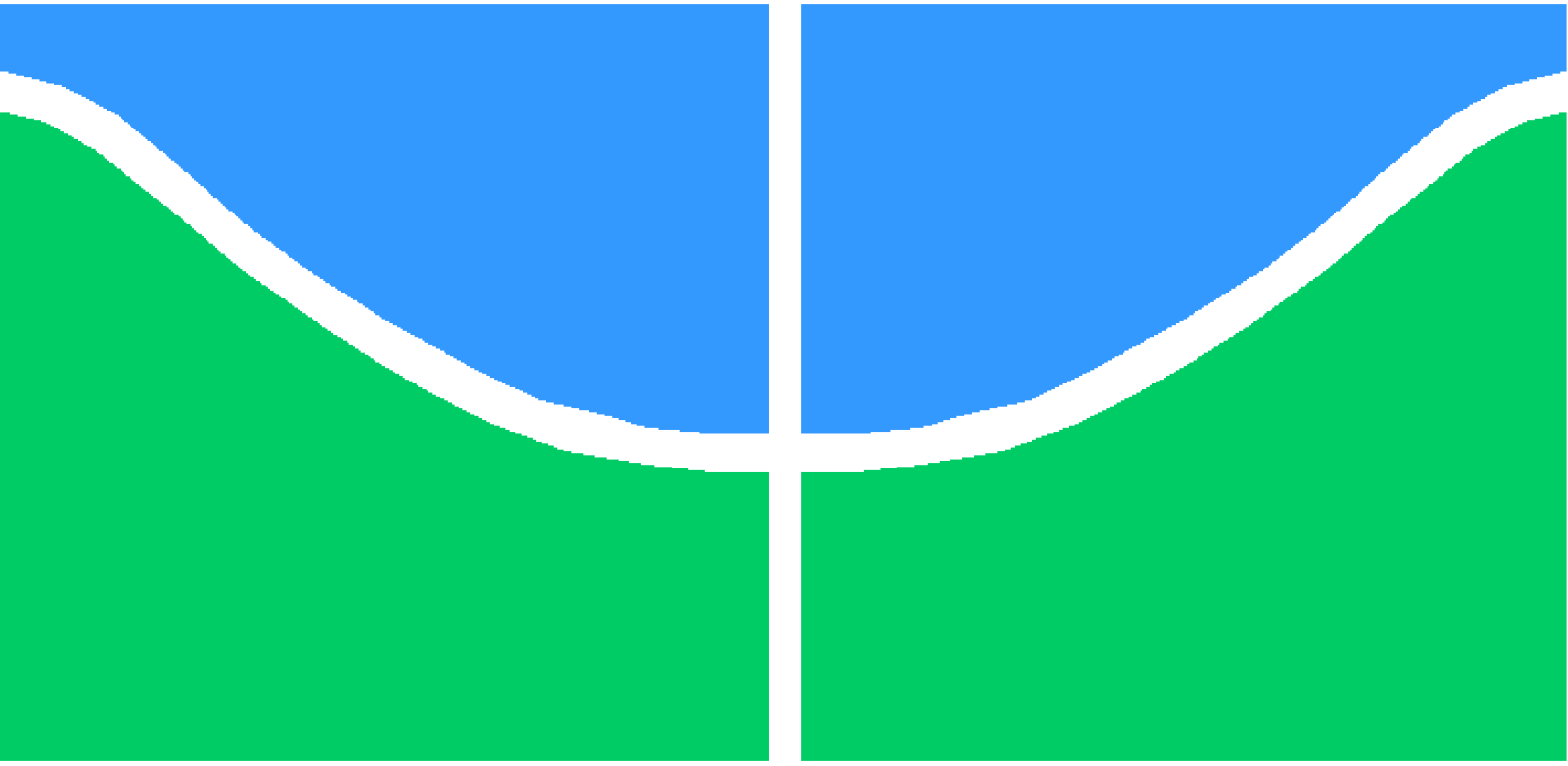


Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Engenharia de Software

Tenho Dito: uma aplicação para análise de discursos parlamentares utilizando técnicas de processamento de linguagem natural

Autor: Matheus Souza Fernandes

Orientador: Prof. Dr. Fábio Macedo Mendes

Brasília, DF

2017

Matheus Souza Fernandes

Tenho Dito: uma aplicação para análise de discursos parlamentares utilizando técnicas de processamento de linguagem natural

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Prof. Dr. Fábio Macedo Mendes

Brasília, DF

2017

Matheus Souza Fernandes

Tenho Dito: uma aplicação para análise de discursos parlamentares utilizando técnicas de processamento de linguagem natural/ Matheus Souza Fernandes. – Brasília, DF, 2017-

102 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Fábio Macedo Mendes

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2017.

1. processamento de linguagem natural. 2. aprendizado de máquina. I. Prof. Dr. Fábio Macedo Mendes. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Tenho Dito: uma aplicação para análise de discursos parlamentares utilizando técnicas de processamento de linguagem natural

CDU 02:141:005.6

Matheus Souza Fernandes

Tenho Dito: uma aplicação para análise de discursos parlamentares utilizando técnicas de processamento de linguagem natural

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Trabalho aprovado. Brasília, DF, 01 de junho de 2013:

Prof. Dr. Fábio Macedo Mendes
Orientador

Prof. Dr. Paulo Roberto Miranda Meirelles
Convidado 1

Prof. Dr. Teófilo de Campos
Convidado 2

Brasília, DF
2017

Agradecimentos

Primeiramente, agradeço a Deus por ter me dado força e saúde para chegar até aqui. Agradeço também aos meus pais e irmãos, pelo apoio e por não medirem esforços para que eu sempre tivesse boas condições de estudo.

Agradeço a Paola, minha noiva e companheira, que sempre me apoiou e me deu forças para continuar lutando, que nos momentos mais difíceis estava ao meu lado, com quem compatilho tudo e posso contar sempre. Obrigado por ser essa pessoa tão especial na minha vida, eu te amo.

Agradeço ao meu orientador, Fábio Mendes, pelo apoio, confiança e paciência de me passar um pouco de seus conhecimentos. Agradeço também aos membros da banca, Paulo e Téo, pelas críticas construtivas e orientações.

Agradeço a todos os amigos do Laboratório Hacker, pelo carinho, apoio no desenvolvimento desse trabalho e compreensão pela ausência nos happy hours.

Agradeço a todos os professores que fizeram parte da minha graduação e ajudaram a formar o profissional que sou hoje.

Agradeço a todo o pessoal do LAPPIS e companheiros de curso pela incrível troca de conhecimentos e, claro, pelo bullying educativo. Agradeço também pelas amizades criadas durante todos esses anos.

Resumo

O processamento de linguagem natural tem sido utilizado com sucesso na área de análise de discurso, onde é possível reconhecer padrões e classificar textos, extraindo informações de grandes volumes de dados. Este trabalho tem como objetivo extrair o perfil temático dos deputados federais, através do processamento dos textos obtidos de seus discursos e proposições, bem como desenvolver uma aplicação *web* para que os resultados dessa pesquisa sejam apresentados de forma lúdica e amigável. O texto discute as técnicas de processamento de linguagem natural utilizadas nesta análise, que incluem a remoção de *stop words*, algumas técnicas de *stemização*, representação dos textos em *bag-of-words* e algumas técnicas de aprendizagem de máquina supervisionada e não-supervisionada, como *Naive Bayes* e k-means.

Palavras-chaves: Processamento de linguagem natural, aprendizado de máquina.

Abstract

Natural language processing has been used successfully in the analysis of discourse where is possible to recognize patterns and classify texts, extracting information from large volume of data. This paper aims to extract the thematic profile of the federal deputies through the processing of texts obtained from their speeches and proposals, as well as develop a web application in wich the results of this research are presented in a friendly way. The text discusses the natural language processing techniques used in this analysis, which include the removal of stop words, stemming, representation of texts in bag-of-words model, clusterization and classification and some techniques of supervised and unsupervised machine learning, such as Naive Bayes and k-means.

Key-words: Natural language processing, machine learning.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

1.1 Contextualização

O desenvolvimento de novas ferramentas de interação entre governo e sociedade é fundamental para o avanço da democracia e a disponibilização de dados governamentais possibilita que novas aplicações surjam, trazendo novas formas de utilização e interpretação destes dados (MAZONI, 2011). O presente trabalho utiliza dados disponíveis publicamente pela Câmara dos Deputados Federal para analisar textos de discursos e proposições parlamentares utilizando técnicas de processamento de linguagem natural e aprendizagem de máquina. O objetivo da análise é determinar eixos temáticos para cada deputado e avaliar o alinhamento entre os temas dos discursos e das propostas encaminhadas.

Com o objetivo de apresentar uma nova forma de visualizar os dados de discursos e proposições já disponibilizados pela Câmara dos Deputados, esse trabalho terá como produto um sistema *web* onde será possível, através de gráficos interativos, conhecer o perfil temático dos deputados federais. Para chegar à esse perfil, foram utilizadas técnicas de processamento de linguagem natural e aprendizado de máquina.

De acordo com a Lei n°12.527/2011, também conhecida como Lei de Acesso à Informação, qualquer cidadão, sem necessidade de justificativa, pode solicitar dados ou informações à qualquer órgão ou entidade pública dos poderes Executivo, Legislativo e Judiciário, além do Ministério Público, nas esferas Federal, Estadual e Municipal (BRASIL, 2011). Para atender à lei mencionada anteriormente, a Câmara dos Deputados (2016) criou um portal que tem como objetivo disponibilizar dados brutos para a utilização em aplicações desenvolvidas pelos cidadãos e entidades da sociedade civil que permitam a percepção mais efetiva das atividades parlamentares.

No contexto governamental, os Dados Abertos[1] fortalecem três características indispensáveis para a democracia: transparência, participação e colaboração (MAZONI, 2011). Transparência tem o papel de informar a sociedade sobre as ações que estão sendo tomadas ou que serão tomadas pelo governo. Participação permite que os cidadãos auxiliem o poder público a elaborar políticas mais eficazes. Finalmente, a colaboração entre a sociedade, diferentes níveis de governo e a iniciativa privada permitem aprimorar a eficácia

[1] O conceito para Dado Aberto considerado neste trabalho é o definido pela Open Knowledge (2016), que estabele que um dado (ou um conhecimento) é aberto quando estiver livre para uso, reuso e redistribuição. Ou seja, a informação deve estar disponível a todos, sem restrições de *copyright*, patentes ou outros mecanismos de controle. Além disso, esses dados devem ser independentes de tecnologia, baseados em formatos padronizados e desvinculados das ferramentas que os originaram. Os dados devem permitir sua manipulação por máquinas e possuir metadados que permitam identificar sua natureza, origem e qualidade (DINIZ, 2010).

do Estado.

A publicação de dados pressupõe o uso de tecnologias que garantam que eles possam ser acessados e reutilizados por máquinas. Apesar de não garantir que os dados estarão disponíveis em um formato conveniente de uso imediato, possibilita o cruzamento de diferentes bases dados e a exibição destes de forma que possam ser melhor apresentados à sociedade (DINIZ, 2010). O presente trabalho utiliza os dados disponíveis da Câmara dos Deputados e apresenta algumas ferramentas para obtê-los de formas mais convenientes para o posterior tratamento.

1.2 Objetivo

O objetivo desse trabalho é utilizar técnicas de processamento de linguagem natural para, através da análise dos discursos e proposições dos parlamentares, determinar um perfil temático para os deputados, bem como evidenciar o termos mais utilizados em seus discursos e proposições e, assim, permitir que o cidadão veja, de forma comparativa, o que seus representantes no parlamento mais dizem em seus discursos e o que mais dizem em suas proposições.

1.2.1 Contribuições Tecnológicas

- Implementar biblioteca Python para consumo de dados abertos governamentais, com foco nos dados abertos da Câmara dos Deputados.
- Implementar aplicação Django para utilização e persistência dos dados abertos governamentais, também com foco nos dados aberto da Câmara dos Deputados.
- Implementar sistema web de comparação temática entre discursos e proposições parlamentares, a ser detalhado no decorrer deste trabalho.

1.3 Metodologia

Devido à natureza deste trabalho, nota-se que o modelo de pesquisa adequado deve possuir características tanto da pesquisa exploratória quanto da pesquisa experimental. Outro método de pesquisa que será utilizado é a pesquisa-ação, onde existirão ciclos de coleta e análise de dados. A cada ciclo, a análise dos dados do ciclo anterior servirão de insumo para tomadas de decisão no desenvolvimento do projeto.

O presente trabalho possui o apoio do Laboratório Hacker da Câmara dos Deputados e por isso foi desenvolvido em colaboração com membros de sua equipe de desenvolvimento. Para facilitar a interação com os outros envolvidos no projeto, foi criada uma

organização no *GitHub*[2], onde foram armazenados os códigos produzidos nesse trabalho. Além disso, a aplicação será hopedada na infraestrutura do Laboratório Hacker.

Por motivos de força maior, o presente trabalho não foi completado até a data limite para sua entrega e a versão atualizada do texto poderá ser obtida no repositório do autor[3], bem como a versão final do código poderá ser acessado no repositório do Tenho Dito[4].

1.4 Organização do Trabalho

Esse trabalho está organizado em 6 capítulos: Introdução, Processamento de Linguagem Natural, Aprendizado de Máquina, Metodologia, Resultados Obtidos e Considerações Finais. O presente capítulo faz uma breve introdução ao tema desse trabalho, bem como aos objetivos que pretendem ser alcançados. Os capítulos 2 e 3 contém todas o referencial teórico necessário para a realização desse trabalho. O capítulo 4, Metodologia, descreve a forma como foi realizado esse trabalho. O capítulo 5 expõe os objetivos alcançado ao final do trabalho. O capítulo 6 possui a conclusão a que se chega com o presente trabalho e as perspectivas futuras.

[2] https://github.com/tenhodito
[3] https://github.com/msfernandes/tcc
[4] https://github.com/tenhodito/tenhodito

2 Processamento de Linguagem Natural

Este capítulo contém o referencial teórico que diz respeito ao Processamento de Linguagem Natural.

2.1 Pré-processamento: modelo *bag-of-words* (BOW)

A maior parte dos dados utilizados nesse trabalho estão dispostos em formato de texto, ou seja, um formato não estruturado que dificulta a extração de informações. Um método comum do processamento de texto em linguagem natural é o modelo *bag-of-words*, onde o texto é representado na forma de um vetor de frequência de palavras (MATSUBARA; MONARD, 2003).

A representação computacional de um texto no modelo mencionado é feita através de um dicionário onde suas chaves são os termos presentes no documento e os valores são as respectivas frequências. Tomando o trecho "fui à padaria e comprei pão" como exemplo, sua representação no modelo *bag-of-words* corresponderia a:

```
bag_of_words = {
    "fui": 1,
    "à": 1,
    "padaria": 1,
    "e": 1,
    "comprei": 1,
    "pão": 1,
}
```

Listing 2.1 – Representação de um trecho no modelo bag-of-words

É fácil ver que esta representação torna a ordem das palavras irrelevante. As frases "e comprei fui padaria pão à" e "à pão comprei padaria e fui" também possuem as mesmas representações.

O texto também pode ser representado na forma de um vetor. Nesse caso, cada índice do vetor representa uma palavra e cada coordenada corresponde à frequência da mesma. Quando mais de um texto é analisado, os vetores que os representam devem possuir a mesma relação "índice-palavra". Por isso, as palavras que compõe cada texto são ordenadas alfbéticamente antes da formação do vetores.

A linguagem de programação *Python* possui, nativamente, ferramentas que facilitam a contagem de elementos de um dicionário. O módulo `collections`[1] possui a classe

[1] https://docs.python.org/3/library/collections.html

`Counter`, que é uma subclasse de `dict` e representa um mapa entre elementos e números. Isso torna o `Counter` uma representação ideal para o modelo *bag-of-words*, já que ele, naturalmente, conta a quantidade de vezes que um termo aparece dentro de uma coleção (uma lista, tupla ou dicionário, por exemplo).

2.1.1 Valores no BOW

Apresentamos o modelo do BOW associando um valor a_i a cada palavra que representa, como a quantidade de vezes que o termo aparece em um texto. Existem, no entanto, outras formas de representação expostas a seguir.

2.1.1.1 *Boolean*

Essa medida associa valores binários para os termos presentes nos documentos, onde o valor de a_i é 0 quando o termo não aparece nenhuma vez ou 1 quando aparece uma ou mais vezes. Essa medida simples é muitas vezes utilizada para a análise de documentos pequenos, como frases e parágrafos.

2.1.1.2 *Term Frequency*

A medida *term frequency* considera a quantidade de ocorrencias do termo t dentro do documento, ao contrário da medida mencionada anteriormente (SALTON; BUCKLEY, 1988)

$$f(t) = \frac{n_t}{N}, \tag{2.1}$$

onde n_t é a quantidade de vezes que o termo t aparece dentro do documento e N a quantidade total de termos do documento.

2.1.1.3 *Term Frequency - Inverse Document Frequency (TF-IDF)*

Alguns termos comuns podem aparecer na maioria dos documentos sem fornecer informações úteis em uma tarefa de mineração de textos. Para diminuir a influência destes termos, é possível utilizar um fator de ponderação, para que os termos que aparecem na maioria dos documentos tenham valores numéricos menores do que aqueles que raramente aparecem (MATSUBARA; MONARD, 2003). Segundo JONES (1972), a especificidade de um termo pode ser quantificada por uma função inversa do número de documentos em que ele ocorre. Uma alternativa comum é utilizar uma função que varia entre 0 e $logN_d$, onde N_d é o número total de documentos e $d(t)$ a quantidade de documentos nos quais o termo t aparece ao menos uma vez:

$$i(t) = log\frac{N_d}{d(t)} \tag{2.2}$$

Portanto, o valor final de a_i é dado pela equação:

$$f(t) = \frac{n_t}{N} \cdot log \frac{N_d}{d(t)}, \tag{2.3}$$

onde $\frac{n_t}{N}$ é a frequência do termo dentro do texto e $log \frac{N_d}{d(t)}$ sua taxa de ponderação.

Existem outras formas de ponderação que penalizam os termos mais comuns de formas mais ou menos extremas. O formato logarítmo possui a característica de anular termos que apareçam em todos os documentos, considerando-os portanto, como totalmente não-informativos.

2.1.2 Modelo *N-Gram*

O modelo de representação de textos através de um conjunto de palavras pode limitar qualitativamente a análise, já que termos compostos não são considerados. Por exemplo, "Santa Catarina", um estado brasileiro, possui um significado completamente diferente quando as palavaras "Santa" (mulher canonizada), e "Catarina" (nome feminino) são analisadas separadamente.

Um n-grama é uma sequência de n elementos dentro de um texto. Os elementos podem ser palavras, sílabas, letras ou qualquer outra base. É comum usar as denominações unigrama, bigrama e trigrama para n-gramas de 1, 2 ou 3 elementos. Usando a frase "Eu não gostei desse filme" como exemplo, temos os seguintes unigramas: "eu", "não", "gostei", "desse" e "filme". Os seguintes bigramas: "eu não", "não gostei", "gostei desse" e "desse filme". E os seguintes trigramas: "eu não gostei", "não gostei desse" e "gostei desse filme".

2.1.3 Dimensionalidade dos documentos

A representação vetorial de uma coleção de documentos no modelo *bag-of-words* pressupõe um espaço dimensional igual ao número de termos presentes em toda a coleção de documentos. Suponha que analisou-se 10 documentos e, em média, foram retirados 200 novos termos de cada um. Logo, a dimensionalidade média dos vetores sera de, aproximadamente, 2000. Se a maior parte dos termos aparecer em apenas um ou dois documentos, teremos a maior parte das componentes vetoriais nulas. (MATSUBARA; MONARD, 2003). Existem métodos cujo objetivo é diminuir a dimensionalidade desses vetores. Dentre eles, citamos a transformação de cada termo no radical de origem, utilizando algoritmos de *stemming*.

2.1.3.1 Stemização

A stemização (do inglês, *stemming*) é o processo de reduzir palavras flexionadas à sua raiz. Essa redução não precisa, necessariamente, chegar à raiz morfológica da palavra, mas apenas a um valor útil do ponto de vista computacional. A raiz obtidada geralmente

é o suficiente para mapear palavras relacionadas à um valor comum, mesmo se este não for uma raiz válida. O estudo de algoritmos de *stemming* é foco de pesquisas desde a década de 60 e o primeiro algoritmo foi publicado por Lovins (1968).

A consequência da aplicação de algoritmos de *stemming* consiste na remoção de prefixos ou sufixos de um termo e ou da transformação de verbos para suas formas no infinitivo. Por exemplo, as palavras **ouvir**, **ouvi**, **ouviriam**, **ouve** e **ouvindo** seriam reduzidas para um mesmo *stem*: **ouv**. Esse método diminui, portanto, a dimensionalidade dos vetores e dicionários dentro de uma *bag-of-words*. Ao invés de analizar a frequência dos termos, analisamos a quantidade de vezes que um *stem* aparece em um documento.

É evidente que os algoritmos de *stemming* são dependentes do idioma analisado. O algoritmo de Porter (1980), um dos algoritmos de *stemming* mais conhecidos, remove os sufixos de termos em inglês, e tem sido amplamente utilizado, referenciado e adaptado desde sua criação. É possível adaptá-lo para a língua portuguesa considerando que as línguas provenientes do latim possuem formas verbais conjugadas em sete tempos e com sete terminações distintas.

Devido ao fato de uma linguagem ter tantas regras e exceções, é pouco provavél que o algoritmo de *stemming* retorne o mesmo *stem* para todas as palavras que tenham a mesma origem ou radical morfológico. Pode-se dizer, também, que a medida que o algoritmo vai se tornando específico o suficiente para atender todas essas regras e exceções a eficiência do algoritmo também diminui, assim como a dificuldade de implementação aumenta (IMAMURA, 2001).

2.1.3.2 *Stop Words*

Palavras que possuem pouco ou nenhum valor semântico, como *"e"*, *"de"* e *"seus"*, são conhecidas como *Stop Words* e, por não agregarem valor à analise textual, podem ser removidas durante o pré-processamento (RAJARAMAN; ULLMAN, 2011). Essas palavras não são exclusividade de uma linguagem específica e geralmente representam a maioria dos termos de um texto. No caso da língua inglesa, por exemplo, palavras como *"of"* e *"the"* também não possuem nenhum valor para a análise. A lista de *Stop Words* obviamente varia de acordo com a linguagem que está sendo analisada (LOPES, 2015) e do contexto da análise. Neste trabalho, os textos analisados são provenientes de discursos parlamentares, o que implica na utilização de *stop words* específicas para o contexto legislativo. Portanto, além das palavras mais comuns da língua portuguesa, também serão removidos da análise termos mais característicos de discurso legislativo, como “vossa excelência”, “senhor” e “pronunciamento”, por serem constantemente ditas pelos deputados em seus discursos e não agregarem muito valor à análise.

O quadro abaixo mostra a lista parcial de *Stop Words* consideradas nesse trabalho.

de	os	tua	tem	estão	da	lhes	essas
e	é	foi	nossas	muito	o	se	tuas
tu	por	as	sua	aquele	entre	não	ele
delas	minhas	às	nos	pela	havia	me	como
ser	aqueles	nossa	vocês	eu	ter	tenho	suas
está	isso	pelos	estes	tinha	depois	foram	este
para	só	quem	deles	isto	um	eles	do
vos	mais	mesmo	num	dele	será	minha	a
no	teus	à	você	em	meus	esses	pelas
com	ao	dela	há	que	na	nosso	te
aos	dos	ou	aquela	era	uma	das	esta
teu	nem	já	até	seja	esse	mas	quando
aquelas	nossos	têm	também	seus	lhe	meu	seu
ela	elas	estas	nós	sem	essa	fosse	qual
		pelo	nas	numa	aquilo		

Tabela 1 – Lista parcial de *Stop Words* consideradas nesse trabalho

O algoritmo utiliza a versão *stemizada* das *stop words*. As palavras “ele”, “ela”, “eles” e “elas”, por exemplo, seriam todas tratadas como um único termo: “el”. Mostramos as palavras originais por uma questão de clareza.

3 Aprendizado de Máquina

O aprendizado de máquina tem como objetivo criar técnicas computacionais e sistemas que automaticamente adquirem conhecimento. Existem diversos algoritmos de aprendizado de máquina, utilizados para resolver problemas específicos. É importante, portanto, compreender suas limitações (REZENDE, 2003).

As tarefas de aprendizado de máquina podem ser classificadas em:

- **Apredizado supervisionado:** é fornecido ao algoritmo um conjunto de entradas e suas respectivas saídas, com o objetivo de aprender uma regra geral que mapeia as entradas às saídas.
- **Aprendizado não-supervisionado:** somente um conjunto de entrada é fornecido, com o objetivo do próprio algoritmo identificar os padrões do conjunto de dados.
- **Aprendizado por estímulo:** o algoritmo interage com o ambiente dinâmico afim de concluir determinados objetivos.
- **Aprendizado semi-supervisionado:** utiliza um conjunto de treinamento que inclui alguns elementos pré-classificados junto com outros elementos não classificados. O objetivo é obter uma performance que se aproxime dos supervisionados com um custo de classificação que se aproxime dos não-supervisionados.

Este trabalho utiliza alguns algoritmos supervisionados e não-supervisionados. Apesar de não utilizar algorítmos de específicos de aprendizado semi-supervisionado, serão utilizados termos previa e manualmente rotulados.

3.1 Aprendizado Supervisionado

No aprendizado supervisionado, cada exemplo de treinamento é descrito por um conjunto de atributos que servem como dados de entrada e são associados a um valor de saída. A partir de um conjunto pré-definido de entradas e saídas, o algoritmo consegue gerar uma saída adequada para uma nova entrada. O aprendizado supervisionado é a principal técnica utilizada para problemas de classificação e regressão (MOHRI; TALWALKAR, 2012).

3.1.1 Aprendizado Bayesiano

O aprendizado Bayesiano é um conjunto de técnicas baseadas em análise estatística que utilizam a fórmula de Bayes. Normalmente são métodos supervisionados, ainda

que alguns algoritmos não-supervisionados possam ser mapeados em métodos Bayesianos (MITCHELL, 1997).

As principais vantagens do aprendizado Bayesiano são o fato de se poder embutir nas probabilidades calculadas o conhecimento de domínio (caso se tenha) e a capacidade das classificações feitas pelo algoritmo se basearem em evidências fornecidas e numa análise estatística bem fundamentada. Por outro lado, frequentemente envolvem o cálculo de médias e outras medidas estatísticas que pode ocasionar em um alto custo computacional.

3.1.1.1 Classificador *naive Bayes*

Uma forma de mitigar a dificuldade de cálculo está em considerar modelos probabilísticos simplificados que permitem um tratamento analítico para as probabilidades calculadas (PARDO; NUNES, 2002).

O classificador Bayesiano ingênuo (ou *naive Bayes*, em inglês), admite que os atributos do elemento a ser classificado são independentes entre si, dada a categoria da classificação (PELLUCCI et al., 2011). Segundo Oguri e Renteria (2007), existem dois tipos principais de classificadores bayesianos ingênuos utilizados em processamento de linguagem natural: o modelo binário e o modelo multinomial. Cada modelo está relacionado a um tipo de BOW. Também é possível tratar a BOW TF-IDF como um modelo Gaussiano adequado para variáveis contínuas (HAND; YU, 2001).

3.1.1.1.1 Modelo Binário

O modelo binário representa um documento como um vetor binário, ou seja, o valor 0 em uma posição k (onde k representa uma palavra do documento) representa a não ocorrência do termo e o valor 1 representa ao menos uma ocorrência desse termo. Este modelo simplesmente especifica a probabilidade de ocorrência de cada termo.

3.1.1.1.2 Modelo Multinomial

Já o modelo multinomial assume que o documento é representado por um vetor de inteiros, representando a quantidade de vezes que um termo ocorre no documento. Este modelo também especifica a probabilidade de ocorrência de um termo, mas permite ocorrências múltiplas.

Os classificadores Bayesianos são baseados na aplicação do Teorema de Bayes:

$$P(classe|A) = \frac{P(classe) \times P(A|classe)}{P(A)}, \tag{3.1}$$

onde:

- $P(classe)$ é a probabilidade da classe em questão, no contexto do teorema de Bayes, ela é comumente denominada probabilidade *a priori* (ou *prior*)
- $P(A|classe)$ é a probabilidade de obter um conjunto de dados A condicional a $classe$. Isto é conhecido como *likelihood* ou verossimilhança. O modelo Bayes ingênuo assume que $P(A|classe) = \prod_i P(a_i|classe)$
- $P(classe|A)$ é a probabilidade de um elemento pertencer a uma classe dado um conjunto de observações A. Conhecido como o *a posteriori* na literatura Bayesiana
- $P(A)$ é a probabilidade da nova instância a ser classificada. Este termo corresponde ao fator de normalização do posterior e é frequentemente ignorado. Na literatura Bayesiana é conhecido como evidência (JAYNES, 2003)

Para calcular a classe mais provável da nova instância, calcula-se as probabilidades de todas as classes possíveis e escolhe-se a classe com maior probabilidade. Em termos estatísticos, isso é equivalente a maximizar $P(classe|A)$. Como muitas vezes utiliza-se um prior uniforme ao longo das classes, o problema matemático principal frequentemente consiste em encontrar o valor de máxima verossimilhança.

Considerando que $A = (a_1, a_2, \ldots, a_n)$, onde a_n são os atributos que compõe A, a suposição "ingênua" que o classificador faz é que todos os atributos de A são independentes entre si, o que simplifica o cálculo da probabilidade de $P(A|classe)$, podendo ser reduzida a $P(a_1|classe) \times P(a_2|classe) \times \ \ldots \times P(a_n|classe)$. Logo,

$$P(classe|A) \propto P(classe) \times \prod_{i=1}^{n} P(a_i|classe) \tag{3.2}$$

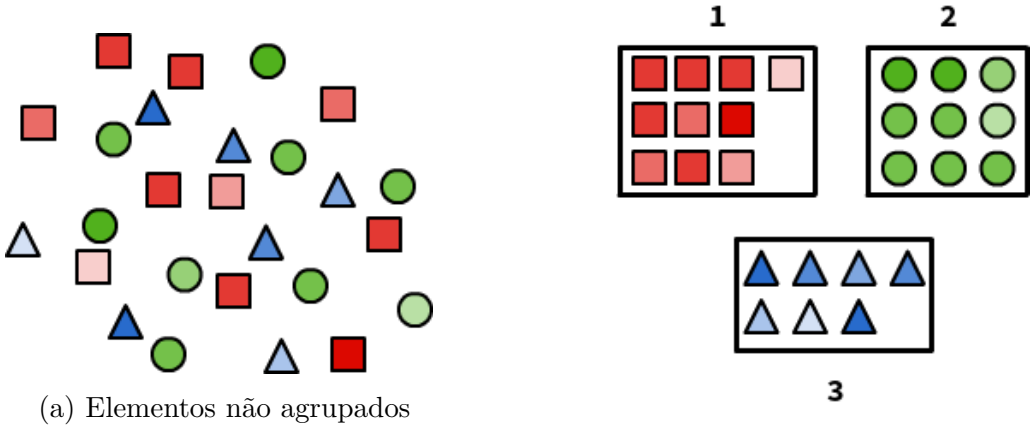
Em problemas de NLP, a_i usualmente correspondem aos valores contidos em uma BOW, ainda que seja possível adicionar outros tipos de atributos para a análise.

3.2 Aprendizado Não Supervisionado

Na aprendizagem não supervisionada, os dados de entrada não possuem classes (ou rótulos) e o objetivo do algoritmo é descrever estruturas dentro do conjunto de dados. Uma vez que os dados não são classificados, não existe um erro ou uma recompensa, o que distingue o aprendizado não supervisionado da aprendizagem supervisionada ou por esforço. A aprendizagem não supervisionada é bastante utilizada para resumir e explicar as principais características dos dados (JORDAN; CHRISTOPHER, 2004).

3.2.1 Clusterização

Quando temos um conjunto grande de elementos, naturalmente tentamos estabelecer padrões entre eles. Uma forma natural de definir padrões em um conjunto é analisar a distância entre seus componentes. Dessa forma, quanto mais parecidos dois elementos são, mais próximos eles estão. A figura abaixo mostra um conjunto de elementos com duas características: forma (quadrado, círculo e triângulo) e cor (tons de vermelho, verde e azul). Ao lado temos os mesmos elementos agrupados em três conjuntos com características semelhantes. No grupo 1, por exemplo, todos os elementos possuem uma tonalidade de vermelho e o formato quadrado.



(a) Elementos não agrupados

(b) Elementos agrupados

Figura 1 – Exemplo de clusterização

A clusterização é uma técnica da mineração de dados que consiste, justamente, em realizar o procedimento descrito acima: organizar um conjunto de elementos, usualmente representados por vetores ou pontos em um espaço multidimensional, em *clusters* (ou agrupamentos), de acordo com alguma medida de similaridade. Ela representa uma das principais etapas da análise de dados, denominada análise de *clusters* (JAIN; FLYNN, 1999).

Não existe uma técnica de clusterização universal capaz de revelar toda a variedade de estruturas que podem estar presentes em conjuntos de dados multidimensio-

nais. Diferentes algoritmos dependem implicitamente de certas hipóteses a respeito da forma dos clusters, da definição da medida de similaridade e dos critérios de agrupamento (ESTIVILL-CASTRO, 2002).

3.2.1.1 Algorítmo *k-means*

O algorítmo de clusterização *k-means*, proposto por Lloyd (1957), tem o objetivo de dividir N elementos em k grupos, onde cada elemento pertence ao *cluster* mais próximo. O valor de k deve ser informado a priori, sendo menor que a quantidade de elementos.

Os passos do algoritmo são:

1. **Gerar centróides:** neste passo os k centróides recebem valores iniciais. O valor inicial dos centróides podem ser definidos randomicamente, através de uma Gaussiana (com média e variância estimados a partir do conjunto de elementos) ou escolhendo aleatoriamente k dos N elementos como centróides iniciais ou definindo-os como centróides de k grupos escolhidos aleatoriamente a partir dos dados iniciais.

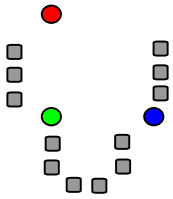


Figura 2 – k centróides (coloridos) recebem valores iniciais.

2. **Calcular distâncias:** aqui são calculadas as distâncias entre cada ponto e cada centróide. É a parte com maior peso computacional do algorítmo, já que o calculo é realizado para cada ponto.

3. **Classificar os pontos:** cada ponto deve ser classificado de acordo com a distância entre ele e o centróide de cada *cluster*. O ponto pertencerá ao *cluster* cujo centróide está mais próximo. O algorítmo converge quando, em uma iteração, nenhum ponto mudar de *cluster*.

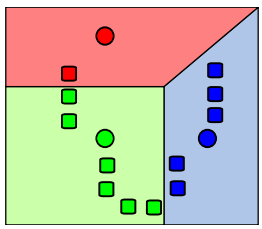


Figura 3 – Cálculo das distâncias entre os pontos e os centróides.

4. **Calcular novos centróides:** para cada *cluster*, um novo centróide é definido como a média de todos os pontos.

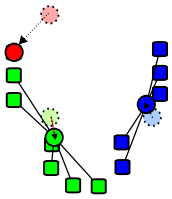


Figura 4 – Novos centróides definidos pela media dos elementos do *cluster*.

5. **Repetir até convergir:** retorna ao passo 2. Como o resultado do algoritmo depende da escolha dos centróides iniciais, a convergência não é garantida ou ele converge para uma solução sub-ótima. Por isso, normalmente o algoritmo é executado várias vezes.

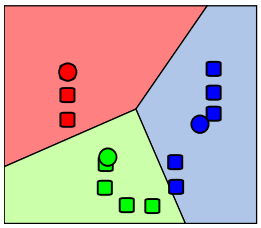


Figura 5 – O algorítmo converge quando nenhum ponto muda de *cluster*.

Mudanças de escala ou de unidade de medidas para determinadas coordenadas dos elementos podem afetar a análise (COLE, 1998). Sugere-se, então, que seja feito o processo de normalização (*whitening*) dos dados antes da clusterização. A normalização consiste em ajustar a escala das distâncias de forma que os valores fiquem em intervalos padronizados, normalmente com média nula e variância 1.

3.2.1.2 *Latent Dirichlet Allocation - LDA*

O *LDA* foi proposto por Pritchard e Donnelly (2000) em uma aplicação inicial na biologia. Em seu modelo, Pritchard e Donnelly (2000) utilizam dados de genótipos multilocus para inferir estruturas populacionais e associar cada indivíduos a uma mistura de populações parentais. Dessa forma, também conseguem estudar zonas híbridas entre as populações e identificar indivíduos migrantes e misturados. A aplicação desse modelo no processamento de linguagem natural foi proposta independentemente por Blei, Ng e Jordan (2003), onde os documentos são tratados como os indivíduos e as populações passam a ser tópicos.

No contexto do processamento de linguagem natural, um documento, ou palavra, pode ser representado por uma mistura de vários tópicos, o que permite uma melhor desambiguação dos termos e uma definição de tópicos mais precisa (GIROLAMI; KABAN, 2003). Usamos como exemplo as seguintes frases:

1. Eu **como peixe**.

2. *Peixes* são *animais*.

3. Meu *gato* **come peixe**.

O *LDA* poderia classificar os termos em **negrito** como pertencentes ao **Tópico C**, que poderia ser rotulado como "**comida**". Da mesma forma, as palavras em *itálico* também poderiam ser agrupadas em um *Tópico A*, rotulado como "*animais*".

Esse modelo tenta inferir duas tabelas de probabilidade: estrutura (P) e mistura (Q). Em P temos as probabilidades de um termo estar presente em uma frase de cada tópico, sendo que essas probabilidades são independentes entre si. Já em Q, temos as probabilidades de cada documento pertencer à cada tópico, mas, nesse caso, as probabilidades devem somar um total de 100%. Foram utilizados dados hipotéticos nas tabelas abaixo apenas para ajudar na explicação.

Termo/Tópico	**Comida**	**Animal**
comer	0.9	0.3
peixe	0.5	0.6
animal	0.2	0.8
gato	0.2	0.8

Tabela 2 – Tabela de Estrutura dos Termos (P)

Documento/Tópico	**Comida**	**Animal**
Frase 1	0.8	0.2
Frase 2	0.1	0.9
Frase 3	0.6	0.4

Tabela 3 – Tabela de Mistura dos Documentos (Q)

Essas tabelas influenciam diretamente uma à outra, ou seja, se um termo começa a ser predominante em determinado tópico, o documento a qual ele pertence passa a tender à esse tópico. Da mesma forma que se um documento começar a tender a um tópico, seus termos também serão favorecidos neste tópico.

Para se obter o conteúdo de cada tópico, basta observar as colunas de P, onde teremos uma *bag of words* contendo a probabilidade de todos os termos pertencerem à esse tópico. Analisando as linhas de Q, temos as misturas de tópicos de cada documento. Vale notar que, por ser uma mistura de tópicos, o *LDA* dificilmente vai atribuir um documento à um único tópico com 100% de pertencimento.

Para chegar à essa conclusão, o *LDA* segue três passos:

- **Quantidade de tópicos:** assim como no *k-means*, o primeiro passo é definir a quantidade de tópicos que serão obtidos ao final da análise. Esse número pode ser obtido através de uma análise prévia dos dados ou simplesmente por uma escolha aleatória.

- **Inicializar as tabelas P e Q:** É definida uma distribuição de tópico inicial para cada documento, que será atualizado no passo seguinte. A escolha desses tópicos se dá de forma semi-aleatória, de acordo com a distribuição de *Dirichlet*, ou seja, valores aleatórios entre 0 e 1 com a restrição de que a soma deles deve ser 1. As probabilidades P podem ser definidas de forma uniforme para todos os termos: $P = 0.5$.

- **Analisar e atualizar os tópicos:** Para cada palavra, em todos os documentos, a definição do tópico é atualizada de acordo com dois critérios: o quanto essa palavra é predominante entre os tópicos e quão predominante são os tópicos entre as palavras do documento. Em outras palavras, as probabilidades são recalculadas fixando os valores de P para obter os novos valores de Q e, em seguida, fixa-se os valores de Q para recalcular as probabilidades de P. Dependendo do algoritmo esse processo pode ser repetido para cada palavra e em todos os documentos, passando por todo o *corpus* várias vezes até convergir (ANNALYN, 2016).

3.3 Aprendizado Semi-Supervisionado

Essa forma de aprendizado é útil quando quando existem apenas alguns exemplos já classificados e pode ser utilizado tanto em tarefas de classificação quanto em tarefas de *clustering*. A ideia do aprendizado semi-supervisionado é utilizar esses exemplos previamente classificados para se obter informações sobre o problema e utilizá-las para auxiliar o processo de aprendizado a partir de exemplos não classificados (BRUCE, 2001).

Existem várias estratégias para definir algoritmos de aprendizado semi-supervisionado. É comum utilizar modelos de aprendizado não-supervisionado com alterações, afim de se poder inserir informações a priori.

Podemos tornar o algoritmo *k-means* semi-supervisionado se fixarmos *clusters* específicos para determinados termos previamente classificados, de forma que esse eles sempre pertençam aos *clusters* pré-fixados. Dessa forma, a informação inserida influenciará diretamente na posição dos *clusters*.

No *LDA*, também podemos fixar algumas linhas da tabela de mistura Q. Dessa forma, podemos dizer que um texto pertence a um tópico específico ou a misturas específicas de tópicos.

3.4 Distância entre os pontos

Segundo Cole (1998), para clusterizar termos de acordo com sua similaridade, deve-se definir uma medida de quão próximos dois termos estão. Uma medida de distância (métrica) deve ser definida de tal forma que:

- Seja sempre positiva.
- Seja simétrica: a distância de um termo A_i para um termo A_j deve ser a mesma de A_j para A_i.
- Seja reflexiva: se a distância entre A_i e A_j é zero, então $A_i = A_j$.
- Respeite a desigualdade triangular: considerando os termos (A_i, A_j e A_k), a distância $d(A_i, A_k)$ deve ser menor ou igual à soma das distâncias $d(A_i, A_j)$ e $d(A_j, A_k)$

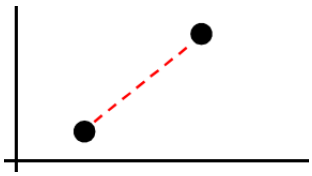
Existem várias medidas de distância. Começamos pela distância Euclidiana entre dois pontos, $A = (a_1, a_2, a_3, \ldots, a_n)$ e $B = (b_1, b_2, b_3, \ldots, b_n)$, dada pela equação

$$dist(A, B) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} \tag{3.3}$$

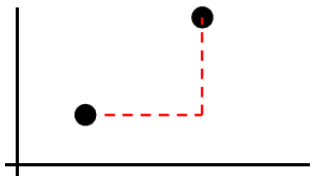
Já a distância de Manhattan entre dois pontos, $A = (a_1, a_2, a_3, \ldots, a_n)$ e $B = (b_1, b_2, b_3, \ldots, b_n)$, é dada pela soma das diferenças absolutas de suas coordenadas:

$$dist(A, B) = \sum_{i=1}^{n} |a_i - b_i| \tag{3.4}$$

Em ambos os casos, temos que a distância entre do ponto A ao ponto B é a mesma distância do ponto B ao ponto A.



(a) Distância Euclidiana



(b) Distância de Manhattan

Figura 6 – Comparação entre distância Euclidiana e de Manhattan

Cada métrica pode gerar resultados diferentes no algoritmo *k-means* e normalmente podem ser interpretadas do ponto de vista estatístico como uma escolha do modelo que gerou cada *cluster*. A distância euclidiana pode ser associada a um modelo Gaussiano para gerar os pontos observados onde a média corresponde ao centróide de cada *cluster* e o desvio padrão é o mesmo para todos os grupos.

3.5 Validação Cruzada

A validação cruzada é uma técnica geralmente utilizada para avaliar modelos preditivos, buscando estimar o quão preciso é o modelo avaliado. A ideia principal da validação cruzada consiste em dividir o conjunto de dados em subconjuntos mutualmente exclusivos, onde alguns desses subconjuntos serão utilizados para a estimação dos parâmetros do modelo (treinamento) e os outros subconjuntos serão utilizados para a validação do modelo (validação ou teste) (KOHAVI, 1995).

O método *holdout* é uma forma de validação de cruzada comumente utilizada e consiste em dividir o conjunto de dados em dois subconjuntos mutualmente exclusivos, onde seus tamanhos podem, ou não, ser diferentes. Um subconjunto servirá para o treinamento do modelo e o outro para a sua validação. Uma proporção comum é considerar 2/3 dos dados para treinamento e 1/3 para a validação. Essa abordagem é indicada quando está disponível uma grande quantidade de dados. Quando o conjunto de dados é muito pequeno, o erro calculado na predição pode sofrer muita variação (KOHAVI, 1995).

Uma das métricas mais simples para avaliar o modelo é a comparação item a item dos resultados esperados pelos resultados obtidos:

$$P_a = \frac{N_{vp}}{N_{it}}, \tag{3.5}$$

onde P_a é a porcentagem de acertos do modelo, N_{vp} é o número de itens classificados corretamente pelo modelo e N_{it} é o número total de itens que fazem parte do conjunto de testes.

4 Metodologia

Este capítulo aborda o planejamento e execução do projeto, contendo os procedimentos e técnicas utilizadas, possibilitando a sua replicação. Tendo em vista os conceitos descritos no capítulo 2 (Processamento de Linguagem Natural), o presente trabalho tem como objetivo responder à seguinte questão problema:

É possível extrair o perfil temático dos deputados através da análise dos seus discursos e proposições utilizando técnicas clássicas de aprendizado de máquina e processamento de linguagem natural?

Além disso, será construído um *website* com o intuito de fornecer uma forma melhor de visualização dos dados obtidos nesta análise, através de gráficos interativos que garantam que o usuário tenha uma boa experiência de usabilidade.

4.1 Trabalhos Relacionados

Durante a pesquisa bibliográfica realizada neste trabalho, encontrou-se alguns trabalhos que também fizeram análise de textos parlamentares utilizando aprendizado bayesiano. As seções a seguir descrevem brevemente alguns deles.

4.1.1 Retórica Parlamentar

O Retórica Parlamentar[1], idealizado por Davi Moreira[2], Manoel Galdino[3] e Luis Carli[4], utiliza os discursos proferidos pelos parlamentares no Pequeno Expediente e no Grande Expediente da Câmara dos Deputados para promover a transparência do mandato e fornecer subsídios para o controle social com a divulgação dos temas mais debatidos em Plenário.

A técnica utilizada pelo Retórica para a classificação dos discursos é um modelo bayesiano hierárquico, descrito por Grimmer (2009), onde através de aprendizado não supervisionado são gerados k *clusters*, sendo k um valor escolhido ao executar o algoritmo. O resultado é exportado para o formato *csv* e contém os termos mais frequentes de cada cluster. Em seguida, um especialista deve ler e rotular cada *cluster*.

[1] http://retorica.labhackercd.net/about.html
[2] https://github.com/davi-moreira
[3] https://github.com/mgaldino
[4] https://github.com/luiscarli

A visualização dos dados é feita através de um gráfico de bolhas, em que cada bolha representa a relevância (medida pela frequência) de cada tema dentre todos os deputados analisados. Dentro de cada bolha são colocados os deputados que enfatizam aquele tema nos seus discursos. Um deputado está associado a um único tema, que é o tema mais enfatizado por ele nos seus discursos.

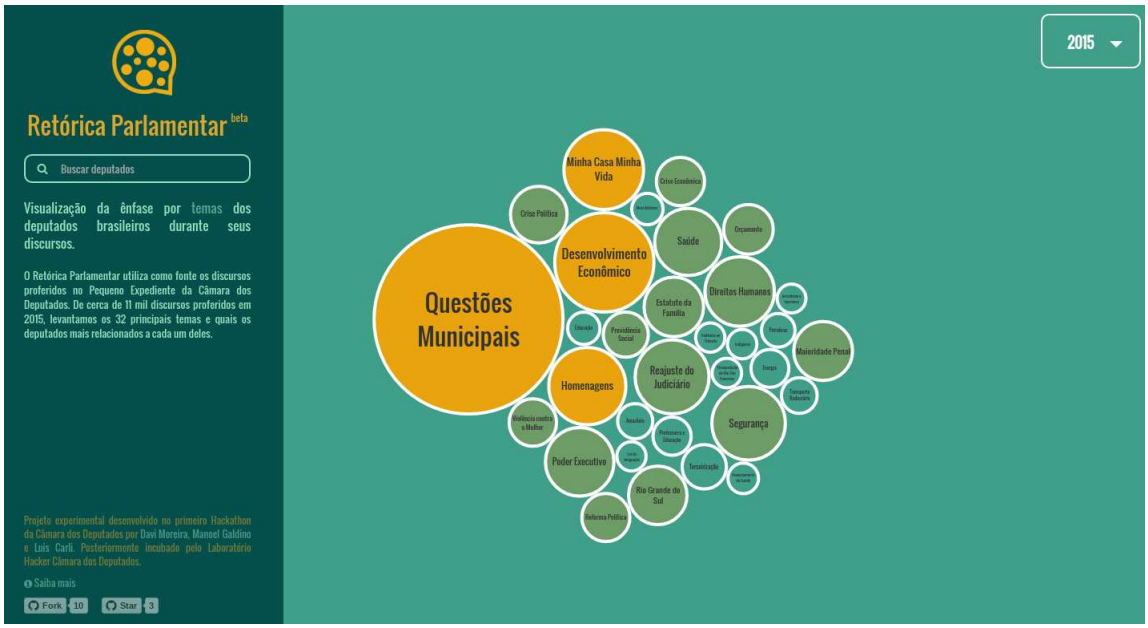


Figura 7 – Retórica Parlamentar

4.1.2 *Automatic content analysis of legislative documents by text mining techniques*

Outro trabalho encontrado foi o de Lin Shih-Yao Chou e Hao (2015), onde propõem uma análise automática dos dados do parlamento de Taiwan, como questionamentos legislativos, discursos em conferências, proposicões legislativas e proposições provisórias do Legislativo de Yuan (poder legislativo unicameral da República da China), utilizando técnicas de processamento de linguagem natural, com o objetivo de representar o desempenho de cada legislador em determinadas categorias, definidas por especialistas do *Institute od Political Science* da *National Sun Yat-sen University*.

Lin Shih-Yao Chou e Hao (2015) definem três aspectos para a análise dos dados do parlamento de Taiwan: direção geográfica, público-alvo e tópico. A direção geográfica se refere ao escopo geográfico abrangido pelos registros legislativos e tem 5 direções. O público-alvo refere-se às pessoas mencionadas explicitamente em seus registros, podendo ser 30 opções. O tópico inclui os temas abordados pelos parlamentares, que possuem um total de 29 tópicos. A figura abaixo mostra os três aspectos e seus valores correspondentes:

Figura 8 – Estrutura de categorização

Após o processo de pré-processamento do texto, os dados passam por duas fases de clusterização. A primeira fase consiste em uma classificação hierárquica, com o objetivo de obter o número de *clusters* ideal. Em seguida, é aplicado o *k-means*, utilizando o número obtido no processo anterior como valor de k. Dessa forma conseguem melhorar a relevância dos termos que serão apresentados aos especialistas para a definição dos termos que representam as classes.

Os autores disponibilizaram uma interface *web* para os epsecialistas, para que eles classificassem os quatro tipos de texto (questionamentos legislativos, discursos em conferências, proposicões legislativas e proposições provisórias). Esses textos classificados serviram como dados de treinamento e teste do modelo *Support Vector Machine*, adotado para fazer a classificação.

Para representar os dados da análise, foram utilizados gráficos de radar para cada aspecto mencionado anteriormente, como mostram as figuras:

Figura 9 – Gráficos de desempenho do parlamentar

4.2 Planejamento das Atividades

Para a realização desse trabalho, foram identificadas algumas atividades relacionadas ao processamento de dados que seguem um fluxo de trabalho previsível, como mostra o diagrama:

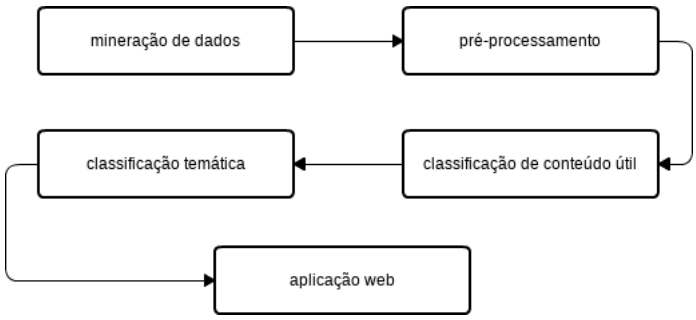


Figura 10 – Diagrama de planejamento

As seções seguintes descrevem cada etapa apresentada acima.

4.2.1 Mineração de Dados

A obtenção e persistência dos dados será realizada por uma biblioteca desenvolvida pelo autor, descrita na seção 5.1 e consiste em realizar consultas ao *webservice* da Câmara dos Deputados, fazendo um processamento inicial, afim de padronizar as informações e transformá-las em seus tipos correspondentes na linguagem *Python*. Os dados provenientes do *webservice* estão em formato *XML* e armazenam valores como *strings*.

Nesta etapa transformamos todos os campos com valores de números inteiros e decimais, datas, horários e textos em, respectivamente, valores *Python* correspondentes. Além disso, também é realizada a persistência dessas informações em banco de dados relacional.

4.2.2 Pré-processamento

Apesar do "pré-processamento" não depender tanto dos dados reais, a definição das *stop words* deve ser feita levando em consideração o conteúdo que será analisado.

A etapa de pré-processamento consiste na análise dos textos obtidos na mineração de dados, afim de identificar as *stop words* presentes em textos do contexto legislativo. Além disso, deve ser realizada o processo de *stemização* para reduzir a dimensionalidade das *bag-of-words* geradas.

Estudamos diferentes estratégias para a utilização de n-gramas, mas por enquanto decidiu-se, por uma questão de simplicidade, limitar-se apenas ao uso de unigramas.

4.2.3 Classificação Conteúdo Útil

Essa classificação tem como objetivo melhorar a qualidade da análise temática dos deputados. Uma parte considerável dos textos não possuem valor semântico significativo, pois tratam de questões protocolares e de trâmite legislativo. Citamos um exemplo: *"É preciso haver quórum de 257 Srs. Deputados para aprovação da matéria, quórum mínimo. A votação é normal. Então, acho que, quando houver uns 300 ou 320 votos, encerraremos."*. Portanto, a classificação entre conteúdo útil/não-útil consiste em separar os parágrafos que realmente possuem valor para a análise posterior dos que não devem ser usados nestas análises.

Essa atividade não foi mais adotada para a segunda parte desse trabalho, pois optou-se pela não utilização dos textos completos dos discursos, substituindo-os pelos sumários e indexações. Com essa nova abordagem, a análise é feita utilizando um resumo (sumário) do discurso, onde cada frase, geralmente, representa um tema abordado pelo parlamentar. Também foi utilizado um conjunto de palavras que são utilizadas para a busca no próprio site da Câmara dos Deputados. Tanto os sumários quanto as indexação são elaborados por um departamento da Casa, por profissionais especializados em Arquitetura da Informação.

4.2.4 Classificação Temática

Após determinar quais parágrafos serão análisados, os mesmos devem ser classificados de acordo com alguns temas. Para o contexto da primeira parte desse trabalho, foram selecionados inicialmente: Agropecuária, Saúde, Esporte, Educação, Ciência e Tecnologia, Economia, Política, Meio Ambiente, Direitos Humanos e Segurança. Para a realização dessa tarefa, foi necessário construir um texto inicial para cada um dos temas listados, afim de fornecer um parâmetro inicial ao classificador. Os textos iniciais de cada tema têm como base textos previamente classificados em portais de notícias brasileiros e consistem em apenas uma listagem de palavras comuns relacionadas a estes temas.

Para a segunda parte do trabalho, a base inicial de palavras foi obtida através do Tesauro da Câmara dos Deputados, que possui uma base de 14611 termos, agrupados em 49 áreas temáticas. Tal agrupamento foi realizado por profissionais da área de arquitetura da informação da própria Câmara.

54 temas é uma quantidade relativamente grande, o que poderia dificultar um pouco a análise. Por isso, foi realizado um agrupamento dessas áreas temáticas e reduzidos para 22 temas. As tabelas a seguir mostram todos os temas e seus agrupamentos (macrotemas):

Quantidade de Termos	**Tema**	**Macro-tema**
57	Administração	Gestão
1481	Administração Pública	Administração Pública
304	Agricultura, Pecuária e Pesca	Agricultura, Pecuária e Pesca
78	Política Fundiária	
97	Ciência da Informação	Artes, Cultura e Informação
221	Arte e Cultura	
17	Artes e Letras	
180	Ciência e Tecnologia	Ciência e Tecnologia
192	Informática e TI	
264	Comunicações	
45	Comércio Exterior	Relações Exteriores
132	Relações Internacionais	
36	Comunicação Social	Comunicação Social
139	Economia	Economia e Finanças Públicas
45	Contabilidade	
281	Finanças Públicas e Orçamento	
38	Política Econômica	
282	Sistema Financeiro	
235	Tributação	
27	Desenvolvimento Regional	Desenvolvimento Regional
61	Arquitetura e Urbanismo	Cidades
166	Desenvolvimento Urbano	
285	Desporto e Lazer	Esporte e Lazer
125	Turismo	
1033	Direito Civil e Processual Civil	Justica
134	Direito e Justiça	
1756	Direito Constitucional	Direito Constitucional
115	Direito e Defesa do Consumidor	Comércio e Consumidor
477	Indústria e Comércio	

Tabela 4 – Relação de Temas - Tesauro da Câmara dos Deputados (Parte I)

Quantidade de Termos	Tema	Macro-tema
150	Defesa e Segurança Nacional	Segurança
874	Direito Penal	
264	Segurança Pública	
1054	Direito do Trabalho	Trabalho
210	Trabalho e Emprego	
667	Educação	Educação
318	Meio Ambiente e Desenvolvimento Sustentável	Meio Ambiente e Energia
263	Recursos Hídricos, Minerais e Política Energética	
19	Ciência Política	Política
1610	Processo Legislativo	
552	Organização Política	
208	Direitos Humanos e Minorias	Direitos Humanos e Minorias
21	Antropologia	
45	Teologia	
9	Demografia	
120	Previdência e Assistência Social	Assistência Social
25	Serviço Social	
87	Sociologia	
938	Saúde	Saúde
660	Viação e Transporte	Viação e Transporte

Tabela 5 – Relação de Temas - Tesauro da Câmara dos Deputados (Parte II)

4.2.5 Aplicação *Web*

Os dados obtidos da classificação temática serão utilizados para alimentar um sistema *web* para a exibição dos mesmos. As principais funcionalidades planejadas são: visualizar os temas mais abordados por deputados, organizados por região, partido e bancada, visualizar todos os temas de uma determinada categoria, bem como o quanto cada tema é discutido e visualizar todos os temas abordados por um determinado deputado, mostrando separadamente os temas abordados em seus discursos e nas suas proposições. Os protótipos das telas do sistema estão disponíveis no apêndice B.

A primeira versão do Tenho Dito está disponível sub o domínio do Laboratório Hacker[5] da Câmara dos Deputados e conta com apenas uma das funcionalidades prevista nos protótipos no apêndice B. Após a entrega final desse trabalho, a ferramenta continuará sendo desenvolvida pela equipe de desenvolvedores do Laboratório Hacker, onde as outras funcionalidades previstas serão implementadas, bem como novas funcionalidades. Por enquanto, apenas a visualização por estados foi implementada.

[5] http://tenhodito.labhackercd.net

4.3 Ferramentas e Tecnologias

4.3.1 Linguagem de Programação

Devido ao tamanho da comunidade, grande utilização na área de aprendizado de máquina, possibilidade de desenvolvimento *web* e conhecimento prévio do autor e orientador, decidiu-se utilizar a linguagem *Python*[6] para o desenvolvimento das aplicações do presente trabalho.

4.3.2 *Frameworks* e Bibliotecas

O desenvolvimento da aplicação *web*, utilizará o *framework Django*[7], o que implica no uso da arquitetura *MVT* (*Model View Template*). Similar ao *MVC*, no *MVT* o ciclo começa por uma ação do usuário, a View notifica a Model, para que seu estado seja atualizado, a Model efetua as modificações necessárias e alerta as suas dependências que foi alterada, assim a Template consulta o novo estado da Model, e atualiza a sua visualização.

Além disso, utilizou-se a biblioteca *Javascript* D3.js[8] para auxiliar na visualização de dados. Por possuir características que podem facilitar o desenvolvimento, como *default parameters*, *arrow functions* e Classes, o código *Javascript* desse trabalho será escrito utilizando a versão *ES6*[9] (ou *ECMAScript 6*) e para garantir melhor suporte aos navegadores mais antigos, será utilizado também o *Babel*[10], um *transpiler* que transforma o código *ES 6* em código *ES 5*, suportado pela maioria dos navegadores atuais. Os estilos serão todos escritos utilizando a sintaxe *SCSS* e as ferramentas *node-sass* e *postcss* para transformar o código *SCSS* em *CSS*, além de adicionar estilos de suporte *cross-browser* automaticamente.

Esse trabalho não está focado na implementação de algoritmos de processamento de linguagem natural e aprendizado de máquina, mas sim na integração de algoritmos implementados por bibliotecas de terceiros. Serão utilizadas as bibliotecas *plagiarism*[11], *gensim*[12] e *texblob*[13], que encapsula a biblioteca *NLTK*[14], para tarefas de processamento de linguagem natural e aprendizado de máquina. Alguns cálculos são implementados utilizando o *stack* científico do *Python*, que inclui o *numpy*[15], *scipy*[16], *matplotlib*[17] e *sklearn*[18].

6 https://www.python.org
7 https://www.djangoproject.com
8 https://d3js.org/
9 http://es6-features.org/
10 https://babeljs.io/
11 https://github.com/fabiommendes/plagiarism
12 https://radimrehurek.com/gensim/
13 https://textblob.readthedocs.io
14 http://www.nltk.org
15 http://www.numpy.org
16 https://www.scipy.org
17 http://matplotlib.org
18 http://scikit-learn.org

4.3.3 Gerenciador de Repositórios de Código

O gerenciamento de versões dos códigos das aplicações desenvolvidas utiliza o *Git*[19] e o serviço de *web hosting* compartilhado *GitHub*[20]. Além disso, os pacotes *Python* são enviados para o *PyPI*[21], sendo facilmente instaláveis por terceiros através do comando *pip*[22].

4.3.4 Gerenciamento de Tarefas

O controle de tarefas executadas ou em execução é feito utilizando o sistema de *issues* e quadro de projetos do *GitHub*.

[19] https://git-scm.com
[20] https://github.com
[21] https://pypi.python.org/pypi
[22] https://pip.pypa.io

5 Resultados Obtidos

5.1 Obtenção dos Dados

Todos os dados utilizados para análise foram obtidos através do portal de dados abertos da Câmara dos Deputados[1], que é dividido em duas partes: dados legislativos e dados referentes à cota parlamentar, que não foram utilizados nesse trabalho. Os dados legislativos estão relacionados à informações sobre deputados, órgãos legislativos, proposições, sessões plenárias e reuniões de comissões.

5.1.1 *Webservice* da Câmara dos Deputados

Atualmente, o *Webservice* da Câmara dos Deputados é estruturado de acordo com os padrões *SOAP* (*Simple Object Access Protocol*, em português Protocolo Simples de Acesso a Objetos), que se baseiam na linguagem de marcação *XML* e utilizam, principalmente, chamada de procedimento remoto (*RPC*) e protocolo de transferência de hipertexto (*HTTP*) para a transmissão das mensagens (GUDGIN et al., 2007).

No entanto, o *webservice* possui alguns aspectos que podem ser melhorados. Como os dados são fornecidos utilizando o formato *XML*, eles não são "tipados", ou seja, independente do tipo (inteiro, data, texto, etc) eles são representados com *strings*. Alguns dados são ambíguos, como os referentes aos deputados, onde existem "ideCadastro" e "idParlamentar", que são utilizados como parâmetros de entrada de requisições distintas. Outro problema é que requisições comuns precisam ser feitas indiretamente pois não agrega conteúdos com *queries* relacionais, como normalmente são as *API REST*. Além disso, o inteiro teor dos discursos parlamentares estão disponíveis apenas em formato *RTF*, o que dificulta um pouco a utilização dos mesmos.

No momento de escrita desse trabalho, a nova *API* de dados abertos ainda se encontra em desenvolvimento e já pode ser acessada pela sociedade[2], porém ainda não possui os mesmos dados disponíveis na versão anterior. Dentre os dados que faltam estão os de discursos em plenário, o que torna o uso dessa nova plataforma desnecessário, já que este é o principal dado utilizado nesse trabalho.

O novo *webservice* segue os padrões REST e possibilita a escolha do formato de retorno, podendo ser em *XML* ou em *JSON*. Além disso, visa corrigir os problemas encontrados na versão anterior (alguns mencionados nos parágrafos anteriores), bem como aumentar a quantidade de dados disponíveis. Uma das promessas é disponibilizar o texto

[1] http://www.camara.leg.br/transparencia/dados-abertos
[2] https://dadosabertos.camara.leg.br/

completo das proposições, que hoje só é disponível via *PDF* sendo que alguns são apenas imagens *scaneadas* dos documentos físicos.

A estrutura do *webservice* da Câmara dos Deputados pode ser encontrada no apêndice C

5.1.2 Dados Utilizados

Para a primeira versão desse trabalho, foram utilizados apenas dados do ano de 2016, tanto para discursos quanto proposições. Foram escolhidos os discursos proferidos no Pequeno Expediente, que é a primeira parte da sessão ordinária do Plenário, tem duração máxima de 60 minutos, é destinado às comunicações de deputados previamente incritos e cada deputado pode discursar por, no máximo, 5 minutos. Já as proposições utilizadas foram apenas Projetos de Lei.

5.2 Proposta de Desenvolvimento

5.2.1 *Pygov-br*

Pygov-br é uma biblioteca *python* desenvolvida no contexto desse trabalho cujo objetivo é centralizar o consumo de *APIs* e *webservices* governamentais brasileiros. Além dos dados, a biblioteca também irá fornecer um conjunto de *plugins* para os principais *frameworks* para desenvovimento *web*, para facilitar a utilização dos dados abertos, bem como o cruzamento de dados provenientes de diferentes órgãos governamentais.

Atualmente, a biblioteca oferece suporte somente ao *webservice* da Câmara dos Deputados, tanto para o consumo dos dados quanto para o uso em aplicações *Django*, já que para o desenvolvimento do presente trabalho apenas esses dados seriam utilizados.

A estrutura para o consumo dos *webservices* não é fixa, pois cada *webservice* possui suas características. A implementação para consumir os dados da Câmara dos Deputados segue a estrutura do *webservice* (figura 11), com algumas alterações. Além disso, todos o código implementado foi escrito em inglês, apesar dos dados estarem em português.

Como dito anteriormente, a *pygov-br* também possui módulos para utilização em conjunto com os *frameworks* de desenvolvimento *web* mais utilizados na comunidade. Porém, como a solução *web* desenvolvida nesse trabalho utilizará o *framework Django*, a atual implementação da *pygov-br* possui suporte somente a esse *framework*.

O módulo **django_apps** contém os *plugins* para utilização em projetos *Django*. Esses *apps* possuem apenas as *models* (na linguagem da arquitetura *MVT* do *Django*) já que o objetivo é somente facilitar a permanência das informações obtidas dos *webservices* governamentais em um banco de dados. No caso da Câmara dos Deputados, os dados

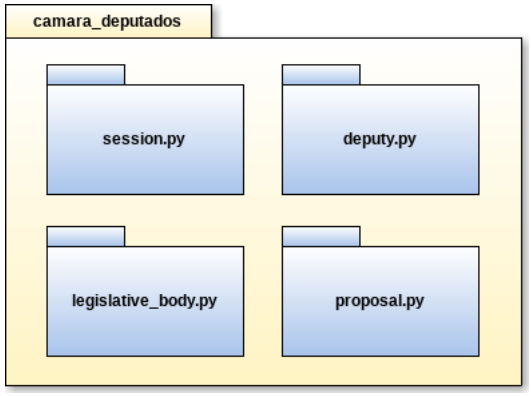


Figura 11 – Estrutura do módulo de consumo de dados da Câmara dos Deputados

utilizados nesse trabalho ficam disponíveis seguindo o modelo entidade-relacionamento na figura 12. Podemos notar que todas as colunas de todas as tabelas se encontram em em inglês, por motivos de padronização do código.

Após a apresentação da primeira parte desse trabalho, foram realizadas algumas sugestões quanto à tradução dos termos para o inglês. Entretanto, como estava prevista uma nova API da Câmara dos Deputados com alterações significativas que implicariam em uma reescrita considerável do código da *pygov-br*, ficou decidido que essas alterações de nomenclaturas seriam realizadas no momento de reescrita da biblioteca.

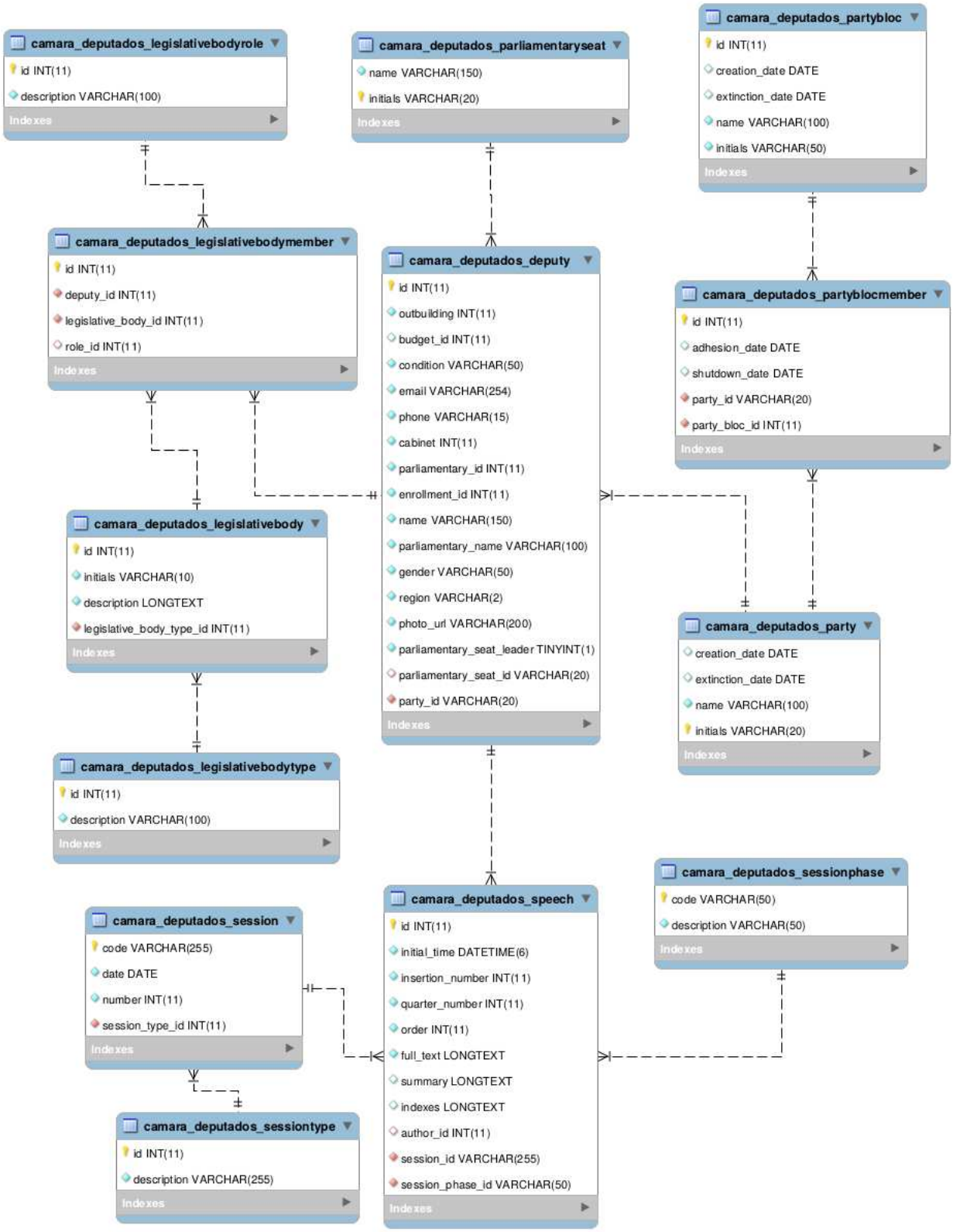


Figura 12 – Modelo entidade-relacionamento do banco de dados utilizado

5.2.2 Tenho Dito

"Tenho Dito" corresponde ao produto deste trabalho visível pelo usuário final. Trata-se de uma aplicação *web*, desenvolvida utilizando a linguagem *python* com o *framework Django*, e tem como objetivo ser uma forma mais lúdica e amigável de visualização de alguns dados disponíveis nos *webservices* de dados abertos da Câmara dos Deputados. Utiliza métodos de processamento de linguagem natural e aprendizado de máquina para extrair o perfil temático dos parlamentares, analisando o texto de seus discursos e proposições. Além disso, também é possível traçar os temas mais discutidos (tanto em propostas quanto nos próprios discursos) pelos deputados de uma determinada região ou por partidos.

A aplicação é dividida em dois grandes módulos: *nlp* e *application*. O primeiro é responsável por todas as operações relacionadas ao processamento dos textos, o que inclui aprendizado de máquina. Já o segundo módulo é responsável pela parte *web*. No momento de escrita desse trabalho, foi implementado apenas a principal funcionalidade do produto: a análise por estado. E estão disponíveis alguns protótipos de tela, mostrando as possíveis funcionalidades do sistema.

Conforme mencionado no item "*frameworks*", em 4.3, a análise dos textos será realizada com o apoio das ferramentas:

- **Plagiarism:** biblioteca em estado *alpha* desenvolvida pelo professor orientador do autor desse trabalho, Fábio Macêdo Mendes, e possui uma série de funcionalidades utilizadas no pré-processamento dos textos, como extração de *tokens*, *stemização*, remoção de *stop words*, geração de *n-gramas* e geração de *bag-of-words* (com os diferentes tipos de representação dos termos, descrito na seção 2.1.1 desse trabalho). Apesar do foco ser a detecção de plágio em textos e códigos, as funcionalidades implementadas podem ser utilizadas em tarefas genéricas de PLN.
- **Textblob:** biblioteca *python* para processamento de dados textuais. Ela fornece uma interface simples para realizar tarefas comuns de processamento de linguagem natural, como análise de sentimento e classificação, por exemplo. Utiliza a biblioteca *NLTK*[3] para realizar essas tarefas.

Com o objetivo de deixar registrado as experiências obtidas na primeira parte desse trabalho, dividimos essa seção em duas partes.

5.2.2.1 Primeira Parte do Trabalho

Na primeira parte desse trabalho, a classificação dos discursos e proposições era dividida em duas etapas. A primeira etapa consistia em, inicialmente, dividir o texto em

[3] http://www.nltk.org

parágrafos, para que a análise fosse realizada com uma quantidade menor de texto, e em seguida os parágrafos seriam classificados entre "conteúdo útil" ou "conteúdo não-útil". Por exemplo, o trecho *"Muito obrigado, nobre Deputado. Pelo PSOL de São Paulo, o nobre Líder Ivan Valente. V. Exª tem cinco minutos na tribuna."* não representa um conteúdo significativo, da mesma forma que *''O SR. ALCEU MOREIRA - Sr. Presidente, primeiro a medida provisória, logicamente."* também não agregaria nenhum valor à análise. Trechos como esses deveriam ser classificados como "conteúdo não-útil" e descartados da análise temática. A segunda etapa do processamento era a classificação temática dos parágrafos classificados como "conteúdo útil", na etapa anterior.

Para ambas etapas o procedimento adotado era o mesmo, com algumas alterações nos classificadores. Primeiro, um classificador *NaiveBayesClassifier*, implementado pela biblioteca *textblob*, era instanciado, utilizando dois conjuntos de palavras iniciais, um para definir "conteúdo não-útil" e outro para "conteúdo". Os conjuntos de palavras podem ser encontrados no apêndice D.

Em seguida, todos os parágrafos eram classificados e, dentre os que foram classificados com uma probabilidade maior que 80%, os 100 melhores colocados eram utilizados para realizar o treinamento inicial do classificador. A partir disso, era realizado um treinamento supervisionado, onde a aplicação sugeria uma classe mais provável e um especialista humano dizia se o trecho correspondia à classe sugerida, caso não fosse ele deveria fornecer a classe correta. Ao finalizar o treinamento supervisionado, todos os parágrafos eram classificados novamente, agora com o classificador melhor treinado.

Vale observar que as classificações útil/não-útil sugeridas após a primeira fase normalmente correspondiam às corretas, ainda que isto não tenha sido medido explicitamente.

Com o resultado a primeira classificação, obtinha-se um conjunto de parágrafos classificados como "conteúdo", que seriam usados na classificação temática. De forma semelhante à primeira classificação, um classificador *NaiveBayesClassifier* era instanciado, agora com um conjunto de palavras específico para cada tema. Os temas e seus respectivos conjuntos de palavras também se encontram no apêndice D.

Todos os parágrafos eram classificados novamente e era gerado um conjunto com os melhores classificados, que era usado para realizar o treinamento inicial do classificador. Após esta etapa, acontecia o treinamento supervisionado, onde um especialista dizia se a classificação sugerida fazia sentido e indicava a classe correta quando não fosse.

Também era possível realizar um treinamento não supervisionado para ambos os classificadores, de forma que as sugestões de classificação eram utilizadas para o treinamento sem a análise de um especialista.

A cada iteração da fase de treinamento todas as probabilidades dos textos adicio-

nados ao classificador são recalculadas, o que implica no aumento significativo do tempo de processamento. Por isso, foi utilizada uma ferramenta de *cache*, possibilitando o armazenamento dos classificadores depois de cada atualização. Toda vez que for realizado um treinamento ou uma classificação seria utilizado o último classificador armazenado em *cache*, com o treinamento prévio.

Esta fase mostrou que alguns discursos são difíceis de classificar ou por terem um conteúdo fragmentado (ex.: parágrafo com apenas uma ou poucas palavras, como *"-Rio de Janeiro"*) ou por conter um conteúdo que aborda mais de um tema simultaneamente (ex.: *"Eu parei de jogar há quase 17 anos e há 15 anos eu criei o Instituto Esporte & Educação - IEE, do qual sou Presidente, que trabalha com esporte e educação. E há 15 anos nós viajamos para cidades do Brasil que não têm acesso à prática motora na escola, que não têm estrutura, que não têm professores e, especialmente, que não têm a visão da educação física, do esporte, do movimento, da ação motora, da atividade motora como um fator de desenvolvimento, cuja presença é importante dentro da escola."*).

5.2.2.2 Segunda Parte do Trabalho

Tendo em vista que o resultado das classificações utilizando o modelo descrito acima não foi satisfatório, algumas alterações foram feitas. Tais alterações serão organizadas por tópicos.

5.2.2.2.1 Dados Utilizados na Análise

Em primeiro lugar, passamos a utilizar uma base de palavras pré-classificadas maior: o *thesaurus* da Câmara dos Deputados, que estará deisponível no repositório do Tenho Dito. Esse conjunto de palavras foi utilizado para treinamento do classificador, o que melhorou, mas não o suficiente, o resultado da classificação temática.

Também deixamos de utilizar o texto completo para a análise e passamos a utilizar apenas o sumário, que é basicamente um resumo do discurso/proposição estruturado de forma simples, onde cada sentença corresponde a um tópico abordado pelo deputado (geralmente). Temos, por exemplo, o sumário de um discurso proferido pelo deputado Alberto Fraga, no dia 10/07/2017: *"Incompetência administrativa do Governador Rodrigo Rollemberg. Protesto contra a derrubada de construções na orla do Lago Paranoá, determinada pelo Governo de Brasília. Repúdio à ação ajuizada contra o orador, em face de críticas à administração do Distrito Federal."*. Analisando o sumário, podemos ver que *"Incompetência administrativa do Governador Rodrigo Rollemberg"* aborda um tema, *"Protesto contra a derrubada de construções na orla do Lago Paranoá, determinada pelo Governo de Brasília"* aborda outro e *"Repúdio à ação ajuizada contra o orador, em face de críticas à administração do Distrito Federal."* outro. Essa característica permitiu o descarte

do primeiro classificador (de "contúdo útil/não-útil") e melhorar um pouco a classificação temática, já que as sentenças obtidas trazem um conteúdo relevante, são menores e abrangem menos temas.

Entre os dados de proposições disponíveis no *webservice* de dados abertos da Câmara dos Deputados não encontramos algo semelhante ao sumário dos discursos, apenas a ementa da proposição, mas que na maioria das vezes contém algo muito técnico, sendo necessário o conhecimento mais profundo das leis, como por exemplo: *"Revoga os artigos 165, 166 e 204 do Decreto-Lei nº 1.001, de 21 de outubro de 1969."*. Buscando utilizar os mesmos dados, tanto para dicursos quanto para proposições, optou-se pela utilização da indexação realizada pela Câmara dos Deputados, já que a mesma está presente em todos os dicursos realizados no Pequeno Expediente e em todos os Projetos de Lei. Somente para exemplificar, o Projeto de Lei que possui a ementa citada anteriormente, possui a seguinte indexação: *"Revogação, dispositivo legal, Decreto-Lei, Código Penal Militar, promoção, reunião, publicação, crítica, indevido, exercício, comércio, militar da ativa"*, o que permite o entendimento do que realmente a proposição aborda.

5.2.2.2.2 Modelos Utilizados

Na primeira parte desse trabalho, tentou-se utilizar uma abordagem não-supervisionada com o *k-means* e ao não obter um resultado satisfatório, optou-se pela utilização do *Naive Bayes*. Este, por sua vez, define apenas uma classe para cada texto analisado, mesmo quando o texto aborda mais de um tema. Com o objetivo de identificar mais de um tema por discurso, iniciou-se o estudo do modelo *LDA*.

O *LDA* é, naturalmente, não-supervisionado. Entretanto, a implementação da biblioteca *Gensim*[4] permite inserir informações a priori no algoritmo.

Para aplicar o *LDA*, instanciamos um objeto da classe *LdaModel* passando como parâmetros o *corpus* de treinamento utilizado (*thesaurus*), a quantidade de tópicos a serem encontrados e o *eta*, que será utilizado pelo algoritmo para definir probabilidades a priori. O *eta* é uma matriz $K \times n$, onde K é o número de tópicos e n o número de termos. Com essa matriz, podemos definir valores específicos para determinadas palavras em relação aos tópicos, sendo que o valor padrão para cada palavra é de $\frac{1}{K}$. Dessa forma, os termos já classificados pelo *thesaurus* foram inseridos no *LDA* através do *eta*, onde a valor dos termos em seus tópicos passam ser definidos de acordo com a seguinte equação:

$$\frac{1}{K} + \frac{(K-1)}{K} \cdot F, \tag{5.1}$$

onde K é a quantidade de tópicos e F é 0.99^6. O valor de F foi escolhido para ser o menor número que classifica todos os conhecidos corretamente, e a ideia do expoente foi somente

[4] https://radimrehurek.com/gensim/models/ldamodel.html#gensim.models.ldamodel.LdaModel

para ajustar esse peso. Dessa forma, o *training set* é classificado corretamente e o tópico correto sempre com mais de 50%.

Para visualizar os dados obtidos com o *LDA*, utilizamos o método *PCA* (*Principal Component Analysis*), que tenta capturar as componentes principais de um conjunto de dados e projeta em um espaço de dimensão mais baixa, para mostrar os dados em duas dimensões. O *LDA* funciona bem quando o *PCA* consegue dividir o *dataset* em *clusters* relativamente bem definidos. No gráfico abaixo, os pontos marcados correspondem a um *cluster* conhecido específico.

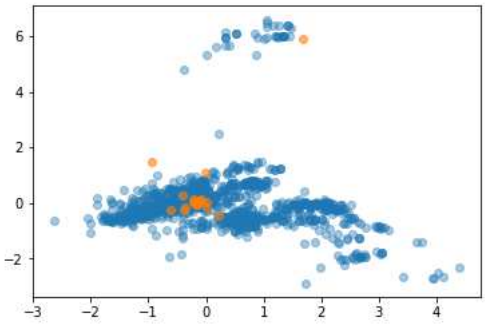


Figura 13 – Análise dos dados do *LDA* utilizando *PCA*

Os *clusters* podem estar em mais de duas dimensões, porém analisando os nossos dados, mesmo em duas dimensões, conseguimos visualizar apenas um *cluster* separado e todo o resto junto, mostrando que a classificação claramente não está correta.

Tendo em vista o resultado acima, voltamos a utilizar o modelo *naive Bayes* para realizar a classificação dos discursos e proposições.

5.2.2.2.3 Classificação dos Discursos e Proposições

Como dito nos tópicos anteriores, serão utilizados os dados do *thesaurus* da Câmara dos Deputados para o treinamento, a indexação dos discursos e proposições para e o classificador *naive Bayes* para a análise.

Primeiramente, assim como na primeira parte do trabalho, é instanciado um objeto da classe *NaiveBayesClassifier*, implementado pela biblioteca *textblob*, passando como treinamento inicial o conjunto de termos já classificados do *thesaurus*. Com o classificador treinado, passamos por cada discurso e em cada um deles obtemos a lista de termos usados em sua indexação. Cada termo é analisado individualmente e apenas aqueles que forem classificados com a probabilidade maior que 60% são utilizados, o restante é descartado. Após classificar todos os termos de indexação de um discurso, calculamos a porcentagem de cada tema e atribuímos ao discurso. Por exemplo, suponha um discurso possui 13 termos na indexação e apenas 8 deles foram classificados com probabilidade maior que 60%. Se 4 desses foram classificados como “Segurança”, 3 como “Direitos Humanos” e 1

como “Saúde”, então o discurso será classificado como sendo 50% sobre segurança, 37,5% direitos humanos e 12,5% saúde. Além disso, definimos como tema principal do discurso aquele com maior porcentagem.

Vale enfatizar que as porcentagens mostradas pelo *naive Bayes* são interpretadas de forma diferente que o *LDA*. Enquanto no segundo trata-se de uma mistura de temas, no *naive Bayes* correspondem à chance de um texto pertencer exclusivamente a cada tópico.

Para as proposições, os mesmos procedimentos são realizados, bem como a utilização do mesmo classificador já treinado.

5.2.2.2.4 Classificação dos Deputados

Uma vez que todos os discursos e proposições estejam classificados, passamos por todos os deputados e em cada um deles obtemos uma lista de discuros e uma lista de proposições. Somamos todas as porcentagens, de todos os temas de discursos e proposições e dividimos pela soma da quantidade de discursos e proposições. Por exemplo, suponha que um parlamentar possui dois discursos e uma proposição classificada. Seu primeiro discurso é 100% sobre “Direitos Humanos” e o segundo 70% “Saúde” e 30% “Direitos Humanos”. A proposição foi classificada como 60% “Segurança” e 40% “Educação”. Temos então que a classificação final do deputado seria: 43,3% “Direitos Humanos”, 23,3% “Saúde”, 20% “Segurança” e 13,3% “Educação”. Assim como nos discursos e proposições, o tema com maior porcentagem é o tema principal de um deputado.

5.2.2.2.5 Classificação dos Estados

Após a classificação dos deputados, passamos por todos os estados e em cada um obtemos a lista de parlamentares que o representam. Utilizando o mesmo procedimento da classificação dos deputados, somamos todas as porcentagens temáticas dos deputados e dividimos pela quantidade total de deputados.

5.2.2.2.6 Acurácia

Após a mudança dos dados utilizados na análise, pudemos notar uma melhoria significativa no resultado da classificação. Utilizando como treinamento 11.151 termos, presentes no *thesaurus*, e como teste 111 novos termos, obtidos da indexação dos próprios discursos e proposições, classificados manualmente pelo autor, obtivemos um índice de acerto de 59,46%. Justificando, assim, a escolha de classificações apenas com probabilidade maior que 60%, descrito nos itens anteriores.

6 Considerações Finais

Nesta seção iremos pontuar alguns resultados obtidos durante o desenvolvimento desse trabalho. Primeiramente, o autor estabeleceu um bom conhecimento sobre processamento de linguagem natural, incluindo técnicas de pré-processamento e aprendizado de máquina, o que possibilitou a aplicação desses conceitos no desenvolvimento das aplicações propostas nesse trabalho.

Para as contribuições tecnológicas, foi construída uma versão inicial da biblioteca *pygov-br*, que facilita o consumo do *webservice* da Câmara dos Deputados em aplicações *Python*, assim como a persistência dessas informações em banco dados, através da aplicação *Django* que faz parte da *pygov-br*.

Iniciou-se o desenvolvimento da aplicação *web*, com a implementação dos algoritmos de classificação temática e sua principal funcionalidade, que consiste em poder visualizar o perfil temático de um parlamentar. Entretanto, apenas a visualização dos dados organizados por estado foi implementado. As outras funcionalidades serão implementadas pela equipe de desenvolvimento do Laboratório Hacker da Câmara dos Deputados.

6.1 Perspectivas Futuras

A partir do resultado desse trabalho podemos aplicar outras formas de aprendizado de máquina sobre o perfil temático dos deputados. Uma opção é a utilização do *k-means* para determinar a proximidade dos parlamentares de acordo com os temas abordados por eles.

Além disso, várias funcionalidades e melhorias na experiência de usuário da aplicação *web* podem ser implementadas, como buscar diretamente um parlamentar pelo nome, outras formas de organização dos dados (por gênero, partido, tema, etc), melhorar a forma de navegação no sistema, entre outras.

A partir do momento que os dados disponíveis na nova *API* da Câmara dos Deputados contemplarem os dados necessários para o funcionamento da aplicação, pretende-se atualizar a biblioteca *pygov-br* para o novo modelo.

Referências

ANNALYN, N. Automated biography for a country - using computational methods to study historical trends. 6th Annual International Conference on Political Science, Sociology and International Relations, Singapore, 2016. Citado na página 36.

BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. University of California, Berkeley, 2003. Citado na página 34.

BRASIL. *Lei nº 12.527, de 18 de novembro de 2011.Regula o acesso à informações*. 2011. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2011/lei/l12527.htm>. Acesso em: 4 out 2016. Citado na página 19.

BRUCE, R. A bayesian approach to semi-supervised learning. In: *Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium — NLPRS-2001*. Tokyo, Japan: Springer Verlag, 2001. Citado na página 37.

CÂMARA DOS DEPUTADOS. *Dados Abertos da Câmara dos Deputados*. 2016. Disponível em: <http://www2.camara.leg.br/transparencia/dados-abertos/perguntas-e-respostas>. Acesso em: 4 out 2016. Citado na página 19.

COLE, R. M. Clustering with genetic algorithms. Department of Computer Science, University of Western Australia, Australia, 1998. Citado 2 vezes nas páginas 34 e 37.

DINIZ, V. *Como Conseguir Dados Governamentais Abertos*. Brasília, Brasil, 2010. Citado 2 vezes nas páginas 19 e 20.

ESTIVILL-CASTRO, V. Why so many clustering algorithms — a position paper. *ACM SIGKDD Explorations Newsletter*, 2002. Citado na página 33.

GIROLAMI, M.; KABAN, A. On an equivalence between plsi and lda. *Proceedings of SIGIR*, Association for Computing Machinery, 2003. Citado na página 34.

GRIMMER, J. A bayesian hierarchical topic model for political texts: Measuring expressed agendas in senate press releases. *Department of Government, Harvard University, 1737 Cambridge Street, Cambridge, MA 02138*, 2009. Citado na página 41.

GUDGIN, M. et al. *W3C SOAP Specifications*. 2007. Disponível em: <https://www.w3.org/TR/soap12/>. Citado na página 51.

HAND, D. J.; YU, K. Idiot's bayes — not so stupid after all? *International Statistical Review*, 2001. Citado na página 30.

IMAMURA, C. Y. Pré-processamento para extração de conhecimento de bases textuais. 2001. Citado na página 26.

JAIN, M. M. A.; FLYNN, P. Data clustering: A review. In: *ACM Computing Surveys*. [S.l.: s.n.], 1999. v. 31, p. 264–323. Citado na página 32.

JAYNES, E. T. *PROBABILITY THEORY THE LOGIC OF SCIENCE*. [S.l.]: Cambridge University Press, 2003. Citado na página 31.

JONES, K. S. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, v. 28, p. 11–21, 1972. Citado na página 24.

JORDAN, M. I.; CHRISTOPHER, M. Neural networks. In: *Computer Science Handbook*. Second edition. Boca Raton, FL: Chapman & HallCRC Press LLC, 2004. Citado na página 32.

KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *International joint Conference on artificial intelligence*. [S.l.: s.n.], 1995. Citado na página 38.

LIN SHIH-YAO CHOU, D. L. F.-R.; HAO, D. Automatic content analysis of legislative documents by text mining techniques. 2015 48th Hawaii International Conference on System Sciences, 2015. Citado na página 42.

LLOYD, S. P. Least square quantization in pcm. *IEEE Transactions on Information Theory*, p. 129–137, 1957. Citado na página 33.

LOPES, E. D. Utilização do modelo skip-gram para representação distribuída de palavras no projeto media cloud brasil. 2015. Citado na página 26.

LOVINS, J. B. Development of a stemming algorithm. In: *Mechanical Translation and Computational Linguistics*. [S.l.: s.n.], 1968. Citado na página 25.

MATSUBARA, E. T.; MONARD, M.-C. *PreTexT: uma ferramenta para pré-processamento de textos utilizando a abordagem bag-of-words*. 2003. Citado 3 vezes nas páginas 23, 24 e 25.

MAZONI, M. V. F. *Dados Abertos para a Democracia na Era Digital*. Brasília, Brasil, 2011. 80 p. Citado na página 19.

MITCHELL, T. M. *Machine Learning*. [S.l.]: McGraw Hill, 1997. Citado na página 29.

MOHRI, A. R. M.; TALWALKAR, A. Foundations of machine learning. *The MIT Press*, 2012. Citado na página 29.

OGURI, R. L. P.; RENTERIA, R. Aprendizado de máquina para o problema de sentiment classification. *Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro*, 2007. Citado na página 30.

OPEN KNOWLEDGE. About. 2016. Disponível em: <https://okfn.org/about/>. Acesso em: 4 out 2016. Citado na página 19.

PARDO, T. A. S.; NUNES, M. das G. V. Aprendizado bayesiano aplicado ao processamento de línguas naturais. *Série de Relatórios do Núcleo Interinstitucional de Linguística Computacional NILC - ICMC-USP*, 2002. Citado na página 30.

PELLUCCI, P. R. S. et al. Utilização de técnicas de aprendizado de máquina no reconhecimento de entidades nomeadas no português. *Centro Universitário de Belo Horizonte, Belo Horizonte, MG*, 2011. Citado na página 30.

PORTER, M. F. An algorithm for suffix stripping. *Program electronic library and information systems*, 1980. Citado na página 26.

PRITCHARD, M. S. J. K.; DONNELLY, P. Inference of population structure using multilocus genotype data. Genetics Society of America, 2000. Citado na página 34.

RAJARAMAN, A.; ULLMAN, J. D. Data mining. In: *Mining of Massive Datasets*. [S.l.: s.n.], 2011. Citado na página 26.

REZENDE, S. O. *Sistemas inteligentes: fundamentos e aplicações*. [S.l.]: Ed. Barueri, SP, 2003. Citado na página 29.

SALTON, G.; BUCKLEY, C. Term-weighting approaches in automatic text retrieval. 1988. Citado na página 24.

Apêndices

APÊNDICE A – Links Importantes

- Aplicação Tenho Dito: <http://tenhodito.labhackercd.net>, 47
- Organização do Tenho Dito no *GitHub*: <https://github.com/tenhodito>, 21
- Repositório do Tenho Dito: <https://github.com/tenhodito/tenhodito>, 21
- Texto atualizado: <https://github.com/msfernandes/tcc>, 21

Biblioteca D3.js: <https://d3js.org/>, 48

Framework Django: <https://www.djangoproject.com>, 48

Gensim: <https://radimrehurek.com/gensim/models/ldamodel.html#gensim.models.ldamodel.LdaModel>, 58

NumPy: <http://www.numpy.org>, 48

Plagiarism: <https://github.com/fabiommendes/plagiarism>, 48

Python: <https://www.python.org>, 48

Texblob: <https://textblob.readthedocs.io>, 48

APÊNDICE B – Protótipos iniciais do Tenho Dito

Foram desenvolvidos alguns protótipos de telas do sistema Tenho Dito, a ser desenvolvido nesse trabalho. Na tela inicial (figura 14) será exibido um mapa político do Brasil e ao passar o *mouse* pelos estados, o tema mais abordado pelos parlamentares que representam o estado é mostrado. Também existe a possibilidade de alterar a forma de visualização, além de ter uma abordagem por estado, o usuário pode escolher por partido ou por tema. Entretanto, a abordagem por tema não foi prototipada.

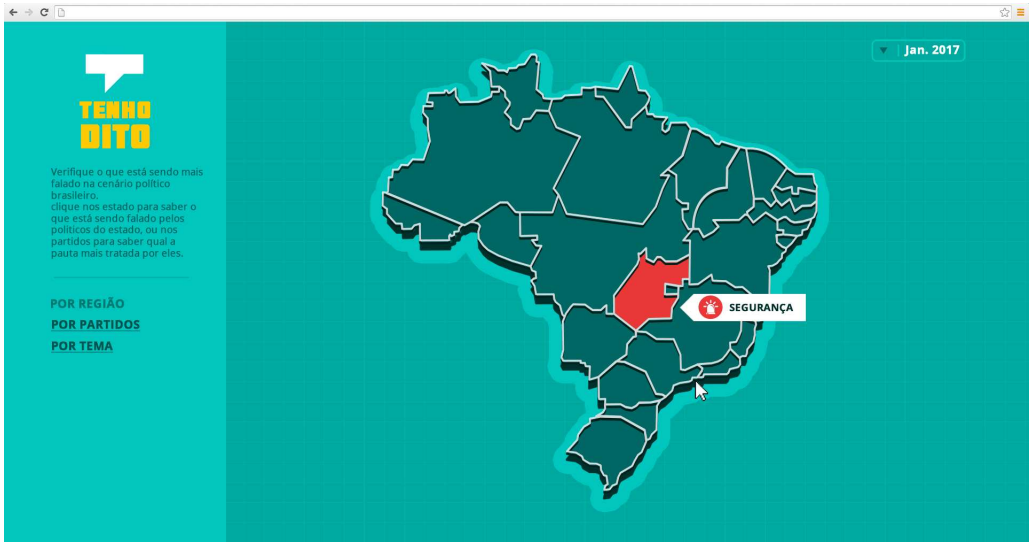


Figura 14 – Tela inicial do Tenho Dito - Visualização por região

Seguindo a abordagem por temas, ao clicar em um estado, o usuário é direcionado a outra página (figura 15), onde encontra um gráfico de bolhas, detalhando os temas abordados pelos parlamentares do estado. Quanto maior a bolha, mais o tema foi abordado. Além disso, também são listados todos os deputados que representam aquele estado, juntamente com sua foto, partido e o tema predominante em seus discursos e proposições. Ainda não foram definidas as interações com o gráfico de bolhas.

O usuário também poderá selecionar um deputado específico e visualizar o seu perfil. Na tela de perfil do deputado (figura 16), são exibidas as informações do deputado e também a quantidade de proposições e discursos analisados. Logo abaixo, será mostrado, dinâmica e randomicamente, trechos de discursos ou proposições e sua classificação. Além disso, serão listados todos os temas e a quantidade de discursos e proposições (por meio de gráfico de barras), com o objetivo de realizar uma comparação entre o que é mais dito pelo deputado e o que é mais proposto.

Figura 15 – Visualização detalhada dos temas, por região

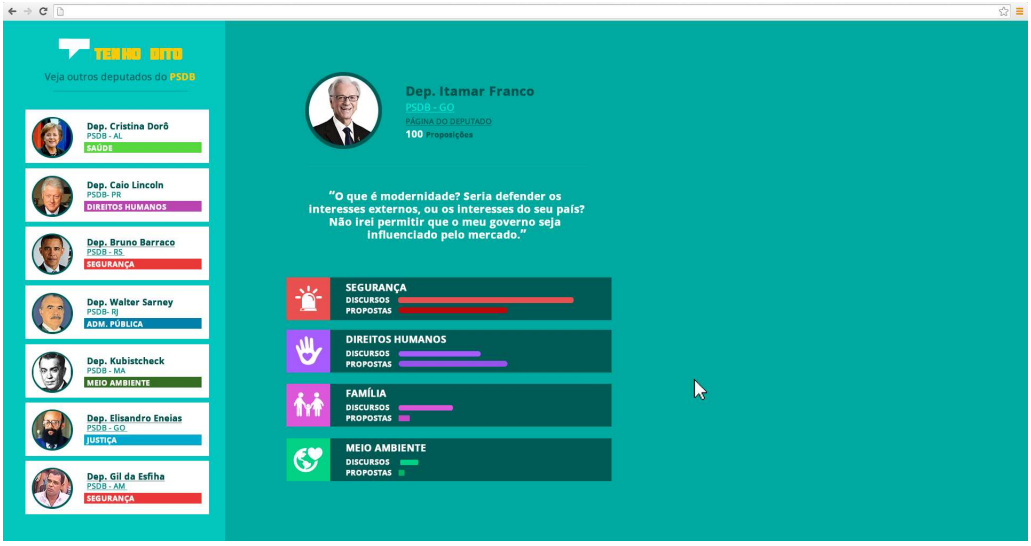


Figura 16 – Página de perfil do deputado

Quando o usuário clicar na opção de visualização por partidos, será exibida uma lista com os atuais partidos com representação na Câmara dos Deputados (figura 17). O sitema possibilitará três tipos de ordenação: por tamanho (quantidade de deputados por partido), por ordem alfabética ou por tema. Nessa tela, também serão exibidos os temas mais abordados pelos partidos, através dos seus membros. Caso o partido tenho mais deputados cujo tema mais abordado em seus discursos e proposições é "segurança", por exemplo, o tema atribuído ao partido será "segurança".

O usuário poderá, assim como na abordagem por estado, escolher um partido para detalhar os temas abordados e, da mesma forma, é exibido um gráfico de bolhas com os temas abordados pelos deputados desse partido. Ao lado são mostrados todos os deputados do partido, independente do seu estado, ao clicar em algum deles o usuário é direcionado à pagina de perfil dele.

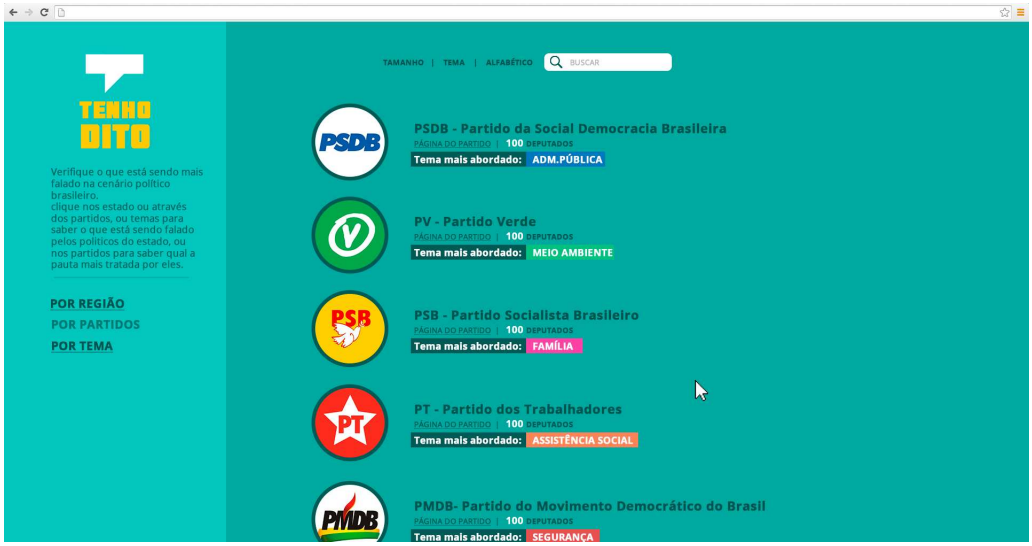


Figura 17 – Listagem dos partidos com representação na Câmara dos Deputados

Figura 18 – Visualização detalhada dos temas, por partido

APÊNDICE C – *Webservice* da Câmara dos Deputados

O *webservice* atual (*SOAP*) possui um total de 28 *endpoints*, onde 5 são relacionados aos deputados, 9 aos órgãos, 9 às proposições e 5 às sessões e reuniões. A seguir descrevemos os *endpoints* utilizados.

Os *endpoints* que fornecem dados de deputados são:

- **ObterDeputados:** retorna os deputados em exercício na Câmara dos Deputados
- **ObterDetalhesDeputado:** retorna detalhes dos deputados com histórico de participação em comissões, períodos de exercício, filiações partidárias e lideranças.
- **ObterLideresBancadas:** retorna os deputados líderes e vice-líderes em exercício das bancadas dos partidos
- **ObterPartidosCD:** retorna os partidos com representação na Câmara dos Deputados
- **ObterPartidosBlocoCD:** retorna os blocos parlamentares na Câmara dos Deputados.

Os *endpoints* que fornecem dados de órgãos legislativos são:

- **ListarCargosOrgaosLegislativosCD:** retorna a lista dos tipos de cargo para os órgãos legislativos da Câmara dos Deputados (ex: presidente, primeiro-secretário, etc)
- **ListarTiposOrgaos:** retorna a lista dos tipos de órgãos que participam do processo legislativo na Câmara dos Deputados
- **ObterAndamento:** retorna o andamento de uma proposição pelos órgãos internos da Câmara a partir de uma data específica
- **ObterEmendasSubstitutivoRedacaoFinal:** retorna as emendas, substitutivos e redações finais de uma determinada proposição
- **ObterIntegraComissoesRelator:** retorna os dados de relatores e pareces, e o link para a íntegra de uma determinada proposição

- **ObterMembrosOrgao:** retorna os parlamentares membros de uma determinada comissão
- **ObterOrgaos:** retorna a lista de órgãos legislativos da Câmara dos Deputados (comissões, Mesa Diretora, conselhos, etc.)
- **ObterPauta:** retorna as pautas das reuniões de comissões e das sessões plenárias realizadas em um determinado período
- **ObterRegimeTramitacaoDespacho:** retorna os dados do último despacho da proposição

Os *endpoints* que fornecem dados de proposições são:

- **ListarProposicoes:** retorna a lista de proposições que satisfaçam os critérios estabelecidos
- **ListarSiglasTipoProposicao:** retorna a lista de siglas de proposições
- **ListarSituacoesProposicao:** retorna a lista de situações para proposições
- **ListarTiposAutores:** retorna a lista de tipos de autores das proposições
- **ObterProposicao:** retorna os dados de uma determinada proposição a partir do tipo, número e ano
- **ObterProposicaoPorID:** retorna os dados de uma determinada proposição a partir do seu ID
- **ObterVotacaoProposicao:** retorna os votos dos deputados a uma determinada proposição em votações ocorridas no Plenário da Câmara dos Deputados
- **ListarProposicoesVotadasEmPlenario:** retorna todas as proposições votadas em plenário num determinado período
- **listarProposicoesTramitadasNoPeriodo:** retorna uma lista de proposições movimentadas em determinado período.

Os *endpoints* que fornecem dados de sessões e reuniões são:

- **ListarDiscursosPlenario:** retorna a lista dos deputados que proferiam discurso no Plenário da Cãmara dos Deputados em um determinado período.
- **ListarPresencasDia:** retorna a lista de presença de deputado em um determinado dia.

- **ListarPresencasParlamentar:** retorna as presenças de um deputado em um determinado período.
- **ListarSituacoesReuniaoSessao:** retorna a lista de situações para as reuniões de comissão e sessões plenárias da Câmara dos Deputados
- **ObterInteiroTeorDiscursosPlenario:** retorna o inteiro teor do discurso proferido no Plenário.

APÊNDICE D – Treinamento Inicial dos Classificadores

Para realizar o treinamento inicial dos classificadores *naive* Bayes, é necessário fornecer um texto inicial e a sua classificação. Esse apêndice descreve os textos usados nesse trabalho para cada classificação.

D.1 Classificação de Conteúdo Útil/Não-útil

Para a classificação de "conteúdo útil" e "conteúdo não-útil", foram utilizadas os seguintes conjuntos de palavras iniciais:

- **Conteúdo não-útil:** "agradecimento agradeço muito obrigado v.exa. digníssimo nobre deputado amigo peço registro pela ordem pedir um aparte mérito emendas votado sessão comissão protocolo regimento pronunciamento divulgação"

- **Conteúdo útil:** "educação universidade estudante professor ensino escola educador saúde médicos hospitais sus remédios atendimento hospitalar tratamento leitos religião templo igreja deus bíblia fé jesus segurança polícia crime violência punição arma contrabando ditadura militar golpe 31 de março tortura censura mulher aborto feminicídio feminismo feminista maria da penha petrobras pré-sal refinamento gasolina álcool combustível petrolão corrupção ministério público agu lava-jato mensalão impeachment crime de responsabilidade agronegócio agricultura agrícolas soja lavoura rural indústria desendustrialização empregos competitividade direitos humanos minorias tortura tráfego de pessoas trabalho escravo"

D.2 Classificação Temática

Para a classificação temática, os temas escolhidos e seus respectivos conjuntos de palavras utilizados foram:

- **Agropecuária:** "agropecuária fertilizantes agronegócio abate suínos ovos cabeças bovinos frangos exportação carne animal milho ração aviária laranja safra frutos pomares laranjeiras fazenda pés produzir hectares quilos fruta produtor orgânico consumidor toneladas embrapa bezerros pecuária veterinária filhotes sementes agro

produção água sol área degradação produtor café importação agrícola pescador alimento alimentação açúcar ibge fertilizante lavouras grão bovino soja etanol frutos rural"

- **Saúde:** "saúde médico doença vírus zika pesquisa paciente estudo mosquito epidemia chikungunya tratamento procedimento tremor causa gêmeos dengue transmissão cubano bebês cirurgia cientista risco sintomas dor ultrassom dr aegypt ovário microcefalia gravidez sistema imune imunológico drogas fertilização febre diagnóstico renal sangue insuficiente insuficiência cérebro idade nascimento hipotálamo morte dna corpo cardio muscular vacina"

- **Esporte:** "esporte jogo jogador clube time contrato treino mundial atleta surf futebol disputa penalidade compo estádio ataque atacante bola goleiro treinador seleção técnico campeonato gol pontuação futsal vitória perde perdedor lutador torcedor torcida rival diretor falta conquista prorrogação empate surfista assistência ufc"

- **Educação:** "educação estudo ensino escola médio prova enem universidade faculdade matemática avaliação aluno curso pesquisa inep exame pública mec professor redação criança texto reforma currículo curricular campus leitura literatura desempenho formação qualidade disciplina fies superior analfabeto analfabetismo português física química geometria"

- **Ciência e Tecnologia:** "ciência tecnologia novidades empresa startup smart serviço smartphone consumidor produto google aparelho samsung celular internet inteligência artificial desenvolvimento dispositivo lançamento aplicativo inovar inovação sony conectar conectado comunicação 3g 4g 5g iphone sistema telecomunicações satélite design científico artigo computador tráfego eletrônico apple whatsapp televisão tv telefone avanço espacial"

- **Economia:** "economia trabalho crédito compra banco bilhões milhões vendas contas inflação consumidor juros queda crise taxa resultado econômico gasto pagamento valor financeiro investimento dinheiro índice comércio empresa desemprego fgts limite emprego cartão varejo déficite fundo recessão recuo salário lojista tesouro fiscal inadimplente recurso dólar euro moeda bolsa endividado projeções crescimento capital ações negócios"

- **Política:** "política deputado congresso pt partido estado união reforma lei legislatura legislação pec pmdb aprovar voto bancada população senado senador câmara deputado sindicato candidato candidatura mandato comissão ministério constituição eleição eleições delação judiciário votações prefeitura prefeito vereador assembleia procurador corrupção"

- **Meio Ambiente:** "ambiente área água rio empresa desastres multa seca barragem furacão desmatamento floresta tropical ibama parque preservação região terra planeta poluição ambiental espécie animais plantas platações petróleo emissão gás chuva temporal sol clima temperatura estufa aquecimento global umidade terremoto planeta biodiversidade biologia mar oceano calor energia sustentável madeira reflorestamento tempestade niño florescimento hídrico climática"

- **Direitos Humanos:** "direitos humanos mulher tortura violência morte justiça onu sexual vítima sexual adolescente presídio prevenção união negro branco segurança refugiado homens humanitario conflito sociedade racismo sexismo machismo machista feminismo feminista defensoria estupro jovens criança prostituição assassinato liberdade idoso inclusão social preconceito gay homossexual heterosexual lgbt lésbica bissexual travesti transexual transgênero impunidade imigrante"

- **Segurança:** "segurança ataque polícia suspeito morte crime terror rebelde investigação civil federal guerra onu vítima invasão preso presídio assassinato bombardeio apreensão incidente defesa exército marinha aeronáutica prisão ameaça bomba testemunha promotor policial tragédia assalto protesto"

APÊNDICE E – Índice de Incoerência

A ideia inicial do Tenho Dito foi concebida em uma *hackathon*, promovida pela aceleradora de *startups* Cotidiano. Para a competição, o objetivo do sistema era criar um índice que mostrasse a coerência entre o que os parlamentares diziam em seus discursos em plenário e o que propunham em seus projetos de lei. Entretanto, por motivos pessoais, o autor desse trabalho preferiu não divulgar no Tenho Dito tal índice.

O Índice de Incoerência varia de 0 a 1, onde quanto maior o número, maior é a incoerência do parlamentar. Ou seja, definimos como "incoerente" o parlamentar que discursa muito e propõe pouco sobre um determinado tema, por exemplo.

O valor do Índice de Incoerência i é obtido através da equação:

$$i = \frac{\sum_{t}^{T} \frac{|D_t - P_t|}{max(D_t, P_t)}}{Q_T} \tag{E.1}$$

onde:

- T são todos os temas abordados pelo parlamentar,
- D_t é a porcentagem de discursos classificados como pertencentes ao tema t,
- P_t é a porcentagem de proposições classificadas como pertencentes ao tema t,
- $max(D_t, P_t)$ é o maior valor entre D_t e P_t,
- Q_T é a quantidade total de temas abordados pelo parlamentar.

A tabela mostra o Índice de Incoerência obtido para cada deputado, levando em consideração os dados utilizados nesse trabalho.

Parlamentar	Partido	UF	Discursos	Proposições	Índice de Incoerência
ADILTON SACHETTI	PSB	MT	0	0	0
ALBERTO FILHO	PMDB	MA	0	0	0
ANDRÉ DE PAULA	PSD	PE	0	0	0
ANÍBAL GOMES	PMDB	CE	0	0	0
ARIOSTO HOLANDA	PDT	CE	0	0	0

AROLDE DE OLIVEIRA	PSC	RJ	0	0	0
ARTHUR LIRA	PP	AL	0	0	0
ASSIS MELO	PCdoB	RS	0	0	0
CESAR SOUZA	PSD	SC	0	0	0
CÉSAR MESSIAS	PSB	AC	0	0	0
DEJORGE PATRÍCIO	PRB	RJ	0	0	0
DEOCLIDES MACEDO	PDT	MA	0	0	0
DIMAS FABIANO	PP	MG	0	0	0
EDMAR ARRUDA	PSD	PR	0	0	0
ELMAR NASCIMENTO	DEM	BA	0	0	0
GENECIAS NORONHA	SD	CE	0	0	0
GEORGE HILTON	PSB	MG	0	0	0
HERMES PARCIANELLO	PMDB	PR	0	0	0
IZAQUE SILVA	PSDB	SP	0	0	0
JAIME MARTINS	PSD	MG	0	0	0
JARBAS VASCONCELOS	PMDB	PE	0	0	0
JOSUÉ BENGTSON	PTB	PA	0	0	0
JOSÉ PRIANTE	PMDB	PA	0	0	0
JOÃO CARLOS BACELAR	PR	BA	0	0	0
JOÃO PAULO KLEINÜBING	PSD	SC	0	0	0
JÉSSICA SALES	PMDB	AC	0	0	0
LEONARDO QUINTÃO	PMDB	MG	0	0	0
LUANA COSTA	PSB	MA	0	0	0
LUCIANO BIVAR	PSL	PE	0	0	0
LUCIO VIEIRA LIMA	PMDB	BA	0	0	0
LUIS TIBÉ	PTdoB	MG	0	0	0

LUIZ FERNANDO FARIA	PP	MG	0	0	0
LUZIA FERREIRA	PPS	MG	0	0	0
MACEDO	PP	CE	0	0	0
MAGDA MOFATTO	PR	GO	0	0	0
MARCELO CASTRO	PMDB	PI	0	0	0
MARCELO DELAROLI	PR	RJ	0	0	0
MARCOS ABRÃO	PPS	GO	0	0	0
MARCOS MEDRADO	PODE	BA	0	0	0
MARINHA RAUPP	PMDB	RO	0	0	0
NELSON MEURER	PP	PR	0	0	0
NILSON PINTO	PSDB	PA	0	0	0
NILTON CAPIXABA	PTB	RO	0	0	0
NORMA AYUB	DEM	ES	0	0	0
PASTOR LUCIANO BRAGA	PRB	BA	0	0	0
PAULO ABI-ACKEL	PSDB	MG	0	0	0
PAULO FREIRE	PR	SP	0	0	0
PAULO MALUF	PP	SP	0	0	0
PEDRO CHAVES	PMDB	GO	0	0	0
POLLYANA GAMA	PPS	SP	0	0	0
REINHOLD STEPHANES	PSD	PR	0	0	0
RENATO ANDRADE	PP	MG	0	0	0
RICARDO TEOBALDO	PODE	PE	0	0	0
ROBERTO GÓES	PDT	AP	0	0	0
ROBINSON ALMEIDA	PT	BA	0	0	0
ROGÉRIO SILVA	PROS	MT	0	0	0
RUBENS OTONI	PT	GO	0	0	0
SABINO CASTELO BRANCO	PTB	AM	0	0	0

SARAIVA FELIPE	PMDB	MG	0	0	0
SERGIO ZVEITER	PMDB	RJ	0	0	0
SÉRGIO BRITO	PSD	BA	0	0	0
SÉRGIO MORAES	PTB	RS	0	0	0
TADEU ALENCAR	PSB	PE	0	0	0
TIRIRICA	PR	SP	0	0	0
TONINHO WANDSCHEER	PROS	PR	0	0	0
VAIDON OLIVEIRA	DEM	CE	0	0	0
VALADARES FILHO	PSB	SE	0	0	0
VITOR LIPPI	PSDB	SP	0	0	0
WALNEY ROCHA	PEN	RJ	0	0	0
WALTER IHOSHI	PSD	SP	0	0	0
WILSON BESERRA	PMDB	RJ	0	0	0
WLADIMIR COSTA	SD	PA	0	0	0
WOLNEY QUEIROZ	PDT	PE	0	0	0
YEDA CRUSIUS	PSDB	RS	0	0	0
ZECA DO PT	PT	MS	0	0	0
ZÉ AUGUSTO NALIN	PMDB	RJ	0	0	0
SILAS FREIRE	PODE	PI	3	2	0.292
ALBERTO FRAGA	DEM	DF	64	66	0.474
ANDRÉ AMARAL	PMDB	PB	5	5	0.501
CARLOS HENRIQUE GAGUIM	PODE	TO	45	37	0.532
HILDO ROCHA	PMDB	MA	53	17	0.559
RÔMULO GOUVEIA	PSD	PB	24	102	0.587
CREUZA PEREIRA	PSB	PE	6	1	0.592
CARLOS BEZERRA	PMDB	MT	15	69	0.604
FLAVINHO	PSB	SP	9	26	0.625
ROSANGELA GOMES	PRB	RJ	8	2	0.626
CABO DACIOLO	PTdoB	RJ	7	12	0.632
EDUARDO BOLSONARO	PSC	SP	8	6	0.644

MARCELO ARO	PHS	MG	4	2	0.663
NILTO TATTO	PT	SP	17	10	0.681
VITOR VALIM	PMDB	CE	29	11	0.682
AUGUSTO CARVALHO	SD	DF	20	17	0.69
VALDIR COLATTO	PMDB	SC	35	9	0.691
FRANCISCO FLORIANO	DEM	RJ	11	40	0.692
HELDER SALOMÃO	PT	ES	5	8	0.694
MIRO TEIXEIRA	REDE	RJ	15	4	0.7
MOISÉS DINIZ	PCdoB	AC	1	4	0.704
AFONSO HAMM	PP	RS	8	8	0.706
ALFREDO NASCIMENTO	PR	AM	13	16	0.712
PROFESSOR VICTÓRIO GALLI	PSC	MT	10	16	0.724
JULIO LOPES	PP	RJ	6	12	0.727
LUIZ CARLOS HAULY	PSDB	PR	43	10	0.727
CARLOS MANATO	SD	ES	23	11	0.733
ZECA CAVALCANTI	PTB	PE	1	1	0.733
AFONSO MOTTA	PDT	RS	15	7	0.74
DELEGADO EDSON MOREIRA	PR	MG	61	6	0.742
JOÃO PAULO PAPA	PSDB	SP	1	4	0.75
ROBERTO FREIRE	PPS	SP	2	1	0.75
TAKAYAMA	PSC	PR	1	2	0.75
ZÉ SILVA	SD	MG	9	3	0.75
LAURA CARNEIRO	PMDB	RJ	19	37	0.754
TENENTE LÚCIO	PSB	MG	5	9	0.755
SHÉRIDAN	PSDB	RR	2	2	0.756
LUCIANA SANTOS	PCdoB	PE	2	3	0.757
PROFESSORA DORINHA SEABRA REZENDE	DEM	TO	3	5	0.76
ROBERTO ALVES	PRB	SP	14	6	0.76

CAPITÃO AUGUSTO	PR	SP	26	5	0.768
CABO SABINO	PR	CE	26	56	0.772
LAUDIVIO CARVALHO	SD	MG	7	14	0.775
CARMEN ZANOTTO	PPS	SC	37	7	0.78
CHICO D'ANGELO	PT	RJ	7	9	0.786
ELIZIANE GAMA	PPS	MA	7	2	0.786
IRACEMA PORTELLA	PP	PI	17	6	0.787
ALCEU MOREIRA	PMDB	RS	4	6	0.788
ROCHA	PSDB	AC	36	9	0.788
BACELAR	PODE	BA	7	5	0.79
ELIZEU DIONIZIO	PSDB	MS	6	8	0.793
POMPEO DE MATTOS	PDT	RS	27	9	0.798
ALAN RICK	PRB	AC	5	7	0.8
BRUNA FURLAN	PSDB	SP	2	1	0.8
MARCO MAIA	PT	RS	3	9	0.8
VICENTINHO JÚNIOR	PR	TO	7	13	0.801
ARNALDO FARIA DE SÁ	PTB	SP	32	15	0.802
CARLOS ZARATTINI	PT	SP	11	6	0.803
RONALDO CARLETTO	PP	BA	5	12	0.803
ERIKA KOKAY	PT	DF	88	14	0.804
ROGÉRIO MARINHO	PSDB	RN	3	1	0.806
GLAUBER BRAGA	PSOL	RJ	16	5	0.809
LINCOLN PORTELA	PRB	MG	12	4	0.81
LUCIANO DUCCI	PSB	PR	14	4	0.81
RICARDO IZAR	PP	SP	4	8	0.81
PATRUS ANANIAS	PT	MG	3	2	0.812

DR. JORGE SILVA	PHS	ES	5	4	0.815
MIGUEL LOMBARDI	PR	SP	2	5	0.815
ARTHUR VIRGÍLIO BISNETO	PSDB	AM	4	3	0.823
JOÃO ARRUDA	PMDB	PR	6	5	0.825
PAULO TEIXEIRA	PT	SP	3	2	0.828
GILBERTO NASCIMENTO	PSC	SP	19	3	0.831
AELTON FREITAS	PR	MG	2	1	0.833
EDIO LOPES	PR	RR	1	1	0.833
JHC	PSB	AL	13	4	0.834
JOÃO RODRIGUES	PSD	SC	4	5	0.834
MARCOS ROGÉRIO	DEM	RO	3	2	0.836
RUBENS PEREIRA JÚNIOR	PCdoB	MA	13	10	0.839
BONIFÁCIO DE ANDRADA	PSDB	MG	6	12	0.842
MÁRCIO MARINHO	PRB	BA	6	2	0.842
ODORICO MONTEIRO	PSB	CE	6	2	0.843
VALMIR ASSUNÇÃO	PT	BA	29	3	0.844
DIEGO GARCIA	PHS	PR	5	11	0.846
RUBENS BUENO	PPS	PR	18	7	0.846
MOSES RODRIGUES	PMDB	CE	6	20	0.847
ZÉ CARLOS	PT	MA	1	2	0.847
JAIR BOLSONARO	PSC	RJ	8	5	0.849
RONALDO FONSECA	PROS	DF	7	3	0.849
ALEX MANENTE	PPS	SP	2	1	0.85
HERCULANO PASSOS	PSD	SP	2	3	0.85
JÚLIA MARINHO	PSC	PA	3	3	0.852
MARCIO ALVINO	PR	SP	12	3	0.854
PEDRO UCZAI	PT	SC	4	3	0.854

RAIMUNDO GOMES DE MATOS	PSDB	CE	10	2	0.856
LAERTE BESSA	PR	DF	10	7	0.857
IZALCI LUCAS	PSDB	DF	6	4	0.858
WEVERTON ROCHA	PDT	MA	6	21	0.858
VINICIUS CARVALHO	PRB	SP	20	12	0.861
JOÃO DERLY	REDE	RS	5	22	0.863
ARTHUR OLIVEIRA MAIA	PPS	BA	9	3	0.866
SUBTENENTE GONZAGA	PDT	MG	3	6	0.868
SÓSTENES CAVALCANTE	DEM	RJ	15	12	0.869
RAFAEL MOTTA	PSB	RN	3	9	0.871
MARCUS PESTANA	PSDB	MG	14	3	0.874
NILSON LEITÃO	PSDB	MT	3	4	0.874
KEIKO OTA	PSB	SP	5	2	0.876
CELSO PANSERA	PMDB	RJ	2	4	0.877
GEOVANIA DE SÁ	PSDB	SC	7	4	0.877
VICENTE CANDIDO	PT	SP	1	2	0.878
ALEXANDRE BALDY	PODE	GO	6	1	0.881
MAJOR OLIMPIO	SD	SP	23	4	0.881
CARLOS GOMES	PRB	RS	4	2	0.883
SEVERINO NINHO	PSB	PE	17	3	0.884
ANTONIO BULHÕES	PRB	SP	13	6	0.885
ROBERTO DE LUCENA	PV	SP	20	6	0.888
DANILO CABRAL	PSB	PE	3	1	0.889
PEPE VARGAS	PT	RS	8	2	0.89
ONYX LORENZONI	DEM	RS	13	6	0.891
MARCOS REATEGUI	PSD	AP	7	3	0.894

HEITOR SCHUCH	PSB	RS	30	4	0.895
LUCIO MOSQUINI	PMDB	RO	2	4	0.896
VALMIR PRASCIDELLI	PT	SP	3	1	0.896
DANILO FORTE	PSB	CE	11	1	0.9
WILSON FILHO	PTB	PB	2	13	0.901
DOMINGOS SÁVIO	PSDB	MG	23	2	0.902
LUIZ COUTO	PT	PB	74	3	0.902
RÔNEY NEMER	PP	DF	2	3	0.902
CAIO NARCIO	PSDB	MG	11	6	0.903
LOBBE NETO	PSDB	SP	15	3	0.903
ALESSANDRO MOLON	REDE	RJ	4	2	0.905
GOULART	PSD	SP	3	26	0.905
FELIPE MAIA	DEM	RN	10	5	0.906
DÂMINA PEREIRA	PSL	MG	3	3	0.908
GIVALDO VIEIRA	PT	ES	24	2	0.912
ESPERIDIÃO AMIN	PP	SC	9	2	0.913
CÉLIO SILVEIRA	PSDB	GO	3	11	0.916
ANTONIO CARLOS MENDES THAME	PV	SP	2	4	0.917
CHICO ALENCAR	PSOL	RJ	36	2	0.917
CONCEIÇÃO SAMPAIO	PP	AM	2	2	0.917
JOÃO CAMPOS	PRB	GO	3	1	0.917
PEDRO CUNHA LIMA	PSDB	PB	3	5	0.917
SÁGUAS MORAES	PT	MT	18	2	0.917
GERALDO RESENDE	PSDB	MS	22	3	0.918
MARA GABRILLI	PSDB	SP	1	7	0.923
FAUSTO PINATO	PP	SP	1	13	0.924
JOÃO DANIEL	PT	SE	50	4	0.924
BETO ROSADO	PP	RN	3	4	0.927
JOVAIR ARANTES	PTB	GO	5	1	0.929
BETINHO GOMES	PSDB	PE	14	3	0.93

VANDERLEI MACRIS	PSDB	SP	10	2	0.93
CACÁ LEÃO	PP	BA	1	3	0.931
TEREZA CRISTINA	PSB	MS	2	3	0.931
BENEDITA DA SILVA	PT	RJ	24	2	0.933
CABUÇU BORGES	PMDB	AP	4	4	0.933
DELEGADO WALDIR	PR	GO	1	17	0.933
JORGINHO MELLO	PR	SC	5	3	0.933
DR. SINVAL MALHEIROS	PODE	SP	5	10	0.935
VICTOR MENDES	PSD	MA	2	7	0.936
ALEXANDRE VALLE	PR	RJ	3	3	0.937
RENATO MOLLING	PP	RS	4	3	0.937
THIAGO PEIXOTO	PSD	GO	2	2	0.937
AUGUSTO COUTINHO	SD	PE	5	2	0.938
RONALDO BENEDET	PMDB	SC	9	3	0.938
ASSIS CARVALHO	PT	PI	17	1	0.939
DANIEL COELHO	PSDB	PE	24	1	0.94
PASTOR EURICO	PHS	PE	9	1	0.941
SORAYA SANTOS	PMDB	RJ	7	4	0.941
ZÉ GERALDO	PT	PA	43	1	0.942
GONZAGA PATRIOTA	PSB	PE	38	2	0.944
FÁBIO MITIDIERI	PSD	SE	1	8	0.948
JOSÉ ROCHA	PR	BA	8	1	0.952
GIOVANI CHERINI	PR	RS	9	3	0.954
NELSON PELLEGRINO	PT	BA	3	1	0.955
JOSE STÉDILE	PSB	RS	11	2	0.958
MARCUS VICENTE	PP	ES	4	2	0.958
MARIA DO ROSÁRIO	PT	RS	21	1	0.958

JÔ MORAES	PCdoB	MG	20	1	0.959
SIMÃO SESSIM	PP	RJ	22	8	0.959
LEO DE BRITO	PT	AC	25	1	0.961
GIUSEPPE VECCI	PSDB	GO	1	4	0.964
REGINALDO LOPES	PT	MG	9	1	0.965
OTAVIO LEITE	PSDB	RJ	4	6	0.966
RODRIGO PACHECO	PMDB	MG	3	3	0.966
CLEBER VERDE	PRB	MA	6	15	0.967
CRISTIANE BRASIL	PTB	RJ	2	1	0.967
ROBERTO BRITTO	PP	BA	5	1	0.967
JERÔNIMO GOERGEN	PP	RS	1	15	0.968
MARCON	PT	RS	37	3	0.968
PR. MARCO FELICIANO	PSC	SP	18	2	0.968
LUIZ SÉRGIO	PT	RJ	11	1	0.969
VICENTINHO	PT	SP	11	1	0.971
ADELMO CARNEIRO LEÃO	PT	MG	2	1	0.972
WADIH DAMOUS	PT	RJ	2	6	0.972
MARIANA CARVALHO	PSDB	RO	5	22	0.974
MARCELO MATOS	PHS	RJ	3	6	0.975
RODRIGO MARTINS	PSB	PI	10	7	0.976
JOSI NUNES	PMDB	TO	12	8	0.979
IVAN VALENTE	PSOL	SP	37	2	0.98
LUIZ LAURO FILHO	PSB	SP	1	12	0.981
SERGIO SOUZA	PMDB	PR	7	1	0.981
ANA PERUGINI	PT	SP	2	6	0.982
HENRIQUE FONTANA	PT	RS	8	1	0.982
MARIA HELENA	PSB	RR	6	1	0.983
LUIZA ERUNDINA	PSOL	SP	7	1	0.984
PAES LANDIM	PTB	PI	28	2	0.984

DANIEL VILELA	PMDB	GO	3	7	0.985
DAVIDSON MAGALHÃES	PCdoB	BA	18	1	0.985
BENJAMIN MARANHÃO	SD	PB	10	1	0.986
CHICO LOPES	PCdoB	CE	28	2	0.986
FÉLIX MENDONÇA JÚNIOR	PDT	BA	1	8	0.986
JOSÉ GUIMARÃES	PT	CE	6	2	0.987
PAULO AZI	DEM	BA	2	5	0.987
FLÁVIA MORAIS	PDT	GO	1	10	0.989
VANDER LOUBET	PT	MS	1	3	0.989
PAULO FOLETTO	PSB	ES	16	1	0.99
DANIEL ALMEIDA	PCdoB	BA	23	1	0.992
GORETE PEREIRA	PR	CE	3	6	0.992
CAETANO	PT	BA	12	1	0.993
EZEQUIEL TEIXEIRA	PODE	RJ	2	7	0.993
JEFFERSON CAMPOS	PSD	SP	20	2	0.993
ROBERTO SALES	PRB	RJ	1	1	0.993
ALIEL MACHADO	REDE	PR	15	1	0.994
COVATTI FILHO	PP	RS	2	9	0.994
STEFANO AGUIAR	PSD	MG	11	1	0.994
CELSO RUSSOMANNO	PRB	SP	1	6	0.995
CELSO MALDANER	PMDB	SC	33	1	0.996
CHRISTIANE DE SOUZA YARED	PR	PR	1	7	0.996
ARNALDO JORDY	PPS	PA	40	1	0.998
FELIPE BORNIER	PROS	RJ	1	46	0.998
JANETE CAPIBERIBE	PSB	AP	35	1	0.998
ABEL MESQUITA JR.	DEM	RR	1	0	1
ADAIL CARNEIRO	PP	CE	1	5	1

ADALBERTO CAVALCANTI	PTB	PE	1	1	1
ADELSON BARRETO	PR	SE	1	0	1
ADEMIR CAMILO	PODE	MG	0	1	1
AFONSO FLORENCE	PT	BA	12	0	1
AGUINALDO RIBEIRO	PP	PB	1	0	1
ALEX CANZIANI	PTB	PR	4	0	1
ALEXANDRE LEITE	DEM	SP	0	10	1
ALEXANDRE SERFIOTIS	PMDB	RJ	3	0	1
ALFREDO KAEFER	PSL	PR	7	0	1
ALICE PORTUGAL	PCdoB	BA	6	1	1
ALTINEU CÔRTES	PMDB	RJ	2	0	1
ALUISIO MENDES	PODE	MA	1	2	1
ANDRE MOURA	PSC	SE	4	0	1
ANDRES SANCHEZ	PT	SP	2	1	1
ANDRÉ ABDON	PP	AP	0	2	1
ANDRÉ FIGUEIREDO	PDT	CE	0	1	1
ANDRÉ FUFUCA	PP	MA	1	1	1
ANGELIM	PT	AC	23	0	1
ANTONIO BRITO	PSD	BA	1	3	1
ANTÔNIO JÁCOME	PODE	RN	2	0	1
ARLINDO CHINAGLIA	PT	SP	1	0	1
ASSIS DO COUTO	PDT	PR	1	0	1
AUREO	SD	RJ	0	10	1
BALEIA ROSSI	PMDB	SP	4	1	1
BEBETO	PSB	BA	7	0	1
BENITO GAMA	PTB	BA	7	0	1
BETO FARO	PT	PA	1	0	1
BETO MANSUR	PRB	SP	8	0	1
BETO SALAME	PP	PA	0	4	1

BILAC PINTO	PR	MG	4	0	1
BOHN GASS	PT	RS	21	0	1
BRUNNY	PR	MG	3	2	1
CAJAR NARDES	PR	RS	0	3	1
CARLOS ANDRADE	PHS	RR	6	0	1
CARLOS EDUARDO CADOCA	PDT	PE	0	1	1
CARLOS MARUN	PMDB	MS	5	0	1
CARLOS MELLES	DEM	MG	3	0	1
CARLOS SAMPAIO	PSDB	SP	1	0	1
CELSO JACOB	PMDB	RJ	0	11	1
CLAUDIO CAJADO	DEM	BA	7	0	1
CÉSAR HALUM	PRB	TO	1	4	1
CÍCERO ALMEIDA	PMDB	AL	0	2	1
DAGOBERTO NOGUEIRA	PDT	MS	0	3	1
DAMIÃO FELICIANO	PDT	PB	2	2	1
DANRLEI DE DEUS HINTERHOLZ	PSD	RS	1	2	1
DARCÍSIO PERONDI	PMDB	RS	23	0	1
DELEGADO FRANCISCHINI	SD	PR	0	2	1
DELEGADO ÉDER MAURO	PSD	PA	8	0	1
DELEY	PTB	RJ	1	2	1
DIEGO ANDRADE	PSD	MG	0	6	1
DILCEU SPERAFICO	PP	PR	0	1	1
DOMINGOS NETO	PSD	CE	2	0	1
DULCE MIRANDA	PMDB	TO	1	3	1
DÉCIO LIMA	PT	SC	15	0	1
EDMILSON RODRIGUES	PSOL	PA	58	0	1
EDUARDO BARBOSA	PSDB	MG	0	11	1

EDUARDO CURY	PSDB	SP	4	0	1
EDUARDO DA FONTE	PP	PE	0	2	1
EFRAIM FILHO	DEM	PB	4	1	1
ELCIONE BARBALHO	PMDB	PA	0	1	1
ELI CORRÊA FILHO	DEM	SP	0	2	1
ENIO VERRI	PT	PR	5	0	1
ERIVELTON SANTANA	PEN	BA	0	2	1
EROS BIONDINI	PROS	MG	0	4	1
EVAIR VIEIRA DE MELO	PV	ES	6	1	1
EVANDRO GUSSI	PV	SP	6	0	1
EVANDRO ROMAN	PSD	PR	0	3	1
EXPEDITO NETTO	PSD	RO	4	0	1
EZEQUIEL FONSECA	PP	MT	1	2	1
FABIO GARCIA	PSB	MT	4	0	1
FABIO REIS	PMDB	SE	1	0	1
FERNANDO MONTEIRO	PP	PE	0	1	1
FLAVIANO MELO	PMDB	AC	5	0	1
FRANCISCO CHAPADINHA	PODE	PA	0	17	1
FRANKLIN	PP	MG	0	4	1
FÁBIO FARIA	PSD	RN	3	1	1
FÁBIO RAMALHO	PMDB	MG	3	0	1
FÁBIO SOUSA	PSDB	GO	6	4	1
GABRIEL GUIMARÃES	PT	MG	0	1	1
GIACOBO	PR	PR	3	0	1
GIVALDO CARIMBÃO	PHS	AL	2	0	1
GUILHERME COELHO	PSDB	PE	1	0	1

GUILHERME MUSSI	PP	SP	0	3	1
HERÁCLITO FORTES	PSB	PI	23	0	1
HEULER CRUVINEL	PSD	GO	0	1	1
HIRAN GONÇALVES	PP	RR	0	1	1
HISSA ABRAHÃO	PDT	AM	3	1	1
HUGO LEAL	PSB	RJ	1	2	1
HUGO MOTTA	PMDB	PB	1	1	1
HÉLIO LEITE	DEM	PA	2	1	1
IRAJÁ ABREU	PSD	TO	1	2	1
IRMÃO LAZARO	PSC	BA	0	1	1
JANDIRA FEGHALI	PCdoB	RJ	13	1	1
JEAN WYLLYS	PSOL	RJ	0	7	1
JHONATAN DE JESUS	PRB	RR	0	3	1
JOAQUIM PASSARINHO	PSD	PA	11	0	1
JONES MARTINS	PMDB	RS	19	0	1
JONY MARCOS	PRB	SE	3	0	1
JORGE BOEIRA	PP	SC	1	0	1
JORGE CÔRTE REAL	PTB	PE	1	3	1
JORGE SOLLA	PT	BA	14	1	1
JORGE TADEU MUDALEN	DEM	SP	2	0	1
JOSÉ AIRTON CIRILO	PT	CE	12	1	1
JOSÉ CARLOS ALELUIA	DEM	BA	3	0	1
JOSÉ CARLOS ARAÚJO	PR	BA	1	0	1
JOSÉ FOGAÇA	PMDB	RS	5	0	1
JOSÉ MENTOR	PT	SP	0	1	1
JOSÉ NUNES	PSD	BA	1	0	1

JOSÉ OTÁVIO GERMANO	PP	RS	0	1	1
JOSÉ REINALDO	PSB	MA	4	1	1
JOZI ARAÚJO	PODE	AP	1	0	1
JOÃO FERNANDO COUTINHO	PSB	PE	2	2	1
JOÃO GUALBERTO	PSDB	BA	6	0	1
JOÃO MARCELO SOUZA	PMDB	MA	2	0	1
JUNIOR MARRECA	PEN	MA	1	0	1
JUSCELINO FILHO	DEM	MA	2	0	1
JUTAHY JUNIOR	PSDB	BA	3	0	1
JÚLIO CESAR	PSD	PI	5	0	1
JÚLIO DELGADO	PSB	MG	3	0	1
LAERCIO OLIVEIRA	SD	SE	0	18	1
LEANDRE	PV	PR	0	8	1
LELO COIMBRA	PMDB	ES	1	0	1
LEONARDO MONTEIRO	PT	MG	7	0	1
LEOPOLDO MEYER	PSB	PR	0	1	1
LEÔNIDAS CRISTINO	PDT	CE	2	1	1
LINDOMAR GARÇON	PRB	RO	0	3	1
LUCAS VERGILIO	SD	GO	0	3	1
LUIS CARLOS HEINZE	PP	RS	19	0	1
LUIZ CARLOS RAMOS	PODE	RJ	0	3	1
LUIZ CLÁUDIO	PR	RO	2	0	1
LUIZ NISHIMORI	PR	PR	3	0	1
LUIZIANNE LINS	PT	CE	0	3	1
LÁZARO BOTELHO	PP	TO	0	1	1
LÚCIO VALE	PR	PA	0	4	1
MAIA FILHO	PP	PI	0	3	1

MANDETTA	DEM	MS	1	0	1
MARCELO AGUIAR	DEM	SP	1	3	1
MARCELO SQUASSONI	PRB	SP	0	1	1
MARCELO ÁLVARO ANTÔNIO	PR	MG	0	2	1
MARCO ANTÔNIO CABRAL	PMDB	RJ	0	7	1
MARCO TEBALDI	PSDB	SC	1	2	1
MARCOS MONTES	PSD	MG	2	0	1
MARCOS SOARES	DEM	RJ	0	3	1
MARGARIDA SALOMÃO	PT	MG	1	0	1
MARINALDO ROSENDO	PSB	PE	0	15	1
MAURO LOPES	PMDB	MG	0	14	1
MAURO MARIANI	PMDB	SC	0	3	1
MAURO PEREIRA	PMDB	RS	101	0	1
MIGUEL HADDAD	PSDB	SP	3	2	1
MILTON MONTI	PR	SP	0	1	1
MISAEL VARELLA	DEM	MG	23	0	1
MISSIONÁRIO JOSÉ OLIMPIO	DEM	SP	2	0	1
MÁRIO NEGROMONTE JR.	PP	BA	0	2	1
NELSON MARQUEZELLI	PTB	SP	2	0	1
NELSON PADOVANI	PSDB	PR	0	1	1
NEWTON CARDOSO JR	PMDB	MG	4	1	1
NIVALDO ALBUQUERQUE	PRP	AL	0	6	1
ORLANDO SILVA	PCdoB	SP	1	2	1
OSMAR BERTOLDI	DEM	PR	0	3	1
OSMAR SERRAGLIO	PMDB	PR	2	0	1

PADRE JOÃO	PT	MG	16	0	1
PAUDERNEY AVELINO	DEM	AM	10	0	1
PAULO FEIJÓ	PR	RJ	1	1	1
PAULO HENRIQUE LUSTOSA	PP	CE	1	0	1
PAULO MAGALHÃES	PSD	BA	0	4	1
PAULO PEREIRA DA SILVA	SD	SP	0	2	1
PAULO PIMENTA	PT	RS	6	0	1
PAULÃO	PT	AL	24	0	1
PEDRO FERNANDES	PTB	MA	4	0	1
PEDRO PAULO	PMDB	RJ	0	1	1
PEDRO VILELA	PSDB	AL	2	0	1
PROFESSORA MARCIVANIA	PCdoB	AP	5	1	1
RAQUEL MUNIZ	PSD	MG	10	0	1
REMÍDIO MONAI	PR	RR	1	0	1
RENATA ABREU	PODE	SP	0	10	1
RENZO BRAZ	PP	MG	1	4	1
RICARDO TRIPOLI	PSDB	SP	1	2	1
ROBERTO BALESTRA	PP	GO	5	0	1
RODRIGO DE CASTRO	PSDB	MG	9	0	1
RODRIGO MAIA	DEM	RJ	4	0	1
ROGÉRIO PENINHA MENDONÇA	PMDB	SC	2	5	1
ROGÉRIO ROSSO	PSD	DF	0	2	1
RONALDO LESSA	PDT	AL	1	0	1
RONALDO MARTINS	PRB	CE	2	3	1
ROSINHA DA ADEFAL	PTdoB	AL	1	0	1

SANDRO ALEX	PSD	PR	2	2	1
SERGIO VIDIGAL	PDT	ES	0	3	1
SILAS CÂMARA	PRB	AM	4	0	1
SILVIO COSTA	PTdoB	PE	14	0	1
SILVIO TORRES	PSDB	SP	8	0	1
SIMONE MORGADO	PMDB	PA	0	2	1
SÉRGIO REIS	PRB	SP	0	1	1
TONINHO PINHEIRO	PP	MG	1	4	1
ULDURICO JUNIOR	PV	BA	1	3	1
VENEZIANO VITAL DO RÊGO	PMDB	PB	0	5	1
VINICIUS GURGEL	PR	AP	0	2	1
WALDENOR PEREIRA	PT	BA	11	0	1
WALDIR MARANHÃO	PP	MA	1	0	1
WALTER ALVES	PMDB	RN	0	3	1
WELITON PRADO	PMB	MG	10	0	1
WELLINGTON ROBERTO	PR	PB	0	2	1
ZECA DIRCEU	PT	PR	1	0	1
ZENAIDE MAIA	PR	RN	0	1	1
ÁTILA LINS	PSD	AM	21	0	1
ÁTILA LIRA	PSB	PI	9	0	1

Tabela 6 – Índices de incoerência dos parlamentares