial representations, since it is less expensive and faster than using the Topics Sequence Alignment, also to the computational cost that the calculation of similarity of one against all requires. The reason why this step is used as a first filter to decrease the set of calculations significantly. For later in the second step, calculate the similarity considering the order sequences of topics of both discourses using the *Topics Sequence Alignment* (Section 2.4).

3.4.3.1 Calculate Similar Content

The details on how such similarity is found between two candidates are exhibited as Algorithm 5. The calculation is performed by looking for similar discourses between two candidates. The discourses are compared as a pair and the pair that has a similarity value higher than a threshold is added to a list. The algorithm does not look at repeated pairs of discourses because it iterates from back to front, visiting candidate by candidate and discourse by discourse, as can be seen in the lines (4-7). To generate the results, the value of the similarity threshold is as 0.6 for Doc2vec and as 0.35 for TF-IDF. We experimented

with different thresholds because TF-IDF has difficulties in finding similar discourses with a higher threshold.

Algorithm 5 Discovering Speeches with Strong Similarity Relationships

Input:

1: S_i, is a list of candidates vectors representation, $S_i = [C_1, C_2, C_3, ..., C_n]$, where n is number of candidates, each candidate is represented $C_j = [d_1, d_2, d_3, ..., d_m]$, where m is number of document speeches by candidate. min_sim, a minimum similarity threshold to indicate that two speeches are similar.

Output: *strong*, a list with pairs of strongly similar speeches

```
2: function STRONG_SPEECHES(S_i, n)
3:     Strongs ← ∅
4:     for i ← 1 to n do
5:         for each candidate j in S_i do
6:             for k ← i + 1 to n do
7:                 for each p in S_k do
8:                     similar ← Similar(j, p)
9:                     if similar ≥ min_sim then
10:                        Strongs ← similar
11:    return Strongs
```

3.4.3.2 Calculate the Semantic Order

This second step takes as input the pairs of discourses found in the first step, both for the vector space TF-IDF and Doc2Vec. POSTURE applies the topic sequence alignment (Section 2.4) on the pairs speeches obtained it. In the Subsection 2.2.1, we saw that TF-IDF is based on BoW that does not manage to capture the order of the words, being able to exist discourses that use the same words but express the opposite, or are in different contexts. On the other hand, Doc2Vec trained with the PV-DM architecture (Subsection 2.2.2), which in addition to training the document vectors, also trains the word vectors. Although it manages to capture the relationship between words, it does not manage to capture the order of the words in the document. The application of TSA on pairs of discourses allows us to detect which discourses have the same thematic content in the same thematic order.

3.4.4 Comparison of the Candidates Evolution of a Specific Topic

We are aware of the importance and evolution of each candidate concerning national issues along with the campaign. This information may be vital for voters to choose their candidates. Similar to Subsection 3.3.3, the topic flow also allows us to compare the behavior

of different candidates. To accomplish that, POSTURE uses the probability distribution of the topics selected at the speeches of one specific candidate, ordered chronologically. Then, the Algorithm 3 is used considering here a window of 10 speeches.

Chapter 4

Experimental Configuration

4.1 Dataset Collection

A dataset was collected at the same time as the political campaign in Brazil took place in 2018. We collected videos of speeches of different duration from 5 min up to 120 minutes approximately. It was decided to collect only videos for reliability and the difficulty of being altered, unlike collecting parts of speeches from a magazine or newspaper, these phrases may contain only important fragments or some interpretation of the person who wrote it that can be influenced for the preference towards a candidate.

It considers videos posted in video platforms such as YouTube®, Globo Play®, *etc*, the discourses originally collected were the subtitles generated automatically by the YouTube® platform. All this collection was done manually, verifying the source of the video, also that each video only speaks the specific candidate, besides checking the high quality of the audio of the video for the subsequent generation of the subtitles.

After downloading the videos, it extracts the audio in a textual format. To perform this task, the component relies on the use of two tools in a pipeline, namely, the `Downsub`[1] tool to extract subtitles into the RST format, followed by `Subtitletools`[2], to convert the RST subtitle files into plain text [3].

In total, 900 speeches were collected from the eight candidates pointed as holding more intention of votes by the preliminary surveys (there were 13, in total). They are Guilherme Boulos, Marina Silva, Ciro Gomes, Geraldo Alckmin, Henrique Meirelles, Alvaro Dias, Jair Bolsonaro, and João Amoêdo. Fernando Haddad data were not collected,

[1] `https://downsub.com/`
[2] `https://subtitletools.com`
[3] This collected dataset is available for download at `https://github.com/UFFeScience/POStURE`

because of his later formalization of his candidacy, because of the preventive prison of the official candidate for his party, the ex-president Lula da Silva. The collected speeches are organized by candidates and periods, as of September 2017 to September 2018, thus totalizing 13 months. September 2018 is the last month with collected data before the presidential elections in Brazil in October 2018. In Table 4.1 shows data 8 official candidates and 4 pre-candidates. In addition we can see the number of speeches collected by candidate, the total size of the speeches of each candidate, the tokens used in all his speeches in addition to the minimum, maximum and the average of token used in a speech for each candidate, and finally the number of stopwords used by each candidate is observed.

Table 4.1: Statistics of the data collected from the speeches of the candidates for the presidential election of Brazil in 2018

	Candidates name	Speeches numbers	Size in MB	Tokens size	Min_tokens	Max_tokens	Mean	Stopwords numbers
1	Alvaro Dias	125	2,93	411462	151	14040	3603,464	295662
2	Jair Bolsonaro	92	2,5	257209	243	17702	4472,413	280544
3	Ciro Gomes	126	5,65	450433	195	23670	7162,587	590831
4	Marina silva	77	2,39	553783	385	17677	4946,298	262064
5	Geraldo Alckmin	56	1,6	120011	224	16976	4593,017	168501
6	Guilherme Boulos	116	3,49	902486	246	14880	4773,991	371971
7	Henrique Meirelles	61	1,79	498679	172	16511	4671,934	195759
8	Joao Amoedo	100	3,07	54150	230	20561	4986,79	355752
9	Cristovam Buarque	15	0,33	60024	142	11787	3610	37512
10	Manuela DAvila	73	1,72	284988	163	16605	3808,164	191702
11	Levy Fidelix	42	0,73	277996	295	11856	2857,404	80723
12	Rodrigo Maia	17	0,37	380865	580	10424	3530,823	41760
	TOTAL	900	26,57	4252086				

4.2 Validate Similarity Model using Triplet

We produced a preliminary experimental evaluation with a subset of the dataset generated from the speeches with at most 12 minutes. Next, we created a triplet dataset $D' = triplet_1(A_1, B_1, C_1), \cdots, triplet_n(A_n, B_n, C_n)$ where $triplet_i(A_i, B_i, C_i) \in D'$ stands for two similar (A_i and B_i) speeches and one speech dissimilar to the other two (C_i). Only short speeches are selected to favor the manual annotation of those triplets since they are likely to hold only a single topic. In total, 842 triplets were generated. We computed their accuracy in D' using the Equation 4.1.

$$Accuracy = \frac{\sum_{i=0}^{t} Triplet(A_i, B_i, C_i)}{t} \tag{4.1}$$

Where t is total number of triplets, and $Triplet(A_i, B_i, C_i)$ is defined in the Equation 4.2 and the function $Similarity(x, y)$ returns the value of similarity for two candidates' speeches.

$$Triplet(A_i, B_i, C_i) = \begin{cases} 1 & Similarity(A_i, B_i) > Similarity(A_i, C_i) \\ & \quad And \\ & Similarity(A_i, B_i) > Similarity(B_i, C_i) \\ 0 & \quad Otherwise \end{cases} \tag{4.2}$$

4.3 Calculate Term Frequency-Inverse Document Frequency

We leverage BoW with TF-IDF to obtain the vector space representations of texts. Nevertheless, before computing the vectorized representations, it is necessary to define the hyperparameters of the methods that induced them. BOW with TF-IDF requires only one hyperparameter, which is the minimum word frequency in a document, set here as 1. The dimension of the vectors depends on the size of the data vocabulary which has $52,836$ non-stop words, so, we obtained vectors of 52,386 dimensions.

4.4 Word Vectors Representation Learning

There are pre-trained models in Portuguese, such as $NILC$[4], the drawback is that these models were trained with texts in both European and Brazilian Portuguese. In addition, the data mostly did not refer to political content, except for Google news, which we assume has political content. Also, the models have trained years ago without recording updated political data. We observed that a certain number of words were not part of the vocabulary of these pre-trained models. The reason why we opted to train our Word2Vec model. The model was trained with the same data obtained in the pre-processing, the best results obtained in training model it was achieved with the implementation of skim-gram, Vector Size 100, Window Size 5, Epoch 20, Min Coun 10, and Negative Sample 20.

One of the forms that exist in the literature [34] to check if a coherent model of word inlays was obtained, is to make analogies or also named sanity check. Considering that each word is a vector in space when performing basic operations such as addition and difference, we can arrive at analogies, as shown in figure 4.1. Each analogy needs to answer a question, a:b c:d where d is unknown, we must map the inlay vectors xa, xb, xc, and calculate y = xb - xa + xc, and it is the continuous spatial representation of the word that we hope will be the best answer. For validate our model, a list of analogy tests was constructed with three input words and one expected output (analogy). In total, 100 analogies were considered, and the accuracy was used as a measure. Some analogies of the test set are shown below.

- Marina_silva + rede - Jair_bolsonaro = psl
- psl + Jair_Bolsonaro – pdt = Ciro_gomes
- Ar + eolica – sol = solar
- Jair_bolsonaro + direita - Ciro_gomes = esquerda
- Ciro_gomes + esquerda – Jair_bolsonaro = direita
- petróleo + Petrobras – elétrica = Eletrobras
- pai + filhos – avo = netos

[4] `http://nilc.icmc.usp.br/embeddings`

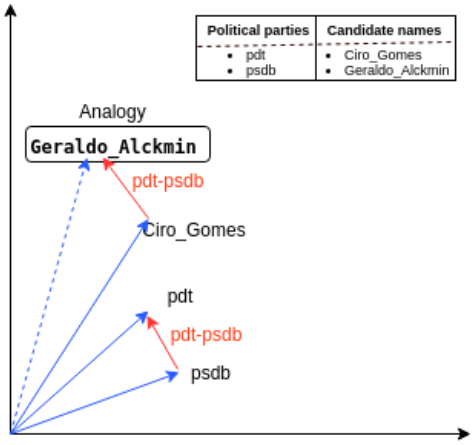


Figure 4.1: Example of analogy for Word2Vec: Pdt + Ciro_gomes – psdb = ?(Geraldo_Alckmin)

4.5 Document Vectors Representation Learning

Doc2Vec, on the other hand has several hyperparameters to be chosen in order to train the neural models, starting from the method implementation, either PV-DM or PV-DBOW. Thus, we train several Doc2Vec models using the hyperparameters recommended in [25] to select the best-trained model that manages to learn good representations of documents.

Finally, to compute the performance of the different Doc2Vec trained models using triplet validation explain in Section 4.2. The Table 4.2 shows the results obtained from the predictive results of this preliminary evaluation, where we can see that the model trained with $PV-DM$, a 200-dimension vector representation, 100 epochs to finish the training, and minimum count of 5, reaches the highest validation accuracy. Thus, we use this best model to calculate the proposed metrics in the next sections.

Table 4.2: Accuracy of evaluation of Doc2vec models, considering the use of two methods $PV-DM$ and $PV-DBOW$, the dimension of the learned vector, the number of epochs to early stopping effects, and the minimum count of frequency to ignore words

Models	Methods	Vect. dimension	#Epochs	Min Count	Accuracy
d2v_1	PV-DBOW	200	200	5	0.7517
d2v_2	PV-DBOW	200	300	5	0.7529
d2v_3	PV-DM	200	200	1	0.4085
d2v_4	PV-DM	200	200	7	0.8135
d2v_5	PV-DM	300	300	5	0.8218
d2v_6	PV-DM	200	100	5	**0.9619**

4.6 Topic modeling using LDA

The first semantic component presented here aims at automatically discovering the latent topics included within the speeches. LDA computes word distributions *per* topic and topic distributions *per* document. Thus, LDA topic model maps each document D_i in the training corpus to an n-dimensional vector space, such that $D_i = [p_1, p_2, p_3, ..., p_n]$. Here, n represents the number of topics in the trained LDA model and each value p_i represents the probability of the document holding the i-th topic.

Several experiments with a different number of topics were executed, for the choice of a suitable model. To evaluate them, POSTURE relies on the coherence measure [40] computed from LDA [5], as the higher the value of coherence, the better the human understanding is. The executions had a rank of 20 to 60 topics, with the LDA parameters α and β initialized at random. The models with the highest coherence values had about 40 to 45 topics and we selected the one with more topics (45) as this holds the potential to carry out more specific themes. From those, only for visualization purposes, in this dissertation, we select the top-8 terms and showed the 12 most relevant topics in the Table 4.3 for the Brazilian political. For a better understanding of the topics, the manual notation of each theme was made with a label that represents the content of the topic expressed by the terms that describe it (Table 4.4). The vector distribution of topics *per* document also allows us to analyze the evolution of the topics and sequence alignment.

4.7 Calculating Topic Sequence Alignment

At the inference level, we had the task of segmenting the texts into sentences, as was discussed in Subsection 2.4.1, the punctuation marks help Spacy to segment the entire text into a sequence of sentences; however, in informal texts as the ones we capture here, this is not enough. Because after to apply Spacy in our data, Spacy divide the sequence into segments with a single word and other segments have long sequence size, affecting the final similarity result. This is the reason why we propose a division of text in balanced sequences, with a fixed number of words per sentence, detailed below.

[5] We benefit from LDA implementation within the `Gensim` Library[39].

Table 4.3: The top-8 terms associated with the 12 most relevant latent topics.

Topic 2	**Topic 3**	**Topic 7**	**Topic 9**	**Topic 11**	**Topic 12**
education	companies	social	law	security	economy
teaching	petrobras	house	democracy	police	inflation
school	privatization	dwelling	judiciary	violence	central_bank
quality	privatize	occupation	justice	crime	growth
teacher	cash	living	prison	drugs	employment
fundamental	Caixa	roof	judge	intervention	conditions
middle	energy	rent	defense	federal	foresight
basic	water	families	left	weapons	develop
Topic 20	**Topic 21**	**Topic 24**	**Topic 25**	**Topic 31**	**Topic 45**
production	retirement	women	petroleum	environment	health
agribusiness	pension	abortion	Petrobras	energy	family
agriculture	millions	law	prices	crisis	program
rural	money	men	gasoline	millions	life
field	billions	rights	truckers	policies	safety
credit	tax	program	strike	problems	care
security	worker	age	fuel	development	millions
property	pays	plebiscite	crisis	social	doctor

Table 4.4: The 12 most relevant topics labeled from the terms that compose them.

Topic 2: Education in schools
Topic 3: Companies privatization
Topic 7: Housing program
Topic 9: Democracy, Lula's prison
Topic 11: Security, weapons and Military Intervention
Topic 12: National economy
Topic 20: Agriculture and agribusiness
Topic 21: Reform of social security
Topic 24: Plebiscite Abortion
Topic 25: Petroleum and Petrobras
Topic 31: Environment and energy
Topic 45: Health

4.7.1 Balancing in Sequence Segmentation

The goal of this balanced segmentation is to divide a thematic sequence in such a way that each sub-sequence is well balanced, that is, they have a homogeneous number of elements in each sub-sequence. With two sequence segment balanced, the alignment in the sentence level works better. Contrary, if there is a so minor sub-sequence aligned with another larger sub-sequence, or between two larger sub-sequences. These results depend on the gap penalty, for now, let's assume that it is the punishment of comparing two sub-sequences.

For that reason, we propose a balancing algorithm for segmenting sentences. In order to find a balance between the number of elements in each sequence segment, only indexes are considered in the segmentation, due to the importance of the order of the elements in each sub-sequence. This segmentation is defined, each document is considered a sequence segment $D = \langle t_0, t_1, ..., t_n \rangle$, that contain segments $S_{j,i} = \left\langle seg_0^{k+c_0}, seg_1^{k+c_1}, ..., seg_p^{k+c_t} \right\rangle$ where p is the number of segments, k is number of elements by segment, and $c_i = \{0, 1, 2, ..., k-1\}$ is the extra elements assigned by segments. Each segment is represented by $seg_j^{k+c} = \{t_0, t_1, ...t_{k+c}\} \in S_{j,i}$, and is delimited for $\frac{k}{2} < seg_i^k \leq k + c$. This segmentation has four cases that are exemplified in Figure 4.2, and the Algorithm 6 begins by dividing the sequence into groups with n size (lines 10-11), then, in the case that the remaining number of elements not assigned is less than half the size of segment n, it is distributed homogeneously in the existing groups. On the other hand, is considered as a new segment (line 24). In the first case, if these unassigned elements are less than the number of groups, the assignment is made element by element to each group (line 15). If the number of remaining elements is greater than the number of existing groups, these elements are distributed to each group (line 18), and if it is an indivisible number by the number of groups, the rest is assigned to the groups as well in a homogeneous way, starting from the first segment (line 22). After balancing the number of segments and the number of elements for each segment, called Algorithm 7, to divide the segments using the indexes calculated with Algorithm 6.

4.7.2 Validate Topic Sequence Alignment

The accuracy of this algorithm depends on five factors: i) The topic modeling, ii) Semantic words representation, iii) The gap penalty for sequence alignment at the sentence level, iv) Algorithm to sequences segments alignment for Document level, and v) The algorithms used for topics sequence segmentation. The last factor was not addressed in the original

Algorithm 6 Balancing in Sequence Segmentation

Input:

1: D is a thematic flow of document, is represented by $\{t_1, t_2, ...t_k\}$, where t is a topic that represents a word, and k is numbers of words of document. n is the size of sequence segments and is defined as input parameter.

Output: List of topic-sequence segments

2: **function** BALANCE_SEGMENTS(D, n, k)
3: groups ← k/n
4: unassigned_elements ← k - $n \times groups$
5: assigned_by_group ← unassigned_elements/groups
6: list_groups ← ∅
7: half_sequence ← n/2
8: **if** k/n <= 1 **then**
9: **return** D
10: **for** $i \leftarrow 0$ to $groups$ **do** ▷ Distributed n elements per group
11: list_groups ← list_groups ∪ n
12: **if** unassigned_elements < half_sequence + 1 **then** ▷ Remaining elements are distributed
13: **if** unassigned_elements < groups **then**
14: **for** $j \leftarrow 0$ to $groups$ **do**
15: $list\_groups_j = list\_groups_j + 1$
16: **else**
17: **for** $j \leftarrow 0$ to $groups$ **do**
18: $list\_groups_j = list\_groups_j + assigned\_by\_group$
19: remaining ← unassigned_elements % groups
20: **if** remaining != 0 **then**
21: **for** $j \leftarrow 0$ to $remaining$ **do**
22: $list\_groups_j = list\_groups_j + 1$
23: **else** ▷ A new group is added
24: list_groups ← list_groups ∪ unassigned_elements
25: **return** $split\_sequence(D, list\_groups)$ ▷ Call Algorithm 7

Algorithm 7 Split Sequence Using Index

Input:

1: D is a topic sequence of document. $list\_groups$ is a number of elements by each sub-sequence

Output: List of topic-sequence segments

2: **function** SPLIT_SEQUENCE($D, list\_groups$)
3: split_list ← ∅
4: x ← 0
5: **for each** number i in $list\_groups$ **do**
6: split_list ← $D_{(x:i+x)}$
7: x ← x + i
8: **return** $split\_list$

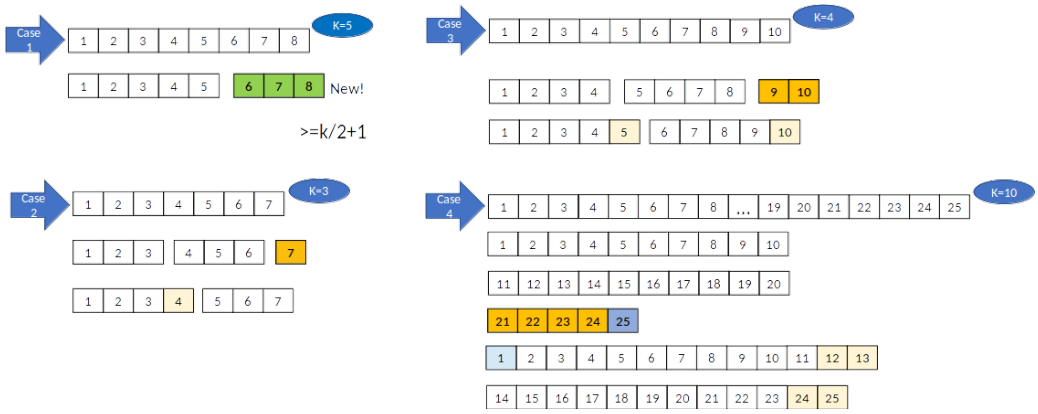


Figure 4.2: Segmentation examples

work, due to the naturalness of the data. We noticed that the penalty gap at sentence level doesn't have much relevance in the final score for this work. Therefore, we focus only on points (iv) and (v), which for our case were relevant in the final score.

As we saw in subsection 4.6, the best topic model was obtained (k = 45), which we will continue using the same topics number. On the other hand, the representation of the semantic words for the token level was validated in subsection 4.4. For validate and obtain the best TSA model, triplet dataset used detailed in Section 4.2.

To obtain the best TSA model, the most affected are the sequences segmentation and the sequences alignment between documents. Firstly, sequences segmentation using the Spacy, when applied to our problem was observed that there were segmentation errors, for example, there were sub-sequences of size one or two, which caused the gap penalty to increase and the value of the similarity between documents to decrease.

The segmentation proposed by us, as explained in subsection 4.7.1 segments the sequences into the balanced topic-sequences segment, we perceive that the size of the segments affects the accuracy. That is why it was tested with different sizes of sequences segment as shown in Table 4.5, obtaining a higher accuracy with 15 size sequences segments.

The other factor that involves the performance of TSA, is the topic-sequences segment alignment of the two input documents, as Average between sequences segments, RMSD sequence alignment and Smith-Waterman Alignment in documents. In the Table 4.6 shows the accuracies obtained with the different types of segmentation (Spacy and the proposed balanced segmentation), different uses of sequences alignment, and using the original unsegmented sequence named One Sentence. We can see that model using balanced segmentation with 15 sizes and using Smith-Waterman alignment in document reaches

the highest accuracy.

Besides, we note that the execution time of this model is longer. This is due to the calculation of heavy operations, mainly the lemmatization, and the sequence alignment. While the texts are longer, the model takes more time to calculate the similarity of the pair of discourses.

Table 4.5: Accuracy of evaluation of TSA models, using balance sequence segments with different size of segment

Static Split	Segments Size	Accuracy
SS_1	7	0.8712
SS_2	10	0.8759
SS_3	13	0.8736
SS_4	15	**0.8895**
SS_5	18	0.8652
SS_6	21	0.8052

Table 4.6: The overall accuracy of evaluation of TSA models, considering the document as one sequence, using Spacy and our proposed segmentation, and algorithms to alignment

Segmentation	Algorithm Used	Accuracy
Without segmentation	One Sequence	0.8774
Spacy Segmentation	Average between segments	0.8569
	RMSD alignment	0.8534
	Smith-Waterman in Document	0.8546
Balancing in Sequence Segmentation (15)	Average between segments	0.8764
	RMSD alignment	0.8669
	Smith-Waterman in Document	**0.8895**

Chapter 5

Results: POSTURE Functionalities

The final results of applying POSTURE are shown and detailed, these results, it is intended to inform and support the decision-making process to the voters. POSTURE tries to show its results in the simplest friendly way. First, the exploratory statistical results found when analyzing the data are detailed, and later the result of metrics proposed in the previous chapter, where some parameters are defined in order for POSTURE to analyze and obtain optimal results.

The results shown in this chapter are clearly focused on the 2018 Brazilian Presidential Election. These elections in Brazil are held every 4 years, where not only the president is elected, but also governors, senators, and members of Congress. Voting is done in electronic ballot boxes following a Two-round system: if no candidate receives the required amount of votes, then the two most voted candidates are selected for the second round of voting. In the case of the 2018 elections, the first-round elections took place on 7th October, and the second-round, on 28th October. The candidates had the deadline to formalize their candidacies from 20th July to 5th August. The formal electoral propaganda began on 16th August 2018, but the free-of-charge broadcast in radio and TV started 31st August, with 10 minutes a day per candidate.

5.1 Data Exploration

This component makes use of visualization tools in the form of graphics to present several statistics related to the collected data. This component is tailored towards getting the citizen more familiarized with some necessary information such as the frequency that the candidates talk about their campaign promises considering different periods of the time before and during the campaign, the type and comparative richness of their vocabulary,

etc.

To better visualization, some analyzes are split into several figures. For that reason for some subsections, we discuss only four candidates out of eight. Two of them are aligned to right parties (Alvaro Dias from a moderate-right party and Jair Bolsonaro from a far-right party), while two others are aligned with left parties (Marina Silva and Ciro Gomes, from moderate-left parties).

5.1.1 Distribution of the Speeches Over the Months During the 2018 Brazilian Presidential Election

Figure 5.1, Figure 5.2, Figure 5.3 and Figure 5.4 show bar plots generated by the data exploration component. First, Figure 5.1 presents a monthly distribution of the number of speeches for all the eight candidates together. It is worth mentioning that the campaign activity in the first six months is not intense since at this point we have only pre-candidates, *i.e.*, the campaigns are still unofficial. In Brazil, first, the political party chooses a pre-candidate and only three months before the election day that the candidates are officially enrolled. Note that the amount of data starts increasing in March and peaks from May to August. From the middle of August, the candidates were invited to debate in the official TV and Radio channels. This can be a reason for the decline in the number of speeches in September, which is the last month before the election day.

Figure 5.2 presents the distribution of speeches *per* candidate. Note that the candidates present different behaviors in what concerns their disposal of videos in the social media: Ciro Gomes, Alvaro Dias are the more active during the whole period, the least participatory is Geraldo Alckmin, while Jair Bolsonaro had average participation during his campaign period. Figure 5.3 shows that Alvaro Dias and Ciro Gomes start their unofficial campaign a long time before compared to Marina Silva, for example, she has not presented much activity in the digital media until April. The explanation for this behavior is that when the candidate forms the committee, it eases early fund-raising, and is seen as an unofficial announcement that a candidate is running for the presidency. Jair Bolsonaro had fewer speeches in September possibly because he was stabbed in the stomach at campaign rally what has got him away from the campaign during a short period of time.

Besides, we calculated the number of unique words used by the candidates, which is independent of the number of speeches so that we can have an idea of which candidate uses a richer vocabulary and larger variability of words in the different speeches. Ciro

Gomes was the candidate with the most diversified vocabulary, as seen in Figure 5.4.

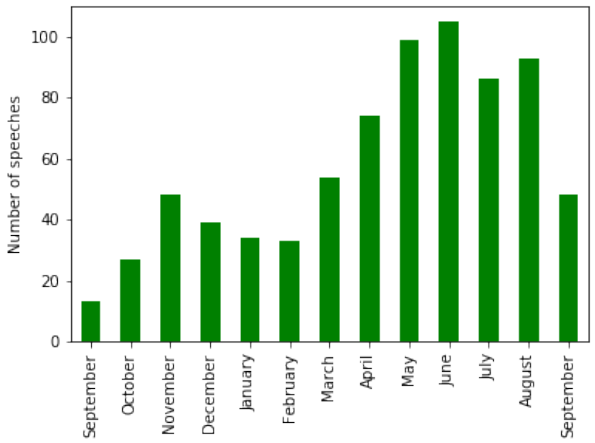


Figure 5.1: Distribution of data by time period

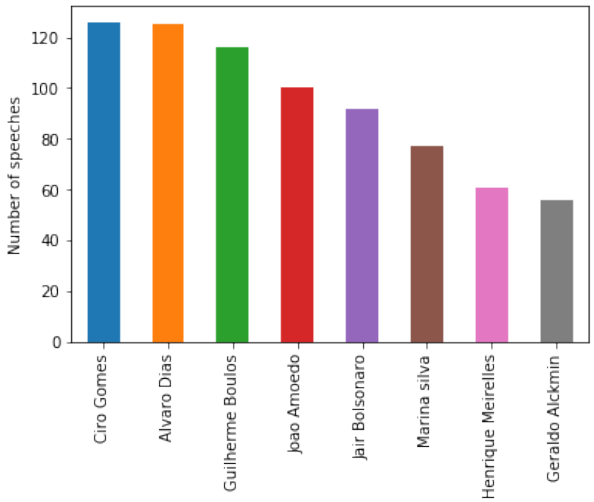


Figure 5.2: Data distribution by candidates

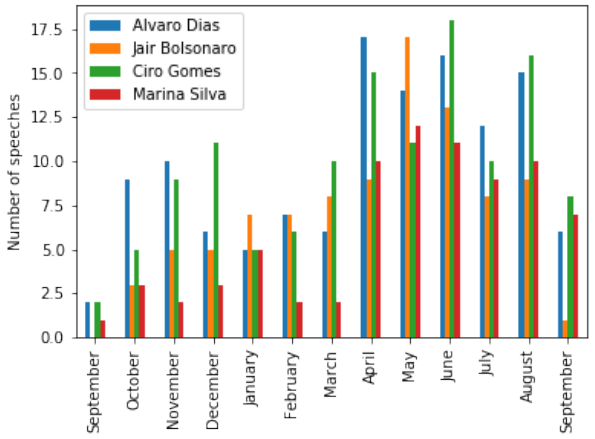


Figure 5.3: Comparison of data distribution along time

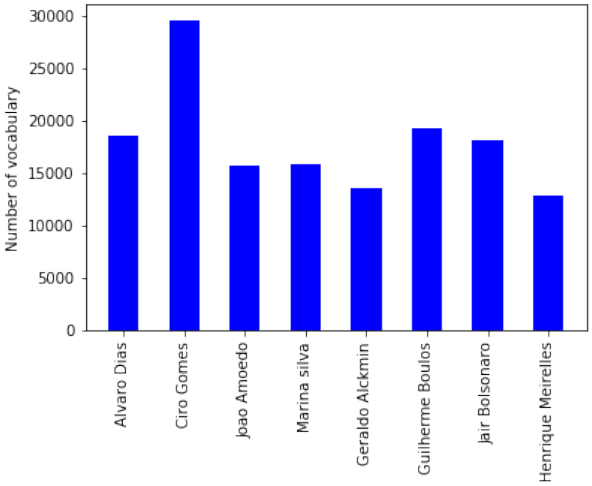


Figure 5.4: Number of unique words used by the candidates

5.1.2 Word Clouds

One way to show the most common words used in texts is with a *WordCloud*[1], which relies on heuristics to rotate and make word frequency variations easy to perceive and understand by humans. First, the component plots the word cloud considering the speeches of all candidates throughout the political campaign period as presented in Figure 5.5. There, note that candidates speak mainly of changes, proposals, programs, and investments. Besides, the emphasis is placed on issues such as economy and security. Next, the

[1] `https://github.com/amueller/word_cloud/`

component offers a way of plotting the word cloud for each candidate, individually. Here, we present the result for only the four aforementioned. Figure 5.6 presents the word cloud for the candidate Jair Bolsonaro, who talks mainly about the economy, security, family, children, and military issues. The word cloud of the candidate Ciro Gomes, in Figure 5.7, gives more emphasis on the economic issues, market, industry, banks, and health. The word cloud presented in Figure 5.8 shows that the candidate Marina Silva, in addition to talking about economic issues, also highlights security programs, respect for democracy, life, and women rights. Finally, in Figure 5.9, Alvaro Dias focuses his speeches mainly on making changes, reforms mainly like that of the Congress, besides dealing with corruption and security issues.

Figure 5.5: General words cloud

Figure 5.6: Jair Bolsonaro cloud

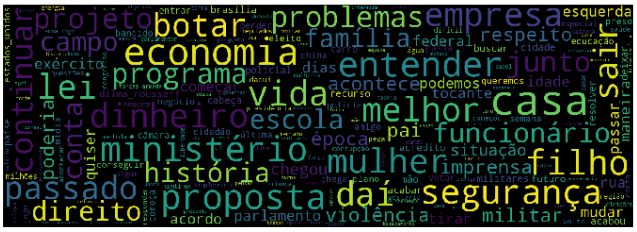


Figure 5.8: Marina Silva cloud

Figure 5.7: Ciro Gomes cloud

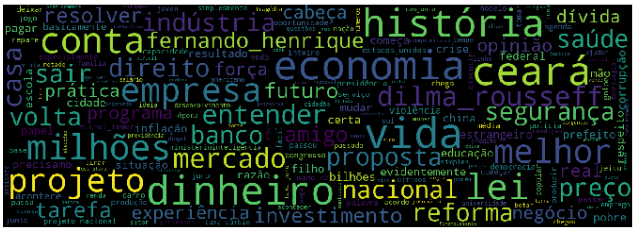
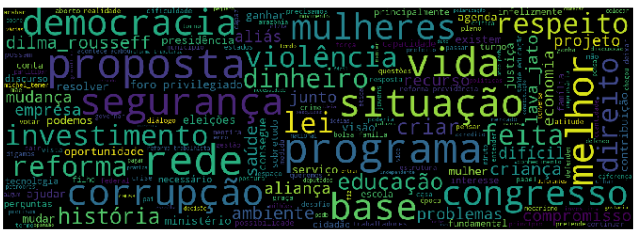


Figure 5.9: Alvaro Dias cloud



5.1.3 Scattertext Plot

For visualizing the linguistic variations in texts, POSTURE includes *Scattertex* [21] plots. This tool allows for comparing two groups of documents, by extracting the common terms shared by each group and also their different terms and plotting their statistics in a dispersion graphic.

The Scattertex tool shows in the axes X and Y the level of frequency of the use of words for both candidates, and the common terms are between both axes. Scatter displays three columns, two of which show the unique terms used by the two candidate candidates called 'Top candidate_x', 'Top candidate_y', and the column named 'Characteristic' are the common terms used by both candidates. To our study case, the groups were Ciro Gomes vs. Jair Bolsonaro, Jair Bolsonaro vs. Marina Silva, and Ciro Gomes vs. Marina Silva are showed. It was observed that the common terms between the candidates were very similar in all of the cases, such as financial issues, human rights, and proposals for changes in the country through reforms.

In the Figure 5.10 of Jair Bolsonaro vs. Ciro Gomes, it can be seen that there are many points of intersection of terms used by both candidates, among them, the most important ones are economy, financial issues, and reforms. On the other hand, the most frequent terms differing from each candidate are the topics of military police and gun possession from the side of the candidate Jair Bolsonaro, while Ciro Gomes highlights issues such as industrial development, and investment capacity compared to foreign countries.

Comparing Jair Bolsonaro vs. Marina Silva (Figure 5.11), there are fewer terms in common, while the unique terms of Marina Silva are related to the economic and political crisis, commitment, investigations of corruption, and environment issues. In the case of Ciro Gomes vs. Marina Silva (Figure 5.12), they have several intersection terms, where Marina Silva, in addition to the issues mentioned earlier, emphasizes the legalization of abortion. These observed differences and similarities reflect their political parties, as Ciro Gomes and Marina Silva are candidates of moderate-left parties, while Jair Bolsonaro is a candidate of a far-right party.

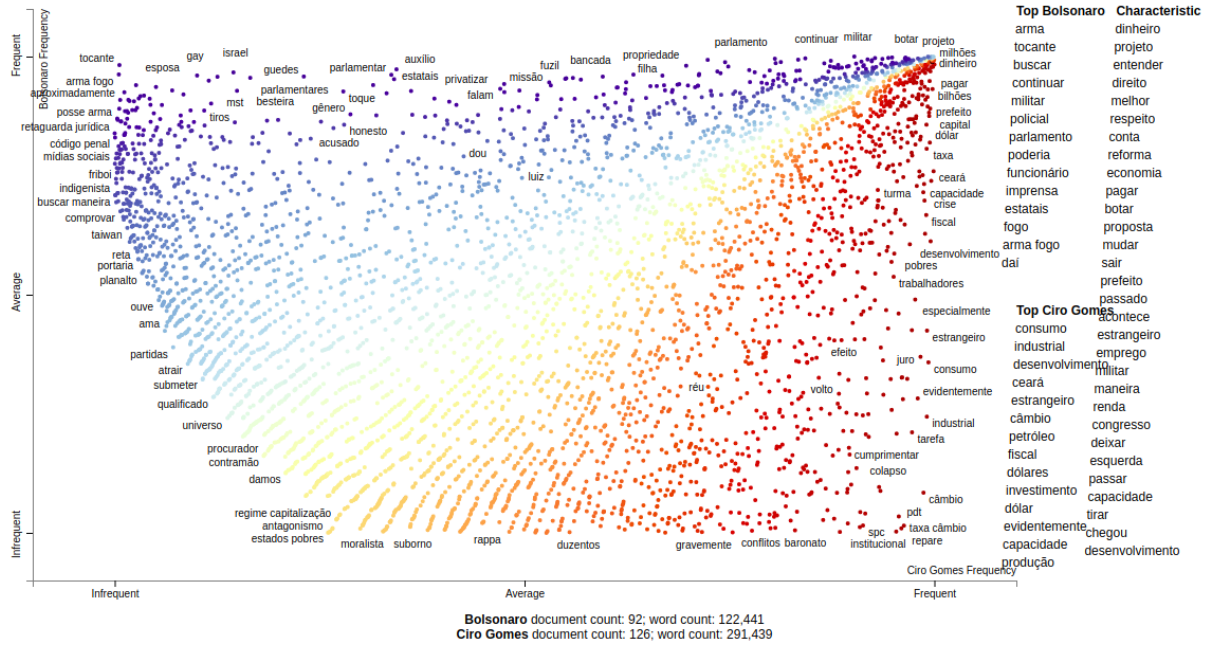


Figure 5.10: Jair Bolsonaro vs Ciro Gomes

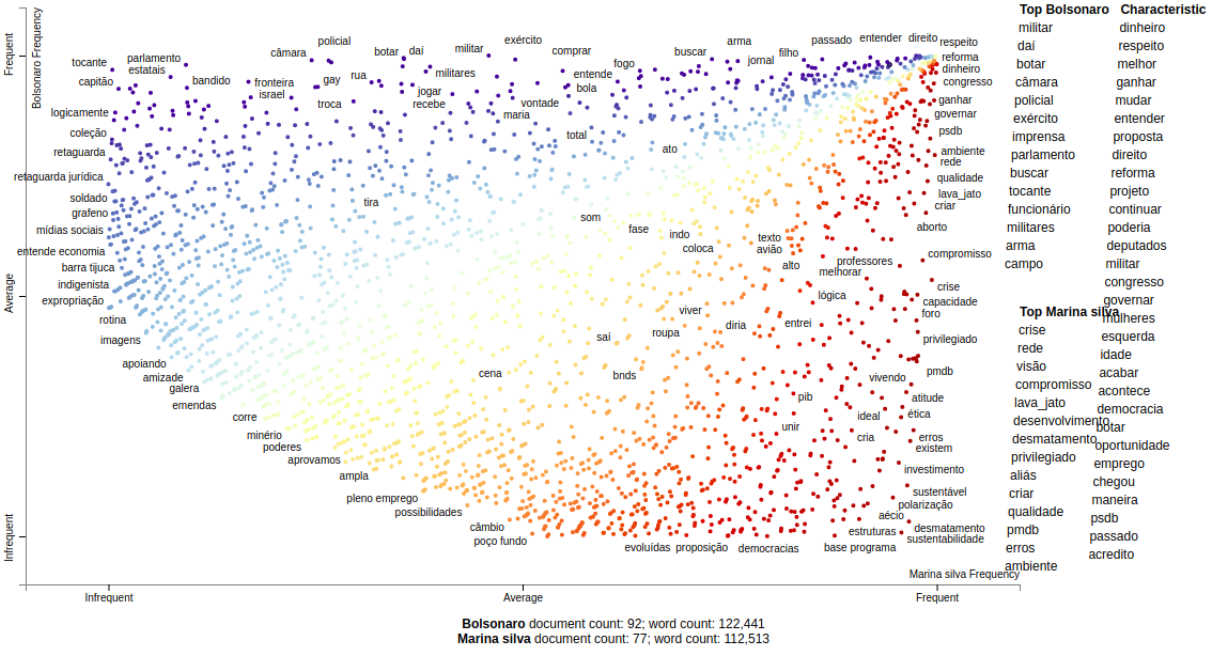


Figure 5.11: Jair Bolsonaro vs Marina Silva

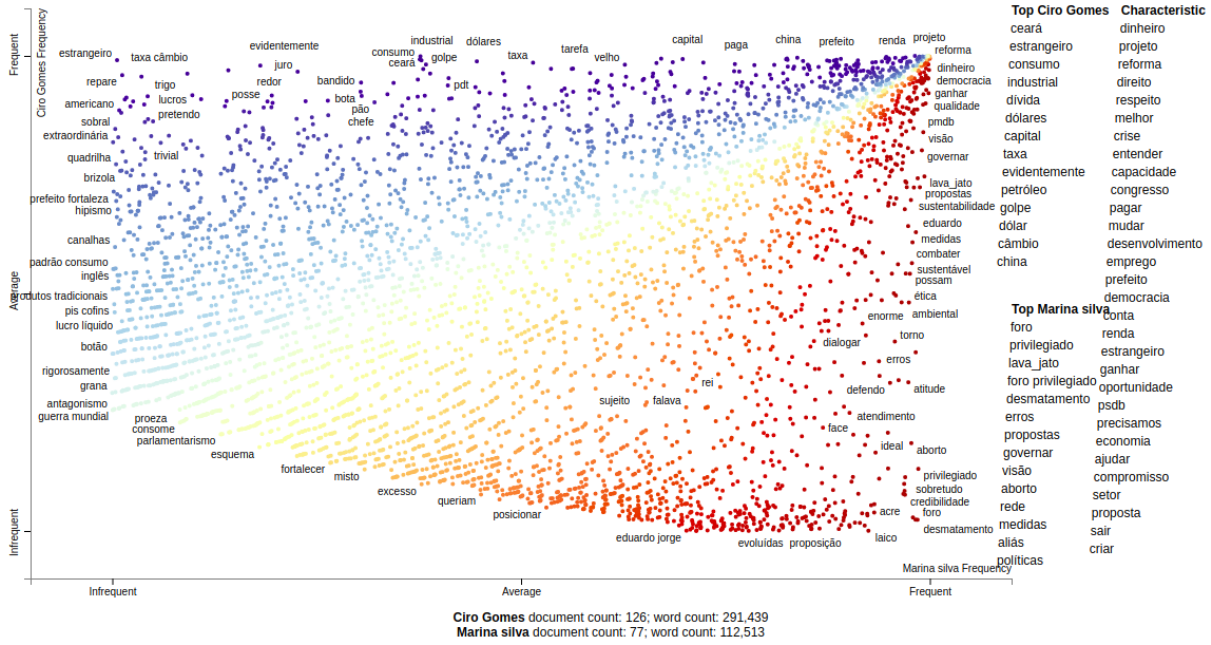


Figure 5.12: Ciro Gomes vs Marina Silva

5.2 Result of Candidate Speeches Evolution

In this section, we present the results obtained from the metrics proposed in the previous chapter, for the study of the evolution of the candidates throughout the campaign.

5.2.1 Constant Discourse Analysis

Considering our study case, we call Algorithm 1 with the speeches of each candidate in the period from October 2017 to September 2018. The value of the similarity threshold is set $min\_sim = 0.55$ for Doc2vec and $min\_sim = 0.35$ for TF-IDF, $min\_seq\_size = 3$, and $k = 10$.

In the Table 5.1 presents the number of sequences representing constant discourses calculated for all the candidates using the vectors induced with Doc2vec. Only the results of Doc2Vec are presented because TF-IDF cannot find any constant speeches with the threshold min_simi set as 0.55, only when the threshold was set to 0.35, which we considered as a very small value to state that the speeches are similar. Beside, the *Months in the sequence* represents the period where sequences with a size bigger than 3 were found, *size* is the number of elements in the discovered sequences, and *Qtd* is the number of sequences with a particular *size*.

In the Figure 5.13, there is an example of a candidate holding a sequence of size 3,

involving three months: October, November, and December, where each edge is annotated with the distance similarity metric between its first node. There are 15 sequences of size 3 related to the candidate *Ciro Gomes* and Figure 5.13 shows only one of them. On the other hand, Figure 5.14 shows the single sequence of size 4 found to the candidate *Guilherme Boulos*. Both of these cases represent constant speeches: in the first case, there is a large number of small-size speeches, meaning that the candidate is frequently repeating some subjects but not for a very long time. The second case shows a candidate that is holding his discourse for a very long time but not quite frequently.

Table 5.1: Statistics of constant speeches for all the candidates during the political campaign using Doc2vec representations.

Candidate	**Months in the Sequences**	**Size**	**Qtd**
Álvaro Dias	Jun-Aug	3	2
	Nov-Fev	4	1
Jair Bolsonaro	x	x	x
Ciro Gomes	Oct-Dec, Feb-Apr, Jun-Aug	3	15
	Jun-Sep	4	2
Geraldo Alckmin	Mar-May, May-Jul	3	2
	May-Aug	4	1
Guilherme Boulos	May-Jul, Jun–Aug	3	12
	Jun–Sep	4	7
Henrique Meirelles	x	x	x
João Amoêdo	Jun–Aug, May–Jul	3	5
Marina Silva	May–Jul	3	1

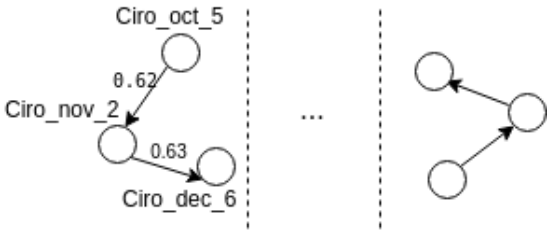


Figure 5.13: An example of a size-3-chain sequence of a frequent but small size constant discourse, computed according to Doc2Vec

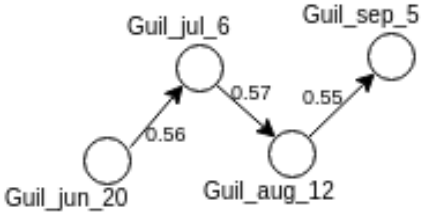


Figure 5.14: An example of a size-4-chain sequence of a less frequent but larger size constant discourse, computed according to Doc2Vec

5.2.2 Coherence Balance of Speeches by Period of Time

This metric does not need any parameter, the coherence is calculated for both the Doc2Vec and TF-IDF models. The Figure 5.15 and Figure 5.16 show that coherence reached in Doc2vec is between [0.35 - 0.57] and TF-IDF between [0 - 0.3]. One can observe that TF-IDF does not manage to calculate coherence, as the maximum observed value is smaller than the minimum value of Doc2Vec.

We believe this is due to TF-IDF be limited only to frequent relationships among words, without perceiving the semantic behind the discourses. Doc2vec shows that the greatest coherence is of Guilherme Boulos in December, while the minimum coherence is of Ciro Gomes in January. Both Ciro and Guilherme were the candidates who presented the most coherence variations during the campaign. In addition, we can see that the remaining candidates show a medium coherence over time, the most constant candidates being Joao Amoedo and Marina Silva.

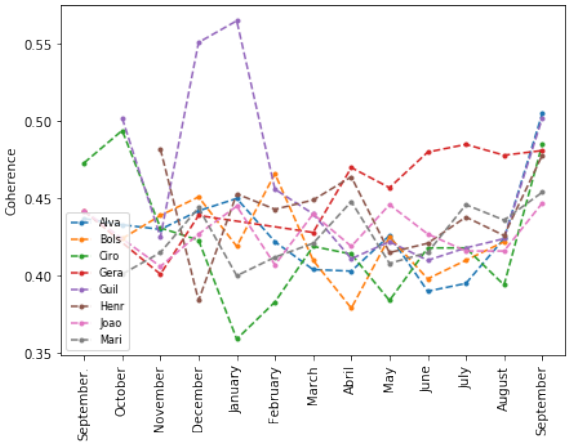


Figure 5.15: Coherence Analysis of all Candidates for each month during the entire campaign, using Doc2vec.

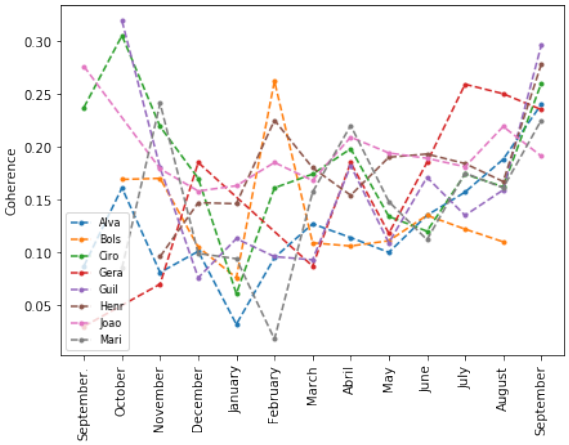


Figure 5.16: Coherence Analysis of all Candidates for each month during the entire campaign, using TF-IDF

5.2.3 Topics Evolution during the Political Campaign

This metric has only one parameter to be defined, which is the number of previous data to be considered for the construction of the time series, in this case, the previous 10 data were used. In the Figure 5.17, different flows of relevant topics are shown. We measure the amplitude of each wave so that a larger amplitude indicates that this topic was discussed with greater constancy, and the existence of high average peaks informs us the importance that the topic had in some period. We see the *"Petroleum and Petrobras"* topic achieving

the highest peaks only in June, due to the truckers' strike over the fuel crisis[2].

In the same way, the topic of *"Democracy, Lula's prison"* (former Brazilian President), has three high peaks, the first in mid-January when his judgment was taking place, and the second peak occurred at the beginning of April when the ex-president's preventive detention order was given. The last peak was in the beginning of July, where there was an intention of a judge to give him conditional freedom. The *"Security, weapons and Military Intervention"* topic was spoken almost during the entire campaign mostly due to Jair Bolsonaro, that was the only candidate that proposed to "relax" gun control.

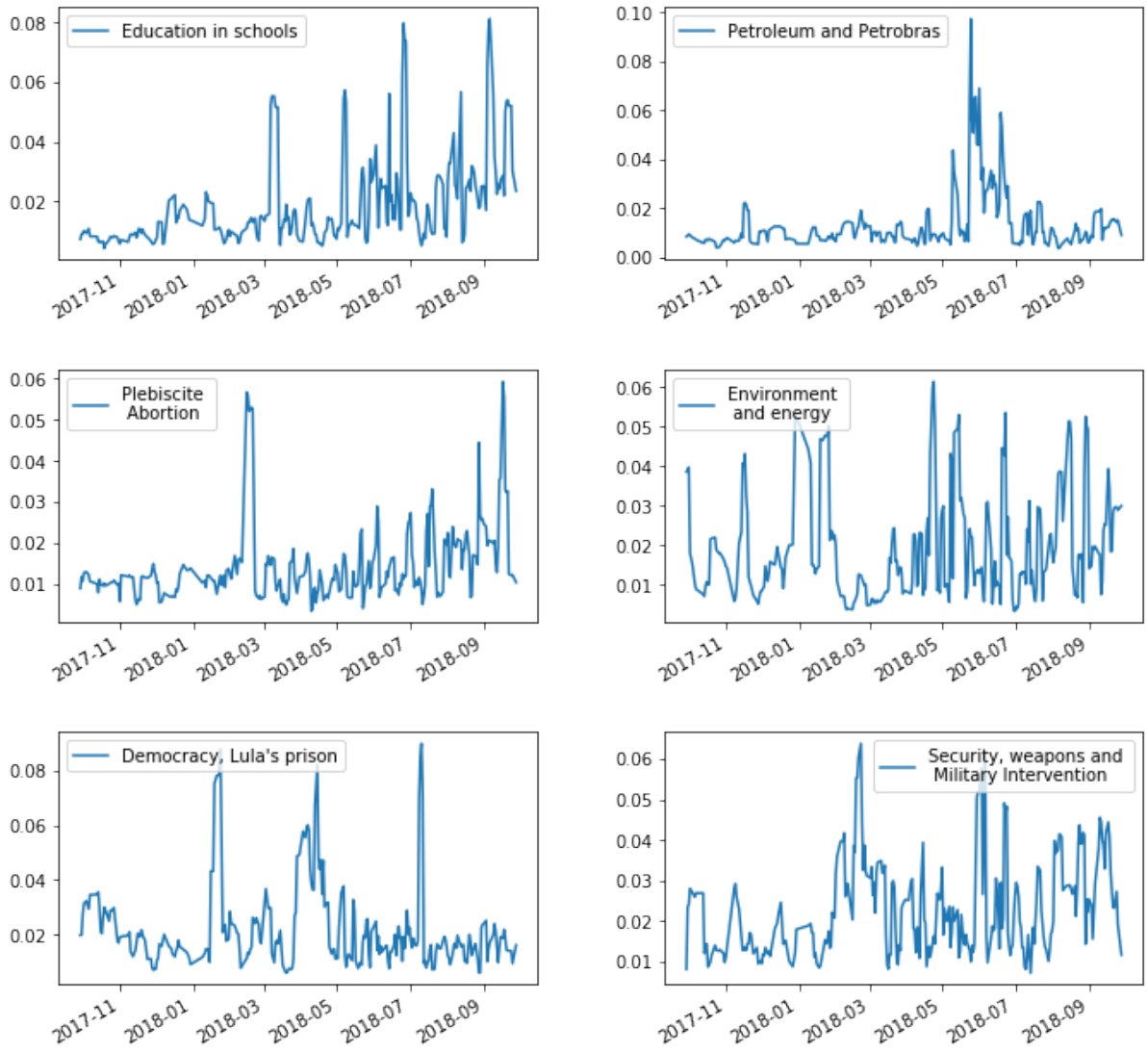


Figure 5.17: Evolution of some relevant topics discussed during the Brazilian political campaign, using the probabilities of topics in each speech.

2

5.3 Result of Candidates Comparison

We present the results obtained from the metrics for candidate comparison. The similarity between candidates, political positions, political content relationship, and topics evolution by a candidate are explained.

5.3.1 Similarity Between Candidates

To calculate the similarity between candidates, as explained in the Algorithm 4, it is necessary to define the number of sub-samples of pairs of discourses to be compared. This number depends on the number of speeches per candidate that is available, so in this case a sub-sample of 60 pairs of speeches was used. Table 5.2 shows the maximum similarity of each candidate for TF-IDF and Doc2Vec. Both models found the same candidate most similar for one each, except for candidates Alvaro Dias and João Amoêdo.

Table 5.2: Similarity of the candidate, shows the first similar candidate with their respective short name: GB, Marina Silva: MS, Ciro Gomes: CG, Geraldo Alckmin: GA, Henrique Meirelles: HM, Alvaro Dias: AD, Jair Bolsonaro: JB, and João Amoêdo: JA.

Candidate name	**Candidate Most Similar**	
	Doc2Vec	TF-IDF
Guilherme Boulos	CG: 0.520	CG: 0.272
Marina Silva	GB: 0.507	GB: 0.299
Ciro Gomes	GA: 0.528	GA: 0.274
Geraldo Alckmin	CG: 0.528	CG: 0.274
Henrique Meirelles	JA: 0.496	JA: 0.280
Álvaro Dias	CG: 0.514	JA: 0.263
Jair Bolsonaro	CG: 0.517	CG: 0.247
João Amoêdo	GB: 0.511	HM: 0.280

5.3.2 Intuiting Political Positions

The last four presidents in Brazil were from moderate parties. Due to the economic crisis that Brazil is facing in the last four years, part of the population (the conservative ones) has associated the crisis with the moderate-left agenda, which includes discussing social issues, abortion, gender discussions, *etc.* On the other hand, millions of citizens benefited with social programs and support moderate-left parties, and think that the right parties

are not a good option. In summary, in the 2018 Brazilian election, there was a "popular polarization" (or mass polarization) that refers to the polarization in the electorate and general public.

The only candidate who declared himself to be from the right - or far-right - was Jair Bolsonaro, while the other candidates were labeled as moderate-right, center, moderate-left, and left. POSTURE allows for a visualization tool for the speeches represented in a vector space aiming at checking the political positions of the candidates compared to each other w.r.t. their discourse. To reduce the original dimension of the speeches' vector so that it is possible to plot them in a human-comprehensible space, POSTURE relies on the PCA algorithm [47] to reduce the dimensionality to three dimensions.

The Figure 5.18 and Figure 5.19 exhibit the vector space of the candidate's discourse where each point represents one speech from a single candidate and each color represent the speeches of the same candidate. In Figure 5.18 the Tf-IDF vectorization presents a cone shape, where all candidates are intercepted in the vertex area, but, outside of the interception, there are three groups without any relationship: as Ciro Gomes, Guilherme Boulos, and the rest of candidates. In comparison to the Doc2vec vectorization, in the Figure 5.19 we can note some points of intersection, but most of the candidates are separated from the others, except the candidate Geraldo Alckmin (moderate-right), who is almost entirely in the intersection. The closest proximity is observed between Ciro Gomes (moderate-left) with Jair Bolsonaro (Right), and last João Amoêdo with Henrique Meirelles belonging to both moderate-right.

In Brazilian political in total there are five political positions composed for *i) Left(L):* Guilherme Boulos. *ii) Moderate-left(ML):* Marina Silva and Ciro Gomes. *iii) Center(C). iv) Moderate-Right(MR):* Geraldo Alckmin, Henrique Meirelles, and Alvaro Dias. (v) Right(R): Jair Bolsonaro and João Amoêdo. In Table 5.3 show the a priori political position of each candidate, the most similar position for each candidate, and the result of intuition political position after to apply the Equation 3.2.

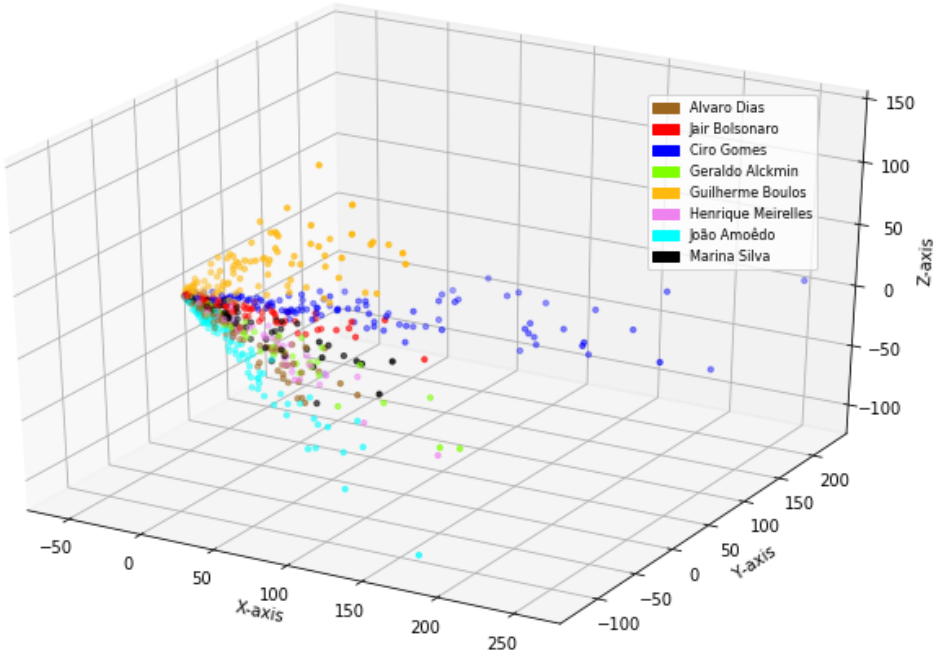


Figure 5.18: Vector space representation in three dimensions of the speeches of all candidates by TF-IDF.

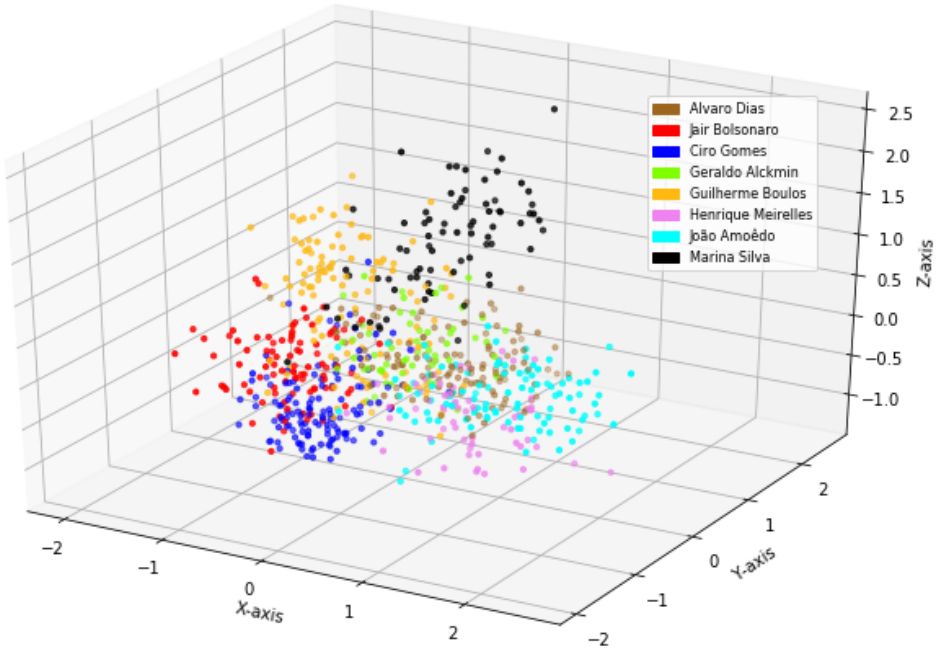


Figure 5.19: Vector space representation in three dimensions of the speeches of all candidates by Doc2Vec.

Table 5.3: Intuition Political Positions, shows the position of the first similar candidate with their respective position and short name: Guilherme Boulos GB, Marina Silva MS, Ciro Gomes CG, Geraldo Alckmin GA, Henrique Meirelles HM, Alvaro Dias AD, Jair Bolsonaro JB, and João Amoêdo JA.

Candidate Name	**Political Positions**			**Final Position**
	Position	Doc2vec	TF-IDF	
Guilherme Boulos	L	CG: ML	CG: ML	ML
Marina Silva	ML	GB: L	GB: L	ML
Ciro Gomes	ML	GA: MR	GA: MR	C
Geraldo Alckmin	MR	CG: ML	CG: ML	C
Henrique Meirelles	MR	JA: R	JA: R	MR
Álvaro Dias	MR	CG: ML	JA: R	C
Jair Bolsonaro	R	CG: ML	CG: ML	C
João Amoêdo	R	GB: L	HM: MR	C

5.3.3 Analyzing Speeches with Strong Similarity Relationships

For calculate *the similar content*, it necessary to determinate the similarity between speeches, in general, depends on the degree of the similarity search. In the speeches comparison of the different candidates tend to have a considerable similarity, hence it was defined as 0.55 as minimum similarity for the Doc2Vec model and 0.35 for TF-IDF.

The Figure 5.20 and The Figure 5.21, in general, reveal that most similar discourses are carried out in short periods of time, especially in TF-IDF pairs, with some exceptions with Doc2Vec pairs. This could happen because of an important event in Brazilian politics when most of the candidates are going to speak of it. In the pairs found by TF-IDF, we can first observe that the *maximum similarity* is low (0.43), due to the very different frequency of words used in the discourses. The most similar pairs were found in the same month or a month later. We also note that Doc2Vec pairs manage to capture some similar pairs with long-term discourses. The maximum similarity value found by Doc2Vec is 0.73. These discourses in distant months have captured the semantics rather than only similar lexical content. This semantic similarity is reflected, for example, in discourse A about the privatization of Petrobras, and months later in a discourse B that speaks about the privatization of an electric power company, Eletrobras.

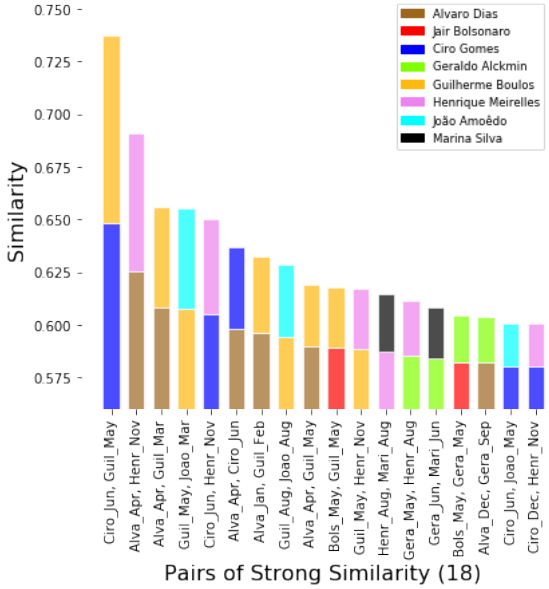


Figure 5.20: Pairs of Strong Similarity speeches ordered in descending similarity discovered by Doc2Vec.

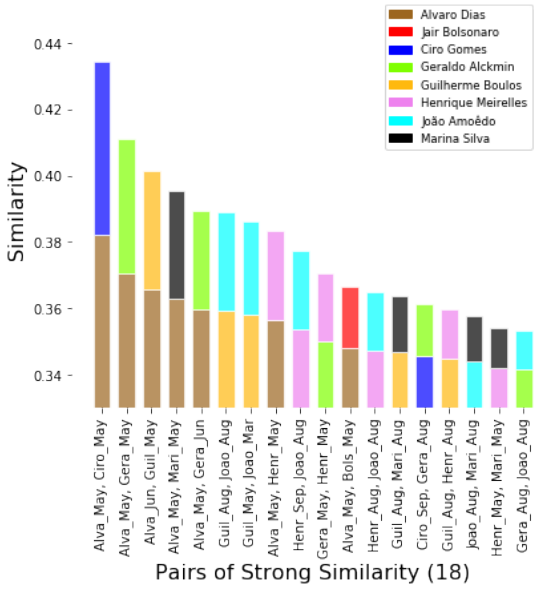


Figure 5.21: Pairs of Strong Similarity speeches ordered in descending similarity discovered by TF-IDF

After *calculating the semantic order* in the Figure 5.22, it is observed the similarity of the pairs calculated by TF-IDF together with TSA, where the TSA similarity in most pairs reaches a higher value than TF-IDF. As previously mentioned, TSA manages to capture the semantic similarity considering the thematic order. When there is a high TF-IDF similarity, the document share a lot of words in common, that is probability that both discourses also have the same order. These speeches with a similar order are confirmed, through the use of the TSA algorithm.

In the Table 5.4 a fragment of the pair of discourses found by TF-IDF is shown, this pair of discourse has a higher similarity for TSA, that is, a semantic similarity with a similar thematic order. This is explained because these speeches were made in a debate called *'XXI Marcha a Brasilia em defesa dos municipios'*[3]. In this debate, the journalists asked the same questions in the same order with a limited response time to both are candidates.

On the other hand, in the Figure 5.23, we show the pairs calculated by Doc2Vec together with TSA, we know that both of them aim at capturing a semantic similarity. For example, the first pair has very similarity values for Doc2Vec and TSA, this indicates the existence of semantic similarity, and similar thematic order. In table 5.5 is showed a pair of interviews, where both candidates talk about the truck drivers' strike with a very

[3] `http://www.marcha.cnm.org.br`

similar order (Pedro Parente -> fuel rise -> truck drivers' strike -> economy). We also observe that not necessarily all the semantically similar speeches have a thematic order Similarity.

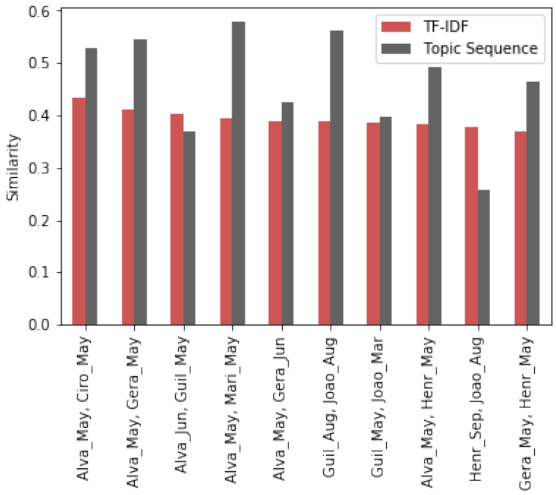


Figure 5.22: Relation between TF-IDF and Topic Sequence

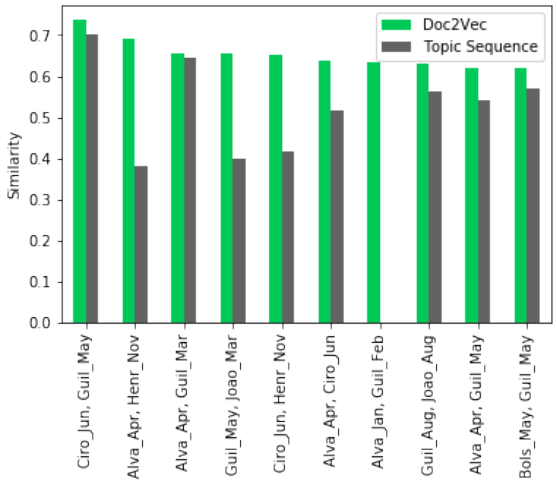


Figure 5.23: Relation between Doc2vec and Topic Sequence

Table 5.4: Segments of the pair of speeches found by TF-IDF with high similarity of TSA

Part of Alvaro' speech on May 22	Part of Marina' speech on May 22
...bolsa família que são muito importantes e outras medidas mais que a gente possa trabalhar com a inclusão produtiva como acontece no chile com os agentes de desenvolvimento social tendo nesses centros um ponto de referência para que as políticas sejam transversais do atendimento da oferta das oportunidades disponíveis pelos municípios a assistente social são responsáveis pela oferta está certo risco só concluindo nosso compromisso é de mesmo em função da 'mesmo com escassez não podemos abrir mão de determinadas políticas obviamente que vamos fazer isso também fechando o dreno da corrupção o senhor barusco devolveu num piscar de olhos mais de 100 milhões imagine a diferença que se faz dentro de um município pobre de até 100 40 30 mil municípios o senhor eike batista levou mais de 9 bilhões o bolsa-empresário custa 5 por cento do pib enquanto o bolsa família apenas 0,5 por cento combater a corrupção também faz parte de políticas sociais que atendam à sociedade mas de forma integrada e transmissão vamos dar seqüência para quarta pergunta a pré-candidata marina os municípios são responsáveis pela oferta da educação infantil que passou a ter a pré escola como obrigatória a partir de 2016 e tem como meta atender ao menos 50% da faixa etária de creches até 2024 para atender essa imposição são necessárias cerca de mais 500 mil matrículas na pré-escola e 2,2 milhões de matrículas na creche o custo especialmente da creche...	...no nordeste brasileiro 66% dos nordestinos vivem com menos de um salário mínimo por mês 15 milhões de brasileiros vivem com até 136 reais por mês como é possível sobreviver dessa maneira e como é possível admitir esse contraste gritante injusto desonesto perverso e cruel com a gente desse país deus foi generoso conosco não podemos admitir esta incompetência ea consagração da corrupção esse sistema é corrupto e sistema fracassou ele tem que ser substituído o brasil hoje o brasil hoje o brasil hoje não está dividido entre esquerda e direita o bar brasil hoje está dividido entre os honestos e os ladrões da república que assaltar os cofres neste país essa é a divisão nós temos que separar o joio do trigo prefeitos municipais não admitam que nos joga em todos o mesmo lá mas ao de corrupção que felicitou este país a decência a competência a inteligência no brasil nós podemos abrir caminhos amplas avenidas na direção do nosso futuro nós podemos mudar esse país e haveremos de mudá lo vamos viver a fé perdida nas estradas da execução vamos ressuscitar as esperanças sepultadas sob os escombros da incompetência e da corrupção instituições públicas que foram destruídas pelos incompetentes e corruptos vamos caminhar por esses caminhos difíceis de caminhar mas na direção do nosso futuro buscando coesão e rumo esse país está desarrumado desarrumaram administração pública brasileira a nossa missão é arrumá lá vamos arrumar o brasil vamos buscar coesão e unidade e vamos caminhar na direção do nosso futuro para a construção da nação que todos nós merecemos a grande nação dos nossos filhos e dos netos vamos juntos mudar esse país prefeitos brasileiros...

Table 5.5: Pair of speeches found by Doc2Vec with high similarity of TSA

Part of Ciro' speech on Jun 1	Part of Guilherme' speech on May 14
a demissão do seu pedro parente da direção da petrobras é uma de duas providências que tinham que ser tomadas a primeira deve ser ele mesmo uma pessoa tem coragem no meio de uma atriz extraordinária mente grave como a que nós vivemos recentemente pela greve dos caminhoneiros a greve dos petroleiros eo desabastecimento que mexeu com a vida de todo mundo o cidadão da ter o desplante o despudor de aumentar a gasolina em quase 1 por cento apenas um dia no meio da crise essa falta de respeito da tutela política do psdb é a política que quer valorizar o financista que quer valorizar a especulação financeira em detrimento seja de quanto esse foco especialmente o interesse popular de interesse nacional brasileiro mas não basta demitiu seu pedro parente é preciso exigir que a política de preços que ele impôs seja trocar e ela não pode ser trocada por nada demagogia apenas o seguinte hoje eles estão transferindo o preço do barril de petróleo da especulação estrangeira para dentro do brasil quando o custo da petrobras é muitas vezes menor do que o custo do petróleo lá fora e esticada na prática o seguinte vão sobrar 70 bilhões nesse período de apropriação nós vamos deixar que os acionistas minoritários são setor financeiro estrangeiro e nacional a própria 70 milhões ou vamos garantir a saúde da petrobras repassar esse excedente para o interesse público é disso que se trata.	a crise dos combustíveis que a gente está vivendo hoje é resultado da política desastrosa do pedro parente e do temer na petrobras para dar lucro para os acionistas da empresa lá fora eles liberaram os preços e tem utilizado inclusive uma capacidade inferior do que as refinarias nacionais têm aí manda o petróleo para refinar lá fora e aí o custo em dólar e é muito mais caro principalmente quando o dólar aumenta como está acontecendo agora mesmo assim a gasolina sai das refinarias hoje a dois reais e três centavos o lucro depois disso vai para a distribuidora e para os postos de gasolina que estão praticando uma verdadeira agiotagem sem nenhuma fiscalização e sem nenhuma regulação do governo além disso é preciso dizer que a política tributária do estado brasileiro que cobra mais imposto do consumo e da produção e menos imposto da renda da propriedade também aumenta os presos a greve dos caminhoneiros é resultado disso o desabastecimento que a gente já está vendo nos postos e nas cidades é resultado disso assim como o aumento abusivo no preço do botijão de gás já tinha sido resultado dessa mesma política pedro parenti se tiver vergonha na cara tem que pedir pra sair agora imediatamente e se não tiver tem que ser tirado de lá aliás o temer o seu chefe já deveria ter saído há muito tempo esses aumentos abusivos tem que ser revogados agora a empresa pública tem que servir para o povo brasileiro e não para dar lucro para meia dúzia de acionista lá fora esse tem que ser o objetivo da petrobras 45 o total.

5.3.4 Comparison of the Candidates Evolution of a Specific Topic

Similar to Subsection 5.2.3, this metric has only one parameter to be defined, also the 10 previous data were used. The Figure 5.24 shows the comparison of the most relevant topics of Brazilian politics for the four selected candidates. For example, considering the *"Education in schools"* topic, the candidate who most discussed it was Alvaro Dias, followed by Marina Silva. In the *"Petroleum and Petrobras"* topic, the image shows that during the month of June all the candidates spoke about this, but Ciro Gomes was the candidate that has made more emphasis on that topic months later. In the *"Security, use of weapons and military intervention"* topic, we can observe that this is a constant topic spoken in some period by Jair Bolsonaro. Finally, we can perceive that the only candidate that talks about *"Environment and energy"* is Marina Silva.

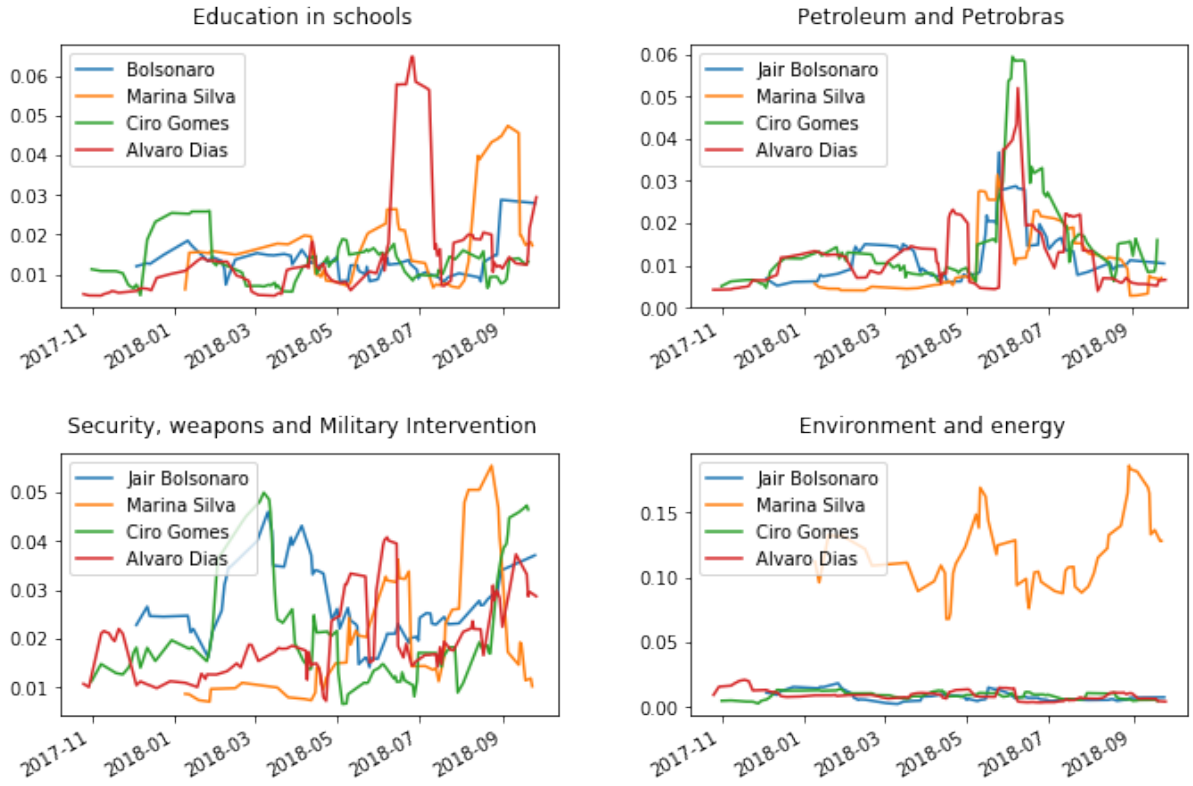


Figure 5.24: Comparison of evolution between candidates of relevant topics discussed during the Brazilian political campaign, using the probabilities of topics in each speech.

Chapter 6

Conclusions and Future Works

This Chapter concludes this dissertation by briefly reiterating the contributions and the results obtained in the Section 6.1. Finally, Section 6.2 presents some ways to improve and extend this research.

6.1 Final Remarks

In this dissertation, we proposed the POSTURE framework for the analysis of discourse and topic level of political discourses, based on natural language processing and unsupervised machine learning techniques. Therefore, POSTURE becomes a unique framework that performs analysis both at the level of discourse and thematic level.

The data was originally collected in the videos form that were pre-processed and converted into plain text format; these data are speeches from 2018 Brazilian presidential election, which were made available for further study.

POSTURE helps to a superficial understanding of the behavior of the candidates with the exploration of data. Besides some metrics were proposed for the analysis of the behavior candidate and the calculation of the difference between candidates using different aspects.

For calculating the proposed metrics in this dissertation, two way to calculating similarity were addressed: similarity based on vector text representation, and the use of TSA similarity that is a not-based in vector representation. For validating these models, the accuracy evaluation metric was used using the same similarity triplet test dataset.

The use of the documents embedding obtained better results in precision in the calculation of the semantic similarity and execution time. Contrary, the time execution of

the TSA takes more time. For that reason, TSA was used as an additional component, since it manages to find a relationship between thematic order and the semantic similarity between documents.

For the similarity based in vector representation, we used lexical and semantic representations. In terms of execution time and computational cost, it is convenient to use TF-IDF. But this representation does not manage to find semantic relations between documents, this can limit in some metrics proposals, as the coherence of the candidates and constant speeches. Conversely, TF-IDF for calculating of the similarity of candidates obtained results more coherent than doc2vec. The TF-IDF and Doc2Vec models were compared, concluding that the use of both techniques facilitates a better understanding of political behavior.

An unfavorable point of this framework is that it still does not differentiate the positioning of candidates in a specific topic, only detect that both candidates speak of the same topic, without knowing if it is in favor of it or not.

6.2 Future Works

There are ways to improve the performance of POSTURE, in the short term, train the Doc2vec model with a higher amount of data, and find a better LDA topic model. In the long term, we could try to improve the results using other types of more recent embedding, which make a better representation of semantic content.

The implementation of this framework, moreover allowing the use of different text vectorial representations, is flexible in the similarity measure. Although in the present dissertation it uses the cosine distance, some studies try to improve or correct some disadvantages found in this measure like work [17], that can help improve the results of POSTURE.

Knowing the position of a candidate in a specific topic is essential in political analysis. A classifier could be trained for the sentiment analysis that allows to capture and differentiate the position of the candidate in a particular topic.

In this dissertation, although the computational models used are valid, it remains to validate the usability of POSTURE, that is, performing a qualitative analysis by the voters.

References

[1] Azarbonyad, H., Dehghani, M., Beelen, K., Arkut, A., Marx, M., Kamps, J. Words are malleable: Computing semantic shifts in political and media discourse. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (2017), ACM, p. 1509–1518.

[2] Bakliwal, A., Foster, J., van der Puil, J., O'Brien, R., Tounsi, L., Hughes, M. Sentiment analysis of political tweets: Towards an accurate classifier. Association for Computational Linguistics.

[3] Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C. A neural probabilistic language model. *Journal of machine learning research 3*, Feb (2003), 1137–1155.

[4] BENITES, A. Candidatos de centro-direita já negociam pacto de não-agressão e união futura. *El Pais* (2018).

[5] Blei, D. M., Ng, A. Y., Jordan, M. I. Latent dirichlet allocation. In *Advances in neural information processing systems* (2002), p. 601–608.

[6] Blei, D. M., Ng, A. Y., Jordan, M. I. Latent dirichlet allocation. *Journal of machine Learning research 3*, Jan (2003), 993–1022.

[7] Brill, E., Mooney, R. J. An overview of empirical natural language processing. *AI magazine 18*, 4 (1997), 13–13.

[8] Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., Blei, D. M. Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems* (2009), p. 288–296.

[9] Christhie, W., Reis, J. C., Moro, F. B. M. M., Almeida, V. Detecção de posicionamento em tweets sobre política no contexto brasileiro. In *7º Brazilian Workshop on Social Network Analysis and Mining (BraSNAM 2018)* (2018), vol. 7, SBC.

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

[11] Gautrais, C., Cellier, P., Quiniou, R., Termier, A. Topic signatures in political campaign speeches. In *EMNLP 2017-Conference on Empirical Methods in Natural Language Processing* (2017).

[12] Gerrish, S. M., Blei, D. M. Predicting legislative roll calls from text. In *Proceedings of the 28th International Conference on International Conference on Machine Learning* (2011), ICML'11, Omnipress, p. 489–496.

[13] Greene, D., Cross, J. P. Exploring the political agenda of the european parliament using a dynamic topic modeling approach. *Political Analysis 25*, 1 (2017), 77–94.

[14] Guthrie, D., Allison, B., Liu, W., Guthrie, L., Wilks, Y. A closer look at skip-gram modelling. In *Proceedings of the 5th international Conference on Language Resources and Evaluation (LREC-2006)* (2006), p. 1–4.

[15] Hajeer, I. Comparison on the effectiveness of different statistical similarity measures. *International Journal of Computer Applications 53*, 8 (2012).

[16] Harris, Z. S. Distributional structure. *Word 10*, 2-3 (1954), 146–162.

[17] Heidarian, A., Dinneen, M. J. A hybrid geometric approach for measuring similarity level among documents and document clustering. In *2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService)* (2016), IEEE, p. 142–151.

[18] Hoffman, M., Bach, F. R., Blei, D. M. Online learning for latent dirichlet allocation. 856–864.

[19] Iyyer, M., Enns, P., Boyd-Graber, J., Resnik, P. Political ideology detection using recursive neural networks. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2014), vol. 1, p. 1113–1122.

[20] Jaccard, P. The distribution of the flora in the alpine zone. 1. *New phytologist 11*, 2 (1912), 37–50.

[21] Kessler, J. S. Scattertext: a browser-based tool for visualizing how corpora differ.

[22] Kietzmann, J. H., Hermkens, K., McCarthy, I. P., Silvestre, B. S. Social media? get serious! understanding the functional building blocks of social media. *Business horizons 54*, 3 (2011), 241–251.

[23] Kornbrot, D. P earson product moment correlation. *Encyclopedia of statistics in behavioral science* (2005).

[24] Kusner, M., Sun, Y., Kolkin, N., Weinberger, K. From word embeddings to document distances. In *International Conference on Machine Learning* (2015), p. 957–966.

[25] Lau, J. H., Baldwin, T. An empirical evaluation of doc2vec with practical insights into document embedding generation. *CoRR abs/1607.05368* (2016).

[26] Le, Q., Mikolov, T. Distributed representations of sentences and documents. In *International conference on machine learning* (2014), p. 1188–1196.

[27] Lee, D. D., Seung, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature 401*, 6755 (1999), 788.

[28] Loper, E. Nltk: Building a pedagogical toolkit in python. *PyCon DC 2004* (2004).

[29] Maheshwari, G., Trivedi, P., Sahijwani, H., Jha, K., Dasgupta, S., Lehmann, J. Simdoc: Topic sequence alignment based document similarity framework. In *Proceedings of the Knowledge Capture Conference* (2017), ACM, p. 16.

[30] Manning, C. D., Manning, C. D., Schütze, H. *Foundations of statistical natural language processing*. MIT press, 1999.

[31] Maynard, D., Funk, A. Automatic detection of political opinions in tweets. In *Extended Semantic Web Conference* (2011), Springer, p. 88–99.

[32] McClurg, S. D. Social networks and political participation: The role of social interaction in explaining political participation. *Political research quarterly 56*, 4 (2003), 449–464.

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

[34] Mikolov, T., Yih, W.-t., Zweig, G. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2013), p. 746–751.

[35] Minsky, M., Papert, S. An introduction to computational geometry. *Cambridge tiass., HIT* (1969).

[36] Nelson, M. L., Bollen, J., Calhoun, J. R., Mackey, C. E. User evaluation of the nasa technical report server recommendation service. In *Proceedings of the 6th annual ACM international workshop on Web information and data management* (2004), ACM, p. 144–151.

[37] O'callaghan, D., Greene, D., Carthy, J., Cunningham, P. An analysis of the coherence of descriptors in topic modeling. *Expert Systems with Applications 42*, 13 (2015), 5645–5657.

[38] Pennington, J., Socher, R., Manning, C. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (2014), p. 1532–1543.

[39] Řehůřek, R., Sojka, P. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks* (Valletta, Malta, maio de 2010), ELRA, p. 45–50. `http://is.muni.cz/publication/884893/en`.

[40] Röder, M., Both, A., Hinneburg, A. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining* (2015), ACM, p. 399–408.

[41] Rumelhart, D. E., Hinton, G. E., Williams, R. J. Learning representations by back-propagating errors. *nature 323*, 6088 (1986), 533.

[42] Salton, G., Buckley, C. Term-weighting approaches in automatic text retrieval. *Information processing & management 24*, 5 (1988), 513–523.

[43] Salton, G., Wong, A., Yang, C.-S. A vector space model for automatic indexing. *Communications of the ACM 18*, 11 (1975), 613–620. The paper where vector space model for IR was introduced.

[44] Smith, T. F., Waterman, M. S. Identification of common molecular subsequences. In *Journal of Molecular Biology* (1981), vol. 147(1), p. 195–197.

[45] Thada, V., Jaglan, V. Comparison of jaccard, dice, cosine similarity coefficient to find best fitness value for web retrieved documents using genetic algorithm. *International Journal of Innovations in Engineering and Technology 2*, 4 (2013), 202–205.

[46] Wallach, H. M. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning* (2006), ACM, p. 977–984.

[47] Wold, S., Esbensen, K., Geladi, P. Principal component analysis. *Chemometrics and intelligent laboratory systems 2*, 1-3 (1987), 37–52.

[48] Zhang, Y., Jin, R., Zhou, Z.-H. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics 1*, 1-4 (2010), 43–52.

APPENDIX A – Similarity Matrix of Candidates

In this section, matrices of similarity between candidates based on their speeches are presented, both for TF-IDF and Doc2Vec.

Table A.1: Candidates Similarity Matrix using TF-IDF Vectors

Candidates	Guilherme Boulos	Marina Silva	Ciro Gomes	Geraldo Alckmin	Henrique Meirelles	Álvaro Dias	Jair Bolsonaro	João Amoêdo
Guilherme Boulos	1	0,265	**0,273**	0,257	0,263	0,254	0,224	0,277
Marina Silva	**0,265**	1	0,247	0,248	0,249	0,249	0,22	0,262
Ciro Gomes	0,273	0,247	1	**0,274**	0,268	0,239	0,247	0,265
Geraldo Alckmin	0,257	0,248	**0,274**	1	0,26	0,249	0,224	0,26
Henrique Meirelles	0,263	0,249	0,268	0,26	1	0,24	0,215	**0,28**
Álvaro Dias	0,254	0,249	0,239	0,249	0,24	1	0,206	**0,263**
Jair Bolsonaro	0,224	0,22	**0,247**	0,224	0,215	0,206	1	0,237
João Amoêdo	0,277	0,262	0,265	0,26	**0,28**	0,263	0,237	1

Table A.2: Candidates Similarity Matrix using Doc2Vec Vectors

Candidates	Guilherme Boulos	Marina Silva	Ciro Gomes	Geraldo Alckmin	Henrique Meirelles	Álvaro Dias	Jair Bolsonaro	João Amoêdo
Guilherme Boulos	1	0,512	**0,52**	0,501	0,48	0,502	0,488	0,519
Marina Silva	**0,512**	1	0,507	0,496	0,479	0,486	0,487	0,494
Ciro Gomes	0,52	0,512	1	**0,528**	0,49	0,514	0,517	0,511
Geraldo Alckmin	0,501	0,496	**0,528**	1	0,485	0,492	0,489	0,501
Henrique Meirelles	0,48	0,479	0,49	0,485	1	0,482	0,47	**0,496**
Álvaro Dias	0,502	0,486	**0,514**	0,492	0,482	1	0,493	0,498
Jair Bolsonaro	0,488	0,487	**0,517**	0,489	0,47	0,493	1	0,489
João Amoêdo	**0,519**	0,494	0,511	0,501	0,496	0,498	0,489	1

APPENDIX B – Topics Extracted using LDA

In this section the 45 topics found by LDA are presented, showing their top 10 terms.

Table B.1: Political topics extracted using LDA (Topic 1 - Topic 18)

Topic 1	**Topic 2**	**Topic 3**	**Topic 4**	**Topic 5**	**Topic 6**	**Topic 7**	**Topic 8**	**Topic 9**
congresso	educação	empresa	congresso	dinheiro	temer	social	dilma	direito
reforma	ensino	empresas	real	cidadão	ministério	movimento	nacional	democracia
programa	escola	petrobras	fernando	liberdade	lei	casa	vida	judiciário
turno	qualidade	privatização	futebol	melhorar	dilma	moradia	globo	justiça
psdb	professor	privatizar	melhor	justamente	presidência	ocupação	lei	preso
eleições	escolas	dinheiro	qualidade	gestão	psdb	vida	projeto	prisão
eleitoral	fundamental	caixa	oposição	entendo	crime	teto	campo	juiz
pmdb	médio	energia	espaço	gostaria	justiça	aluguel	história	defesa
michel_temer	professores	água	psdb	educação	pesquisas	famílias	força	esquerda
presidência	básico	melhor	receber	acaba	discurso	milhões	cultura	prova
Topic 10	**Topic 11**	**Topic 12**	**Topic 13**	**Topic 14**	**Topic 15**	**Topic 16**	**Topic 17**	**Topic 18**
história	segurança	economia	vida	presidência	dívida	reforma	congresso	casa
vida	polícia	inflação	pai	rádio	bilhões	setor	reforma	maneira
futuro	violência	banco_central	casa	nacional	fiscal	tributária	investimento	filho
melhor	crime	crescimento	mãe	dias	milhões	segurança	nacional	direito
precisamos	arma	emprego	igreja	joão	economia	países	enfrentar	china
esperança	federal	condições	livro	semana	mercado	investimento	privilégios	militar
juntos	drogas	crescer	criança	nordeste	taxa	investimentos	michel_temer	entender
agradecer	intervenção	fazenda	filho	jornal	crédito	empresas	imposto	federal
coração	armas	previdência	escola	cidade	bancos	produtividade	juros	acabar
oportunidade	policiais	fundamental	época	estados	econômica	privado	medo	quiser

Table B.2: Political topics extracted using LDA (Topic 19 - Topic 36)

Topic 19	**Topic 20**	**Topic 21**	**Topic 22**	**Topic 23**	**Topic 24**	**Topic 25**	**Topic 26**	**Topic 27**
projeto	produção	reforma	possibilidade	mercado	mulheres	preço	situação	esquerda
base	agronegócio	previdência	presidência	real	mulher	petróleo	rede	projeto
papel	agricultura	milhões	responsabi..	acontece	aborto	petrobras	corrupção	crise
sair	rural	dinheiro	discurso	professor	lei	preços	justiça	sociais
vista	campo	bilhões	volta	começa	homens	gasolina	dilma	movimento
presença	setor	imposto	popular	linha	direitos	venezuela	base	social
simples	crédito	pagar	solução	história	programa	caminhoneiros	ganhar	movimentos
resposta	segurança	trabalhador	opinião	negócio	democracia	greve	lava_jato	golpe
federal	agrícola	paga	discutir	casa	idade	combustível	governar	luta
questões	propriedade	idade	união	tomar	plebiscito	crise	democracia	construir
Topic 28	**Topic 29**	**Topic 30**	**Topic 31**	**Topic 32**	**Topic 33**	**Topic 34**	**Topic 35**	**Topic 36**
paraná	mudar	joão	ambiente	nacional	bahia	milhões	federal	municípios
dinheiro	dinheiro	idéias	energia	saúde	volta	ceará	psdb	estados
dias	mudança	melhor	crise	precisamos	dinheiro	prefeito	emprego	federal
autoridade	sair	vida	milhões	terra	programa	vida	crescer	saúde
recursos	televisão	liberdade	políticas	especialmente	negócio	golpe	renda	recursos
corrupção	começa	deveria	problemas	ciência	história	pdt	aliás	união
privilégios	difícil	difícil	desenvolv..to	renda	daqui	conta	fizemos	educação
salário	deixar	montar	social	passado	chega	cabeça	reformas	prefeito
economia	acredita	econômica	sustentável	agenda	filho	classe	investimento	federativo
mandato	passar	sentido	difícil	inteligência	rádio	experiência	inteiro	prefeitos

Table B.3: Political topics extracted using LDA (Topic 37 - Topic 45)

Topic 37	**Topic 38**	**Topic 39**	**Topic 40**	**Topic 41**	**Topic 42**	**Topic 43**	**Topic 44**	**Topic 45**
cabeça	refundação	programa	militar	corrupção	dólar	ceará	banco	saúde
fácil	reformas	queremos	economia	senado	china	direito	época	família
carro	bilhões	saúde	daí	podemos	modelo	entender	história	emprego
melhor	corrupção	questões	botar	câmara	nacional	fernando	vida	bolsa
domingo	podemos	universidade	buscar	privilegiado	economia	práticas	tecnologia	programa
cidade	mudança	semana	exército	federal	estrangeiro	últimos	milhões	vida
jogo	renda	enfrentar	ministério	projeto	comprar	proposta	direito	segurança
ajuda	reforma	modelo	passado	foro	fernando	professor	ganhar	atendimento
sul	futuro	políticas	mulher	deputados	indústria	lei	fiz	milhões
corpo	dívida	café	câmara	congresso	industrial	simplesmente	criar	sus