UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE FÍSICA
PROGRAMA DE PÓS-GRADUAÇÃO EM FÍSICA

GUSTAVO FRANKLIN VIEIRA DOS SANTOS

ANÁLISE DE DISCURSOS PARLAMENTARES NA CÂMARA DOS DEPUTADOS

FORTALEZA
2025

GUSTAVO FRANKLIN VIEIRA DOS SANTOS

ANÁLISE DE DISCURSOS PARLAMENTARES NA CÂMARA DOS DEPUTADOS

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Física da Universidade Federal do Ceará, como requisito parcial para a obtenção do Título de Mestre em Física. Área de Concentração: Física.

Orientador: Prof. Dr. Saulo Davi Soares e Reis.

FORTALEZA
2025

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

D1f Santos, Gustavo Franklin Vieira dos.
Análise de Discursos Parlamentares na Câmara dos Deputados / Gustavo Franklin Vieira dos Santos. - 2025.
54 f. : il. color.

Dissertação (mestrado) - Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Física, Fortaleza, 2025.
Orientação: Prof. Dr. Saulo Davi Soares e Reis.

1. análise de discursos; 2. aprendizado de máquina; 3. processamento de linguagem; 4. câmara dos deputados; 5. latent dirichlet. I. Título.

CDD 530

GUSTAVO FRANKLIN VIEIRA DOS SANTOS

ANÁLISE DE DISCURSOS PARLAMENTARES NA CÂMARA DOS DEPUTADOS

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Física da Universidade Federal do Ceará, como requisito parcial para a obtenção do Título de Mestre em Física. Área de Concentração: Física.

Aprovada em 31/07/2025.

BANCA EXAMINADORA

Prof. Dr. Saulo Davi Soares e Reis (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. André Auto Moreira
Universidade Federal do Ceará (UFC)

Prof. Dr. Erneson Alves de Oliveira
Universidade de Fortaleza (UNIFOR)

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001

RESUMO

Este trabalho tem como objetivo analisar discursos proferidos por deputados federais brasileiros, utilizando técnicas de Processamento de Linguagem Natural e modelagem de tópicos. Foram coletados mais de 900 mil discursos da Câmara dos Deputados, abrangendo o período de 1950 a 2024, realizando uma análise exploratória desses dados, entretanto, será feita a análise dos tópicos somente do ano de 2001, para que assim seja possível validar a técnica proposta. A análise emprega modelos amplamente utilizados na literatura, como o Latent Dirichlet Allocation (LDA), e modelos mais recentes, como o BERTopic, para identificar padrões temáticos e a evolução de assuntos debatidos ao longo do tempo. Para isso, realizou-se a coleta automatizada dos dados via web scraping e aplicou-se uma série de etapas de pré-processamento textual, como lematização e remoção de stop words. A modelagem permitiu não apenas a identificação de tópicos recorrentes, mas também a correlação entre eventos históricos e o conteúdo dos discursos. A abordagem destacou o potencial dos métodos computacionais para análise de grandes volumes de dados textuais no contexto político.

Palavras-chave: análise de discursos; aprendizado de máquina; processamento de linguagem; câmara dos deputados;latent dirichlet.

ABSTRACT

This work aims to analyze speeches delivered by Brazilian federal deputies using Natural Language Processing techniques and topic modeling. More than 900,000 speeches from the Chamber of Deputies were collected, covering the period from 1950 to 2024, and an exploratory analysis of this data was conducted. However, the topic modeling analysis focuses exclusively on the year 2001, in order to validate the proposed technique. The analysis employs models widely used in the literature, such as Latent Dirichlet Allocation (LDA), as well as more recent models like BERTopic, to identify thematic patterns and the evolution of topics debated over time. To achieve this, data was automatically collected via web scraping, and a series of textual preprocessing steps were applied, such as lemmatization and stopword removal. The modeling enabled not only the identification of recurring topics but also the correlation between historical events and the content of the speeches. This approach highlights the potential of computational methods for analyzing large volumes of textual data in a political context.

Keywords: speech analyzis; machine learning; language processing; chamber of deputies; latent dirichlet.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

O Brasil, ao longo de sua história, teve sete constituições [1], sendo a de 1988, conhecida como Constituição Cidadã, a mais recente. Ela é composta por 250 artigos, o que a torna a segunda maior constituição do mundo. Até o momento, 145 emendas foram incorporadas [2]. A Constituição assegura [3] uma série de direitos e garantias, como, por exemplo, direitos e deveres, direitos sociais, nacionalidade, direitos políticos e partidos políticos. Para este trabalho, o foco será no Título IV - Organização dos Poderes, que abrange os artigos 44 ao 135, onde são definidas a composição e as funções dos três poderes.

Podemos dividir o Poder Executivo em três esferas: o Poder Executivo Federal, o Poder Executivo Estadual e o Poder Executivo Municipal. O papel do Poder Executivo Federal, ou Governo Federal, é desempenhado pelo Presidente da República, que é assessorado pelos Ministros de Estado [4]. A principal função do Presidente é administrar o Estado, e, no âmbito federal, isso significa gerir o país como um todo.

O Poder Executivo Estadual é exercido pelo governador do estado, que é assistido pelos secretários estaduais. As responsabilidades do governador são estabelecidas pela constituição estadual, de acordo com as normas da Constituição Federal e com a estrutura do Executivo da União [5].

Já o Poder Executivo Municipal é desempenhado pelo prefeito, que conta com o auxílio dos secretários municipais. Esses secretários são escolhidos pelo prefeito e permanecem em seus cargos enquanto tiverem a confiança do chefe do Executivo municipal.

O Poder Legislativo é exercido pelo Congresso Nacional, que é composto pela Câmara dos Deputados e pelo Senado Federal. A Câmara dos Deputados é formada por representantes do povo, sendo que o número de deputados eleitos [6] é proporcional à população dos Estados e do Distrito Federal, totalizando 513 deputados. Já o número de Senadores, por representarem os Estados, é fixo (3 para cada Estado e o Distrito Federal), totalizando 81 Senadores. Essa divisão visa manter o equilíbrio na Federação, evitando que algum Estado tenha sub-representação ou super-representação. Outra diferença entre as duas casas é que, enquanto os Deputados são eleitos para mandatos de 4 anos, os Senadores possuem mandatos de 8 anos e têm 2 suplentes [7].

As atribuições do Congresso Nacional são diversas, incluindo questões orçamentárias (arrecadação e gastos anuais), impostos, planos de desenvolvimento, composição das Forças Armadas, limites e divisões do território nacional, telecomunicações e radiodifusão, criação ou extinção de Ministérios, órgãos da administração pública e cargos, entre outros [8].

Existem também atribuições exclusivas do Congresso Nacional, como o direito de processar e julgar as mais altas autoridades da República, incluindo o Presidente da República. Além disso, o Congresso tem a responsabilidade de aprovar nomes indicados para importantes cargos da administração federal, como embaixadores e Ministros do Supremo Tribunal Federal.

Também cabe a ele decidir sobre operações financeiras, incluindo empréstimos, em âmbito federal, estadual, distrital e municipal. Essas atribuições evidenciam a importância do Poder Legislativo em uma democracia como a do Brasil.

O Poder Judiciário é exercido por juízes e tem a prerrogativa de julgar, com base nas regras constitucionais e nas leis criadas pelo Poder Legislativo. Sua instância máxima é o Supremo Tribunal Federal (STF) [9]. A principal função do Poder Judiciário é defender os direitos de cada cidadão, julgar, de acordo com a lei, os conflitos entre cidadãos, entidades e o Estado [10].

Este trabalho tem como foco os discursos proferidos nas sessões da Câmara dos Deputados, concentrando-se, portanto, no Poder Legislativo. Antes de aprofundar a discussão sobre a Câmara, é importante destacar sua relevância no sistema democrático brasileiro, o que envolve compreender o processo de elaboração e aprovação de leis no país.

A criação de uma lei, como a Lei Seca [11] proposta por Hugo leal [12], começa com a apresentação de um projeto de lei por um Deputado ou Senador. Após sua proposição, o projeto é analisado pelas comissões legislativas. Tanto a Câmara dos Deputados quanto o Senado possuem comissões próprias, parlamentares, permanentes ou temporárias, que exercem funções legislativas e fiscalizadoras, conforme definido pela Constituição Federal e seus regimentos internos [13]. Nessas comissões, são apresentados e analisados dados, antecedentes e circunstâncias relacionadas ao projeto. Após discussão, a comissão emite um parecer que orientará o Plenário na apreciação da matéria, indicando se o projeto deve ser aprovado (com ou sem modificações) ou rejeitado.

Uma vez aprovado em uma das Casas, seja a Câmara dos Deputados ou o Senado, o projeto é enviado à outra Casa, denominada Casa Revisora, para nova análise. Por exemplo, se o projeto for originário da Câmara, será revisado pelo Senado. Caso haja modificações, o projeto retorna à Casa de origem para que estas sejam aprovadas ou rejeitadas. A aprovação de um projeto exige o voto favorável da maioria dos membros presentes, desde que haja quórum mínimo de metade dos parlamentares, no caso de projetos de lei ordinária. Já para projetos de lei complementar, a Constituição Federal requer aprovação por maioria absoluta, ou seja, 41 Senadores e 257 Deputados [14].

Dada a significativa diferença no número de Deputados em comparação ao de Senadores, é plausível concluir que a quantidade de projetos de lei propostos pelos Deputados seja proporcionalmente maior. Apenas em 2024, por exemplo, foram apresentados 4.364 projetos de lei pela Câmara dos Deputados [15]. Esse volume de propostas reflete diretamente na quantidade de discursos e reuniões realizadas pelas comissões. No mesmo ano, foram registrados 11.839 discursos, o que evidencia a intensa atividade legislativa e gera uma grande densidade de dados disponíveis, em parte devido à Lei de Acesso à Informação [16]. O site da Câmara dos Deputados oferece acesso detalhado às informações sobre debates e votações de propostas legislativas, leis em vigor, a atuação e os gastos de cada Deputado, além de dados sobre funcionários, concursos, licitações e contratos.

A investigação sobre o comportamento legislativo no Brasil tem se beneficiado da incorporação de métodos cada vez mais avançados, que aliam técnicas quantitativas, ferramentas computacionais e fundamentos da teoria política. Com ênfase especial na utilização de redes complexas, modelagem de tópicos e análise de discursos parlamentares, evidenciando a evolução metodológica e os focos temáticos predominantes nas pesquisas voltadas ao legislativo brasileiro.

A tese de Brito [17] é uma contribuição abrangente no uso de redes complexas para estudar o Congresso Nacional. Nessa pesquisa, o autor propõe a construção de redes de co-votação baseadas na similaridade dos votos entre os deputados, representando essas interações como grafos, nos quais os nós correspondem aos parlamentares e as conexões indicam proximidade no padrão de votação. Aplicando técnicas como *edge pruning*, detecção de comunidades e análise de métricas topológicas (tais como modularidade, isolamento e fragmentação), o estudo revela dinâmicas de coesão partidária e padrões de formação e dissolução de alianças ao longo de quase trinta anos (1991–2019). Dentre os principais resultados, destaca-se a menor coesão observada nas redes mais recentes em comparação com os anos 1990, sugerindo uma crescente fragmentação partidária. As comunidades identificadas, embora muitas vezes próximas às coligações governistas, nem sempre coincidem com os agrupamentos partidários oficiais, apontando para a complexidade intrínseca do modelo de presidencialismo de coalizão brasileiro.

No trabalho de Amancio [18], observa-se um avanço metodológico significativo com a incorporação da análise textual ao estudo estrutural. Utilizando o modelo LDA (*Latent Dirichlet Allocation*), os autores identificam dez macrotemas recorrentes nas proposições legislativas analisadas entre 2002 e 2022, entre eles, educação, tributos, trabalho e energia. A partir disso, redes temáticas distintas são construídas, possibilitando avaliar o impacto do conteúdo das matérias sobre a coesão dos blocos parlamentares. O estudo demonstra que os agrupamentos extraídos dessas redes temáticas não coincidem com os partidos formais e que, em muitos casos, a modularidade dessas comunidades supera a dos agrupamentos partidários. Adicionalmente, constata-se que determinados temas geram mais ou menos disciplina partidária, dependendo da orientação ideológica e do contexto político em que estão inseridos. Essa combinação entre estrutura de rede e conteúdo semântico revela nuances ideológicas e programáticas não capturadas apenas pelas legendas partidárias.

Distanciando-se da ênfase em dados de votação, a pesquisa de Moreira[19] volta-se à análise da retórica parlamentar, com foco nos pronunciamentos realizados durante o Pequeno Expediente da Câmara entre os anos de 1999 e 2014. O autor emprega o modelo de agenda expressa (*expressed agenda model*), baseado em aprendizado não supervisionado, para detectar tópicos presentes em mais de 127 mil discursos. A investigação revela que, embora o processo decisório da Câmara seja pautado majoritariamente pelas divisões entre governo e oposição, os discursos seguem uma lógica mais plural. Os deputados abordam uma diversidade de assuntos que refletem interesses pessoais, regionais ou ideológicos. Além disso, identifica-se variação

temporal na frequência de certos temas, sugerindo que a esfera discursiva possui certo grau de autonomia em relação à atividade legislativa formal.

Neste trabalho, o foco será nos discursos realizados entre os anos de 1947 e 2024. Será feita a análise exploratória desses discursos, verificando como o número de deputados variou ao logo dos anos, lideranças de estados e partidos ao longo dos anos, quantidade de discursos etc. Analisar como períodos históricos, como a ditadura militar, e crises impactaram os temas debatidos na Câmara. O ano de 2001 será amplamente analisado

Essa abordagem é viabilizada pelo avanço do poder computacional disponível atualmente e pela ampla quantidade de dados acessíveis, utilizando ferramentas como *Python* e modelagem de tópicos. A modelagem de tópicos evoluiu a partir da recuperação de informação, com o objetivo de representar documentos de forma eficiente. Técnicas iniciais como TF-IDF destacavam termos relevantes ao ponderar a frequência de palavras nos documentos e no corpus. Em seguida, métodos como o LSI passaram a utilizar decomposição matricial para extrair padrões latentes e reduzir a dimensionalidade, superando limitações do TF-IDF. O LDA introduziu uma abordagem probabilística, considerando que os documentos são combinações de tópicos representados como distribuições de palavras. Mais recentemente, o BERTopic tornou-se amplamente utilizado por empregar representações semânticas profundas e técnicas modernas de agrupamento, permitindo a extração de tópicos com maior precisão, mesmo em textos curtos ou ambíguos.

No capítulo 2, serão explorados redes neurais recorrentes, LSTM e GRU, destacando seus mecanismos de funcionamento e desafios, como o problema do gradiente desaparecendo. Além disso, será abordada a representação vetorial de palavras, incluindo técnicas como Word2Vec, GloVe e BERT, e sua aplicação em tarefas como a classificação de sentimentos. Por fim, será discutido o modelo de atenção, destacando sua importância em arquiteturas modernas de redes neurais e a modelagem de tópicos, tratando sobre as técnicas que surgiram na literatura e explicando sobre as mais utilizadas nos dias de hoje, que é o LDA e o BERTopic.

No Capítulo 3, será abordado o conjunto de dados utilizado. Inicialmente, será apresentada uma explicação sobre o funcionamento do site de origem dos discursos, detalhando o processo de pesquisa e extração dos dados. Em seguida, será descrito o algoritmo utilizado para a coleta das informações, bem como as etapas de limpeza e organização dos dados, com o objetivo de eliminar informações irrelevantes para a análise. Serão listados os problemas encontrados durante o tratamento dos dados e as soluções adotadas.

No Capítulo 4, será feita a exposição da análise exploratória dos dados do ano de 2001, verificando suas propriedades. Serão apresentados os resultados obtidos com a utilização dos modelos de modelagem de tópicos, como o LDA e o BERTopic, evidenciando os tópicos relevantes ao longo de 2001 e demonstrando que fatores externos influenciaram os temas abordados. Além disso, será mostrado que ambas as técnicas possuem os mesmos tópicos relevantes.

2 PROCESSAMENTO DE LINGUAGEM NATURAL

Uma ferramenta fundamental na análise de textos, como discursos parlamentares, é o Processamento de Linguagem Natural (PLN), uma subárea da inteligência artificial que se dedica ao estudo da interação entre a linguagem humana e os computadores. O objetivo do PLN é possibilitar que sistemas computacionais compreendam, interpretem, processem e até gerem linguagem natural de maneira eficiente. Embora o PLN, em si, não seja uma rede neural, técnicas de redes neurais, especialmente as redes neurais profundas, têm sido amplamente empregadas para aprimorar a performance de tarefas como classificação de texto, análise de sentimentos e modelagem de tópicos.

Nesse contexto, destaca-se também o papel do *machine learning* (aprendizado de máquina), que consiste em um conjunto de métodos que permitem que os computadores aprendam padrões a partir de dados, sem serem explicitamente programados para cada tarefa. No PLN, algoritmos de aprendizado de máquina são utilizados para treinar modelos capazes de reconhecer estruturas linguísticas, identificar temas, classificar documentos e realizar outras tarefas complexas com base em grandes volumes de dados textuais. A combinação entre PLN, aprendizado de máquina e redes neurais tem sido decisiva para avanços recentes na compreensão automática da linguagem humana.

2.1 Conceitos fundamentais

Uma das funcionalidades mais comuns do processamento de linguagem natural é a identificação de nomes em frases, por exemplo:

x:	Harry	Potter	e	Hermione	Granger	inventaram	uma	nova	magia
y:	1	1	0	1	1	0	0	0	0

onde x representa a entrada e y a saída. Observe que ambos possuem nove elementos, sendo que o valor 1 representa a identificação de um nome, previamente definido em uma lista, e 0 representa que a palavra não é um nome. É possível identificar as palavras de entrada utilizando $x^{<1>}$ e $x^{<2>}$, onde $x^{<1>} = \text{Harry}$ e $x^{<2>} = \text{Potter}$. De maneira similar, $y^{<1>} = 1$ e $y^{<2>} = 1$, e T_x será o tamanho da entrada, enquanto T_y será o tamanho da saída; para este exemplo, ambos serão 9.

No entanto, essa frase representa apenas um dos vetores possíveis de entrada. Normalmente, serão analisadas várias frases. Portanto, a representação de cada elemento será $x^{(i)<t>}$, que representa o elemento t do vetor de entrada i. De maneira similar, $y^{(i)<t>}$ significa o elemento t da sequência de saída do exemplo de treinamento i.

Um dos maiores desafios do PLN é determinar como representar uma palavra. Para isso, é necessário um vocabulário que contenha todas as palavras do conjunto alvo. O vocabulário é representado por uma matriz, que consideraremos como a seguinte matriz coluna com

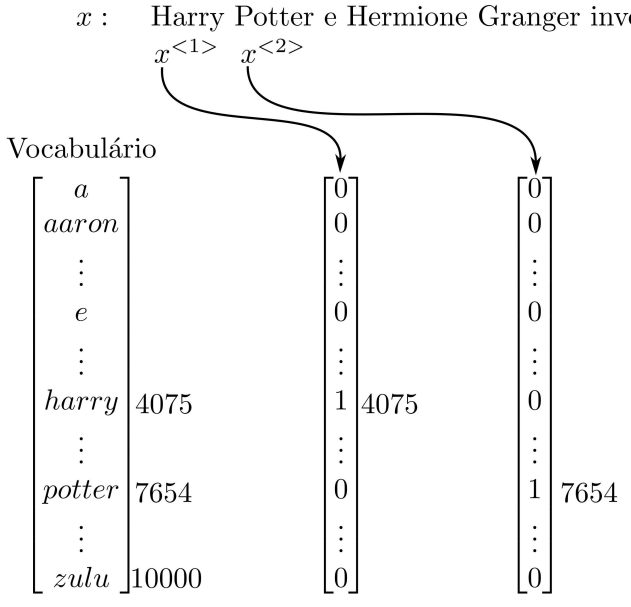
10.000 elementos dispostos em ordem alfabética:

$$\begin{bmatrix} a \\ aaron \\ \vdots \\ e \\ \vdots \\ harry \\ \vdots \\ potter \\ \vdots \\ zulu \end{bmatrix}, \tag{2.1}$$

onde cada palavra possui um índice associado a ela. Esse vocabulário pode ser construído a partir de todos os textos disponíveis, coletando as *m* palavras mais frequentes, ou pode-se simplesmente utilizar listas de palavras mais comuns na língua em questão, disponíveis na internet.

Uma vez definida essa matriz, ao fornecer uma frase como entrada, a saída para cada palavra, *y*, será uma matriz coluna composta de zeros, exceto no índice correspondente à palavra analisada. Isso pode ser exemplificado da seguinte forma:

Figura 1 – *x* representa a entrada, cada matriz coluna está associada a uma palavra da frase, caso a palavra esteja contida no vocabulário então o elemento da matriz será 1 no respectivo índice, caso contrário será 0.



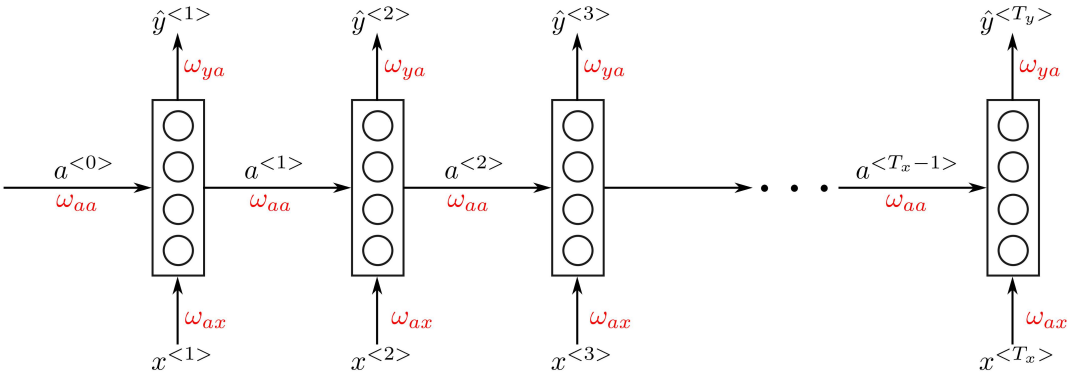
Fonte: Autor

2.1.1 Rede neural recorrente

Esse tipo de rede é amplamente utilizado em tarefas como reconhecimento de escrita [20], reconhecimento de fala [21] e processamento de linguagem natural [22].

As RNNs utilizam conexões recorrentes, onde a saída de um neurônio em um instante de tempo é realimentada como entrada para a rede no próximo instante. Isso permite que a rede capture dependências temporais e padrões dentro da sequência. O componente fundamental da rede é a unidade recorrente, que mantém um estado oculto , uma espécie de memória , atualizado a cada passo, com base na entrada atual e no estado oculto anterior.

Figura 2 – Representação visual de uma rede recorrente, aonde cada entrada representa um passo e $a^{<i>}$ representa o vetor de estado oculto, que armazena as informações dos passos anteriores



Fonte: Autor

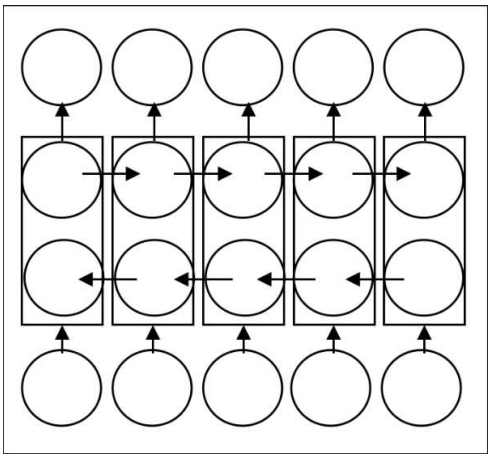
Cada retângulo representa os neurônios da rede, e cada entrada corresponde a um novo passo de tempo. A cada instante, é fornecida uma entrada $x^{<i>}$, que é processada por uma matriz de pesos W_{ax}, gerando uma saída $y^{<i>}$, governada por outra matriz de pesos W_{ya}. Além disso, existe o vetor de estado oculto $a^{<i-1>}$, que inicialmente é composto apenas por zeros, mas é atualizado a cada passo de tempo com base no aprendizado dos estados anteriores. Essa atualização é regida por uma matriz de pesos W_{aa}, que atua como a memória da rede, armazenando informações dos passos anteriores.

No entanto, essa estrutura apresenta uma limitação: a saída $y^{<3>}$ só levará em conta os dois passos anteriores, o que pode resultar em imprecisões em comparação com a saída $y^{<37>}$, por exemplo. Uma maneira de mitigar esse problema é utilizando uma rede neural recorrente bidirecional. Essa abordagem consiste em estruturar a rede com duas camadas ocultas conectadas em direções opostas para a mesma saída. Uma das camadas avança temporalmente (processamento direto), enquanto a outra retrocede no tempo (processamento reverso), permitindo que a rede capture dependências de longo alcance de forma mais eficaz.

Além da estrutura de redes neurais recorrentes bidirecionais, existem outras arquiteturas, como: Um para Um, Um para Muitos, Muitos para Um e Muitos para Muitos. Cada uma delas possui uma aplicação específica, dependendo do problema que se deseja resolver.

- **Um para Um**: Esse modelo é amplamente utilizado quando existe uma correspondência

Figura 3 – Representação visual de uma rede bidirecional, aonde cada círculo dentro do retângulo representa um neurônio e os externos representam entrada e saída

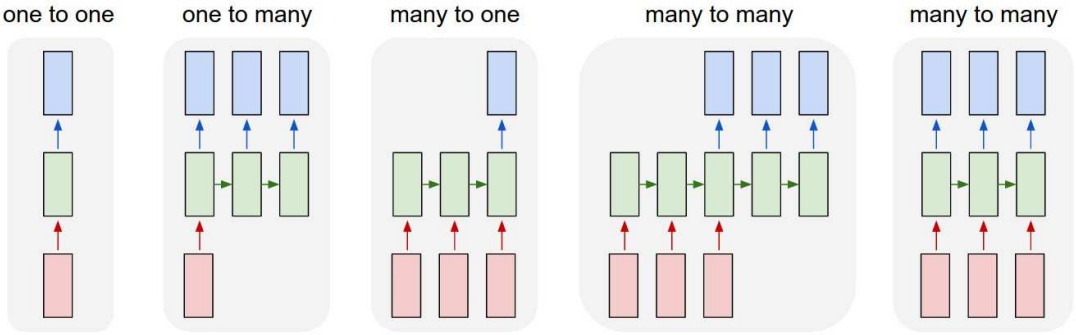


Fonte: Autor

direta entre a entrada e a saída, como em tarefas de classificação de uma única palavra ou frase.

- **Um para Muitos**: Esse tipo de rede é útil em tarefas de geração de texto ou quando se deseja descrever uma imagem. Ele gera uma sequência de saídas a partir de uma única entrada.
- **Muitos para Um**: Essa configuração é frequentemente usada em análise de sentimento, onde várias palavras de entrada são analisadas para gerar uma única saída (ex.: classificação de sentimentos).
- **Muitos para Muitos**: Nesse caso, existem múltiplas entradas e múltiplas saídas, como na tradução automática, onde se recebe uma sequência de palavras em uma língua e se gera uma sequência correspondente em outra língua. Essa arquitetura foi a utilizada até então neste trabalho.

Figura 4 – Representação visual das diferentes arquiteturas para redes recorrentes

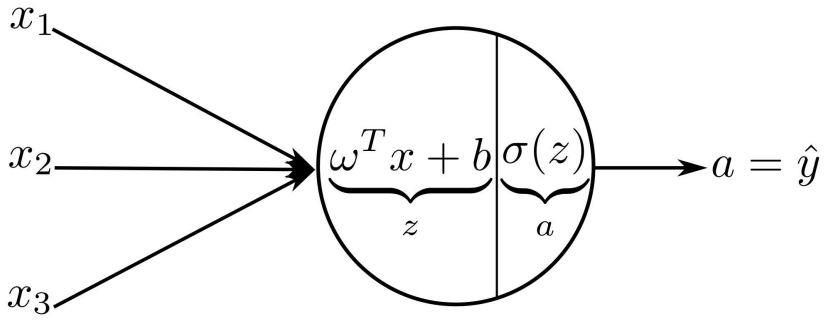


Fonte: Andrej Karpathy [23]

2.1.2 Processando a saída da rede neural (Propagação direta)

No caso em que não há uma camada oculta, o processo é simples: a camada de entrada fornece seus valores diretamente para a camada de saída. A função z, definida como $\omega^T x + b$, é calculada e, em seguida, a função de ativação, neste caso, a sigmoide, transforma esses valores em um número real. O processo está ilustrado na figura 5.

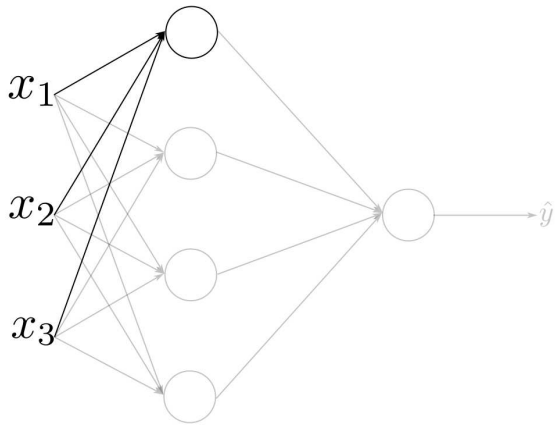
Figura 5 – Representação da propagação direta em uma rede neural simples.



Fonte: Autor

Já no caso com uma camada oculta, a situação não é tão simples. É necessário considerar cada neurônio da camada oculta individualmente, pois eles servirão como a camada de entrada para a camada posterior. Considerando somente o primeiro neurônio da camada oculta, assim como na figura 6.

Figura 6 – Uma rede neural com uma única camada oculta.



Fonte: Autor

O processo é o mesmo para o caso sem a camada oculta, inicialmente será calculado $z_1^{[1]} = \omega_1^{[1]T} x + b_1^{[1]}$, em seguida será processado pela função ativação $a_1^{[1]} = \sigma(z_1^{[1]})$, que retornará um valor real. Como estamos utilizando até então a função sigmoid como função ativação, então o valor será entre 0 e 1. Para o segundo neurônio o processo será o mesmo, primeiramente será calculado $z_2^{[1]} = \omega_2^{[1]T} x + b_2^{[1]}$, observando a devida mudança na notação, o índice subscrito indicando o neurônio sendo considerado, então a função ativação $a_2^{[1]} = \sigma(z_2^{[1]})$ irá retornar um valor real entre 0 e 1. Com todos esses valores de $a_i^{[1]}$, onde i é cada neurônio da camada 1,

uma matriz será montada e será utilizada como entrada para a camada posterior, assim como a matriz x serviu de entrada para a camada oculta.

Computacionalmente, não é eficiente calcular $z_i^{[1]}$ e $a_i^{[1]}$ com um *loop*. Quanto maior for o número de neurônios, mais ineficiente será. Portanto, torna-se necessária uma representação que contorne esse problema. O computador trabalha muito bem com matrizes e sua multiplicação, sendo recomendada a representação matricial quando o objetivo é eficiência. No caso em questão, temos oito equações, duas para cada neurônio. As equações são:

$$\begin{aligned}
z_1^{[1]} &= \omega^{[1]T}x + b_1^{[1]}, \quad a_1^{[1]} = \sigma(z_1^{[1]}) \\
z_2^{[1]} &= \omega^{[1]T}x + b_2^{[1]}, \quad a_2^{[1]} = \sigma(z_2^{[1]}) \\
z_3^{[1]} &= \omega^{[1]T}x + b_3^{[1]}, \quad a_3^{[1]} = \sigma(z_3^{[1]}) \\
z_4^{[1]} &= \omega^{[1]T}x + b_4^{[1]}, \quad a_4^{[1]} = \sigma(z_4^{[1]}),
\end{aligned}$$

é possível representar as equações de z na forma matricial da seguinte forma:

$$\begin{bmatrix} -- \omega_1^{[1]T} -- \\ -- \omega_2^{[1]T} -- \\ -- \omega_3^{[1]T} -- \\ -- \omega_4^{[1]T} -- \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ b_3^{[1]} \\ b_4^{[1]} \end{bmatrix} = \begin{bmatrix} \omega^{[1]T}x + b_1^{[1]} \\ \omega^{[1]T}x + b_2^{[1]} \\ \omega^{[1]T}x + b_3^{[1]} \\ \omega^{[1]T}x + b_4^{[1]} \end{bmatrix} = \begin{bmatrix} z_1^{[1]} \\ z_2^{[1]} \\ z_3^{[1]} \\ z_4^{[1]} \end{bmatrix}, \tag{2.2}$$

A matriz dos ω tem dimensão (Número de neurônios da camada oculta) $\times$ (Tamanho do vetor x). Para as outras quatro equações, $a_i^{[1]}$, a representação matricial será:

$$a^{[1]} = \begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \\ a_3^{[1]} \\ a_4^{[1]} \end{bmatrix} = \sigma(z^{[1]}). \tag{2.3}$$

Esse processo, no qual os dados de entrada são processados por essas equações para se tornarem a entrada para a camada de saída e, eventualmente, a saída, é chamado de propagação direta. Percebe-se que os dados sofrem mudanças baseadas nos pesos e no viés em cada neurônio de cada camada. Por conta disso, ao adicionar uma camada oculta, é possível ajustar a rede para padrões mais complexos, pois haverá uma maior quantidade de tratamento para esses dados até eventualmente obter o resultado.

2.1.3 Gradiente descendente para redes neurais rasas (backpropagation)

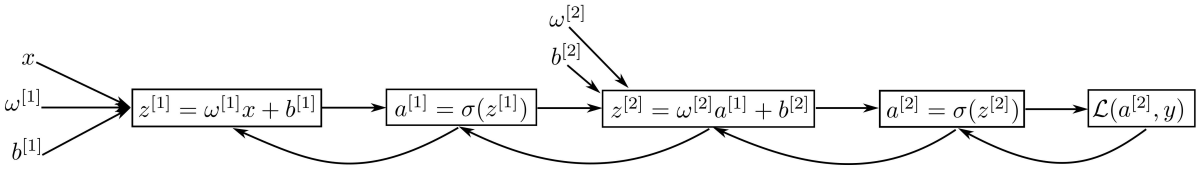
Para o caso da rede rasa a função custo vai depender de mais parâmetros, ela é:

$$J(\omega^{[1]}, b^{[1]}, \omega^{[2]}, b^{[2]}) = \frac{1}{m}\sum_{i=1}^{m} \mathscr{L}(\hat{y}, y), \tag{2.4}$$

aonde m será o número de exemplos utilizados.

Anteriormente, foi explicitado o procedimento para uma rede que não possuía camadas ocultas. O processo segue de forma análoga, e o grafo que representa a equação é dado por:

Figura 7 – Representação da retropropagação em uma rede neural rasa



Fonte: Autor

Começando da direita para a esquerda, as derivadas serão da mesma forma das que já foram demonstradas nas equações (3.2), (3.3) e (3.4) para a camada de saída. Elas serão:

$$\begin{aligned}
\frac{\partial \mathscr{L}}{\partial z^{[2]}} &= a^{[2]} - y, \\
\frac{\partial \mathscr{L}}{\partial \omega^{[2]}} &= \frac{\partial \mathscr{L}}{\partial z^{[2]}} a^{[1]T}, \\
\frac{\partial \mathscr{L}}{\partial b^{[2]}} &= \frac{\partial \mathscr{L}}{\partial z^{[2]}}.
\end{aligned}$$

Para a camada oculta, o cálculo será realizado utilizando a regra da cadeia. A derivada da função custo em relação a z é:

$$\frac{\partial \mathscr{L}}{\partial z^{[1]}} = \frac{\partial \mathscr{L}}{\partial a^{[1]}} \frac{\partial a^{[1]}}{\partial z}. \tag{2.5}$$

Não está sendo considerada nenhuma função de ativação específica, então utilizamos uma função genérica $g(z)$. Como a saída da camada oculta passa por essa função de ativação, temos:

$$\frac{\partial a^{[1]}}{\partial z} = g'(z), \tag{2.6}$$

para a sigmoide, por exemplo, $g'(z) = g(z)(1 - g(z))$.

Aplicando a regra da cadeia novamente em $\frac{\partial \mathscr{L}}{\partial a^{[1]}}$:

$$\begin{aligned}
\frac{\partial \mathscr{L}}{\partial a^{[1]}} &= \frac{\partial \mathscr{L}}{\partial z^{[2]}} \frac{\partial z^{[2]}}{\partial a^{[1]}} \\
&= \frac{\partial \mathscr{L}}{\partial z^{[2]}} \frac{\partial (\omega^{[2]T} a^{[1]} + b^{[2]})}{\partial a^{[1]}} \\
&= \frac{\partial \mathscr{L}}{\partial z^{[2]}} \omega^{[2]T},
\end{aligned}$$

substituindo:

$$\frac{\partial \mathscr{L}}{\partial z^{[1]}} = \frac{\partial \mathscr{L}}{\partial z^{[2]}} \omega^{[2]T} g'(z^{[1]}). \tag{2.7}$$

Para o peso e para o viés, as expressões seguem a mesma lógica do caso sem a camada oculta:

$$\frac{\partial \mathscr{L}}{\partial \omega^{[1]}} = \frac{\partial \mathscr{L}}{\partial z^{[1]}} x^T,$$
$$\frac{\partial \mathscr{L}}{\partial b^{[1]}} = \frac{\partial \mathscr{L}}{\partial z^{[1]}}.$$

A atualização dos pesos será feita da mesma maneira, pois o que muda com diferentes funções de ativação e camadas é o valor do gradiente, enquanto os pesos e viéses já estão determinados inicialmente. É importante observar que, ao contrário da regressão logística, os pesos devem ser inicializados de maneira aleatória. Caso todos os pesos tenham o mesmo valor, os neurônios da camada oculta serão totalmente simétricos, ou seja, irão calcular exatamente a mesma função. Ao calcular o gradiente, a atualização dos pesos também será idêntica. Como as atualizações dos pesos serão idênticas, o aprendizado será redundante, pois todos os neurônios realizarão as mesmas funções.

Quando o processo de propagação direta acaba é possível calcular a função custo, ela vai precisar ser minimizada para que o modelo possua precisão, esse processo de ajustes de peso utilizando derivadas e a atualização dos pesos é conhecida como retropropagação, essa é uma parte fundamental do treinamento, vai permitir que o modelo aprenda a partir de seus erros e melhore suas previsões ao longo do tempo.

2.1.4 Retropropagação e Propagação direta em uma rede recorrente

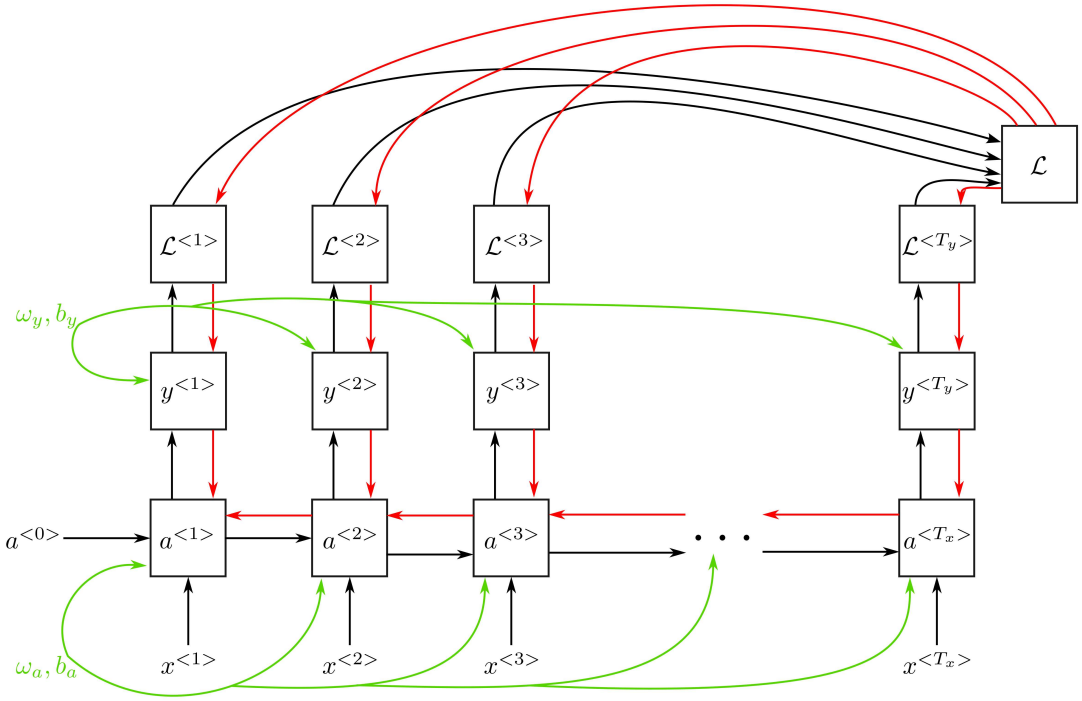
A propagação direta segue de maneira análoga aos casos já trabalhados anteriormente. A diferença é que, além das equações dependerem da entrada x, elas também dependem do vetor $a^{<i>}$. Portanto, as equações serão:

$$\begin{aligned}
a^{<1>} &= g_1(\omega_{aa} a^{<0>} + \omega_{ax} x^{<1>} + b_a),\\
y^{<1>} &= g_2(\omega_{ya} a^{<1>} + b_y),\\
a^{<t>} &= g(\omega_{aa} a^{<t-1>} + \omega_{ax} x^{<t>} + b_a),\\
y^{<t>} &= g(\omega_{ya} a^{<t>} + b_y),
\end{aligned}$$

onde $a^{<0>} = \vec{0}$. A função g é a função de ativação mais apropriada para o problema. Geralmente, para $a^{<i>}$, utilizam-se ReLU ou tanh, enquanto para a saída pode-se utilizar a sigmoide.

A retropropagação será levemente diferente do que foi tratado anteriormente, pois agora há uma relação entre cada passo de tempo devido ao vetor $a^{<i>}$. Ela ocorrerá da seguinte forma:

Figura 8 – Representação da retropropagação em uma rede neural recorrente, as setas pretas representam a propagação direta, as vermelhas representam a retropropagação e as verdes representam como estão sendo considerados os pesos e vieses. Cada entrada é considerado um passo de tempo.



Fonte: Autor

Existem vários passos de tempo. Antes da camada de saída, a mesma matriz de pesos ω_a e o mesmo viés b_a são aplicados em todos os passos. Da mesma forma, na camada de saída, o peso e o viés são compartilhados entre todos os passos. Após todo esse processo, será gerada a função erro, a mesma utilizada no caso da regressão logística. Cada passo terá sua própria função erro, e a média de todas será a função custo, dada por:

$$\mathscr{L}^{<t>}(\hat{y}^{<t>}, y^{<t>}) = -y^{<t>} \log \hat{y}^{<t>} - (1 - y^{<t>}) \log(1 - \hat{y}^{<t>}),$$
$$\mathscr{L}(\hat{y}, y) = \sum_{t=1}^{T_y} \mathscr{L}^{<t>}(\hat{y}^{<t>}, y^{<t>}).$$

As linhas vermelhas representam a retropropagação. Aplicando-se a regra da cadeia, é possível ver que a propagação percorre o passo atual totalmente para baixo e retorna também aos passos de tempo anteriores, até alcançar a primeira camada do primeiro passo. Devido a esse comportamento, esse processo é chamado de retropropagação através do tempo.

2.2 Problema do desaparecimento do gradiente

Um problema constante com redes RNNs é o desaparecimento do gradiente. Os pesos são atualizados com base na derivada parcial da função custo, e quanto maior for o

número de propagações diretas na rede, devido, por exemplo, a uma maior profundidade da rede, os gradientes dos pesos no início da rede serão calculados com um maior número de multiplicações durante a retropropagação. Consequentemente, os gradientes dos pesos mais próximos da entrada serão exponencialmente menores do que os dos pesos mais próximos da saída. Essa diferença na magnitude dos gradientes pode introduzir instabilidades no processo de treinamento, causando lentidão ou até mesmo interrompendo o treinamento [24].

É possível lidar com esse problema utilizando funções de ativação como a *leaky* ReLU, que pode ajudar a reduzir os efeitos do desaparecimento do gradiente. Além disso, técnicas de inicialização de pesos, como as de Xavier ou He [25], também podem contribuir. No entanto, a solução mais eficaz para esse problema é o uso de redes LSTM ou GRU, que serão discutidas a seguir.

2.2.1 Memória de curto longo prazo (LSTM)

LSTM é um tipo de RNN focada em mitigar o problema do desaparecimento do gradiente, ela possui uma memória de curto prazo que pode durar por milhares de passos de tempo [26]. Uma unidade LSTM é tipicamente composta por uma célula e três portas: a porta de entrada, a porta de saída [27] e uma porta de esquecimento [28]. A célula lembra de valores em um intervalo de tempo arbitrário, e os portões regulam o fluxo de informação para dentro da célula e para fora.

As portas de esquecimento decidem qual informação descartar do vetor de contexto $c^{<t>}$, baseando-se no estado anterior e na entrada atual. Um valor próximo de 1 indica que a informação será preservada, enquanto um valor próximo de 0 significa que deve ser esquecida. As portas de entrada determinam quais partes da nova informação serão adicionadas ao vetor de contexto, utilizando o mesmo sistema das portas de esquecimento. Já as portas de saída regulam qual parte do estado atual da célula será usada para gerar a saída, influenciando diretamente a informação extraída do vetor de contexto.

Emitindo informações relevantes a partir do estado atual e do vetor de contexto, a rede pode capturar dependências de longo prazo para fazer previsões tanto no estado atual quanto no futuro.

LSTM é amplamente utilizada em problemas como classificação [29], processamento de dados, reconhecimento de fala [21], traduções [30], entre outros.

2.2.2 Rede recorrente com portas (GRU)

GRU [31] é parecido com a rede LSTM, utilizando mecanismos de portas para emitir e esquecer certas características, mas não possui um vetor contexto ou uma porta de saída, resultando em menos parâmetros em comparação ao LSTM. Essa simplicidade faz com que a GRU seja mais eficiente computacionalmente, o que pode ser vantajoso em tarefas com grandes volumes de dados ou quando a velocidade de treinamento é um fator crítico.

A sua performance em certas tarefas, como PLN, modelagem de sinais de fala [32],

mostrou ter uma performance semelhante ao LSTM, fazendo dela uma alternativa competitiva, especialmente em contextos que exigem uma solução mais eficiente sem perda significativa de desempenho.

2.3 Representações vetoriais de palavras

Como mencionado anteriormente, a representação de palavras no processamento de linguagem natural é um desafio. Uma abordagem inicial foi o uso de matrizes de vocabulário, mas a representação vetorial permite capturar relações entre palavras, como a analogia entre "Rei" e "Rainha".

Uma das principais limitações da representação por vocabulário é que as palavras são tratadas como entidades isoladas, sem permitir ao algoritmo generalizar seu significado. Considere as seguintes frases:

- "Eu quero um copo de _____ de laranja"
- "Eu quero um copo de _____ de maçã"

Um modelo baseado em vocabulário poderia prever com facilidade que a palavra adequada para a primeira frase é "suco". No entanto, para a segunda, ele poderia ter dificuldades caso não tivesse sido treinado especificamente para reconhecer a relação entre "laranja" e "maçã" como frutas.

Dessa forma, uma representação mais eficiente trata as palavras como vetores, onde cada elemento do vetor representa uma característica da palavra, como ilustrado na tabela abaixo:

	Homem	Mulher	Rei	Rainha	Laranja	Maçã
Gênero	-1	1	-0.97	0.97	0.00	0.01
Realeza	0.02	0.02	0.95	0.95	-0.01	0.00
Idade	0.03	0.03	0.70	0.70	0.03	-0.02
Comida	0.01	0.01	0.02	0.01	0.95	0.97
⋮						

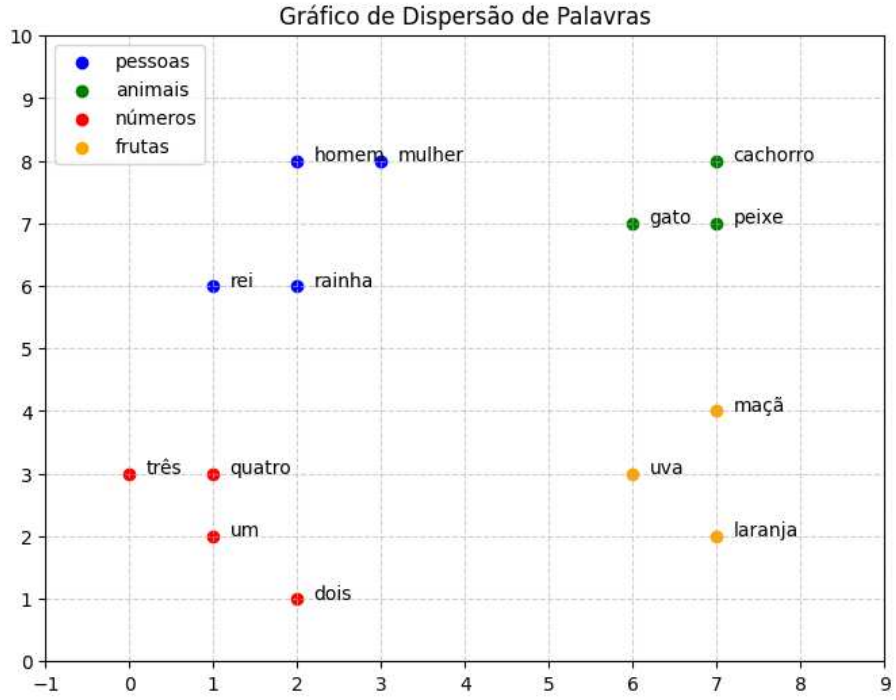
Tabela 1: Exemplo de representação vetorial de palavras

Considere que cada palavra tenha 200 características. Note que o valor associado à característica "Gênero" para "Homem" e "Mulher" são opostos, refletindo o fato de que são conceitos distintos. O mesmo ocorre para "Rei" e "Rainha". Com essa representação, um modelo pode reconhecer mais facilmente a relação entre "Laranja" e "Maçã", permitindo prever corretamente que a palavra ausente nas frases anteriores é "suco".

Outra propriedade dessa vetorização das palavras é que agora é possível estabelecer relações como "Rei $\rightarrow$ Rainha" e "Homem $\rightarrow$ Mulher". Se realizarmos a subtração vetorial entre os vetores de "Homem" e "Mulher", obtemos o vetor $(-2, 0, 0, 0)$. Da mesma forma, ao subtrair "Rei" de "Rainha", também obtemos $(-2, 0, 0, 0)$. Como ambos os vetores resultantes são paralelos, a única diferença semântica entre essas palavras deve ser o gênero.

Para visualizar essas representações vetoriais, pode-se utilizar o algoritmo t-SNE, que reduz as dimensões dos vetores para duas, tornando possível uma representação gráfica dessas relações semânticas.

Figura 9 – Representação em duas dimensões da vetorização das palavras utilizando o algoritmo t-SNE, cada cor representa um grupo de palavras e quanto mais próximos estão os pontos mais semelhantes elas são.



Fonte: Autor

Nota-se que as palavras que tem relações entre si estão mais próximas uma das outras.

2.3.1 Word2Vec

Word2vec é um dos algoritmos mais populares para criar a representação vetorial das palavras, desenvolvido em 2014 por Tomáš Mikolov [33] e seus colegas no Google. O seu treinamento é baseado em uma rede neural rasa, que aprende representações distribuídas das palavras a partir do contexto em que elas aparecem, ou seja, vale para qualquer idioma.

Existem dois métodos principais para esse treinamento: utilizando o modelo saco-de-palavras (CBOW) e o modelo salto de palavras (Skip-gram). O modelo saco-de-palavras faz a previsão baseado nas palavras ao seu redor. Por exemplo, na frase “O _____ está latindo”, o modelo pode inferir que a palavra ausente é “cachorro”. Já o modelo do salto de palavras faz o oposto, dado um termo central, ele prevê quais palavras costumam aparecer próximas a ele. Assim, se a palavra central for “cachorro”, o modelo aprenderá que palavras como “latindo”, “animal”e “pet”costumam estar no mesmo contexto. Durante o treinamento, é possível definir o intervalo de palavras que será considerado próximo à palavra central. Quanto maior o intervalo, maior será o poder computacional necessário.

Uma das propriedades mais interessantes do Word2vec é sua capacidade de capturar relações entre palavras de forma matemática, assim como o exemplo dado da relação entre "Rei"e "Rainha". Isso significa que a estrutura semântica das palavras pode ser manipulada com operações algébricas simples, permitindo aplicações como analogias automáticas e busca semântica.

2.3.2 GloVe

GloVe [34], significa *Global Vectors for Word Representation*, é o algoritmo mais simples para a vetorização das palavras. Ao contrário do Word2Vec, que se concentra no contexto local das palavras, o GloVe utiliza uma abordagem global, aproveitando uma matriz de co-ocorrência.

O funcionamento do GloVe é o seguinte: inicialmente, ele cria a matriz de co-ocorrência onde cada célula representa o número de vezes que duas palavras aparecem juntas no contexto. Por exemplo, se as palavras "Laranja"e "Suco"aparecem frequentemente no mesmo contexto, então elas terão uma co-ocorrência alta na matriz. A matriz é então fatorada em duas menores, uma representando as palavras e outra representando o contexto.

A principal técnica do GloVe é a fatoração da matriz, que tem como objetivo minimizar a diferença entre a co-ocorrência real das palavras e a co-ocorrência predita pelas representações vetoriais. O modelo utiliza a seguinte função custo:

$$J = \sum_{i=1}^{V} \sum_{j=1}^{V} f(P_{ij}) \left(w_i^T w_j + b_i + b_j - \log P_{ij} \right)^2, \tag{2.8}$$

onde V é o número de palavras no vocabulário, ω_i e ω_j são os vetores de palavras para as palavras i e j, b_i e b_j são os termos de viés para as palavras i e j, P_{ij} é o número de vezes que as palavras i e j aparecem juntas (a co-ocorrência), e $f(P_{ij})$ é uma função de pesagem aplicada à co-ocorrência. Essa função ajuda a atenuar as ocorrências que são altas, para evitar que elas dominem o treinamento. Essa função é otimizada ajustando os vetores das palavras para capturar as relações mais precisas entre elas.

O GloVe é computacionalmente mais eficiente em comparação ao Word2Vec, especialmente em textos maiores. Isso se deve ao fato de que o modelo não precisa processar cada par de palavras individualmente, mas pode trabalhar com a matriz como um todo. Além disso, o GloVe também é capaz de realizar operações algébricas de maneira semelhante ao Word2Vec, permitindo a criação de analogias e relações semânticas entre palavras.

Embora o GloVe tenha essas vantagens, ele também pode ser sensível ao viés presente nos dados, o que pode afetar as representações das palavras, refletindo estereótipos e preconceitos presentes no corpus de treinamento. Mesmo assim, ele continua sendo uma técnica poderosa para a criação de vetores de palavras, especialmente quando se lida com grandes volumes de dados.

2.3.3 BERT

O BERT (Bidirectional Encoder Representations from Transformers) é um modelo de linguagem desenvolvido pelo Google com o objetivo de melhorar a compreensão da linguagem natural por parte de sistemas de inteligência artificial. Ele é baseado na arquitetura dos *Transformers*, que permite o processamento de textos inteiros de maneira eficiente, levando em conta o contexto das palavras dentro de uma frase ou parágrafo.

Diferentemente de modelos anteriores, que liam os textos apenas da esquerda para a direita (ou vice-versa), o BERT é bidirecional. Isso significa que ele analisa cada palavra levando em consideração o que vem antes e depois dela, capturando de forma mais precisa o significado contextual. Por exemplo, a palavra "banco" em "sentei no banco" e em "fui ao banco sacar dinheiro" será interpretada de maneiras diferentes, pois o modelo entende o que está ao redor da palavra.

O BERT foi treinado com uma enorme quantidade de textos, aprendendo padrões e relações entre palavras. Durante o treinamento, ele teve que resolver tarefas como prever palavras escondidas em frases ou identificar se uma frase segue logicamente outra. Isso o tornou extremamente eficaz em tarefas de compreensão de linguagem, como responder perguntas, classificar textos, traduzir sentimentos e muito mais.

Uma das grandes vantagens do BERT é que ele pode ser ajustado para tarefas específicas com poucos dados adicionais. Assim, pesquisadores e desenvolvedores podem utilizar o modelo pré-treinado e adaptá-lo para aplicações como análise de sentimentos, resumo de textos, detecção de entidades e outros usos na área de processamento de linguagem natural (PLN).

2.3.4 Classificação de sentimento

A classificação de sentimento é o processo de identificar se o texto tem uma *review* positiva ou negativa. É muito útil em PLN e é usado em várias aplicações, por exemplo:

Figura 10 – x representa a entrada e y a saída, onde as estrelas indicam a classificação de acordo com os sentimentos expressos em cada avaliação.

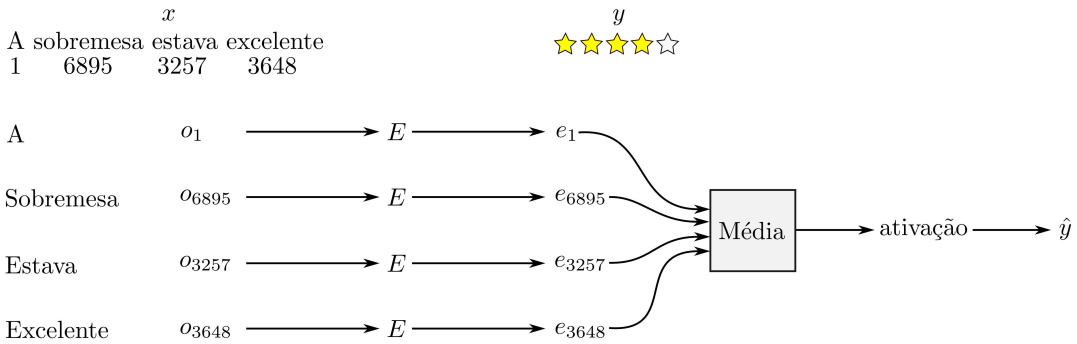
x	y
A sobremesa estava excelente	★★★★☆
Os garçons eram lentos	★★☆☆☆
Bom para lanchar, mas nada especial	★★★☆☆
O restaurante tem falta de: bons modos, boa comida, ambiente aceitável	★☆☆☆☆

Fonte: Autor

Um dos desafios é que talvez não haja dados grandes o suficiente para o treinamento supervisionado. A representação vetorial das palavras pode ajudar nisso. É possível montar um modelo que funcione da seguinte forma:

Onde E é a matriz formada pelas palavras vetorizadas, sendo que cada coluna é uma palavra, as linhas são as características consideradas e cada elemento é um número associado,

Figura 11 – Os números abaixo de cada palavra representa o número da coluna na matriz E, cada palavra é analisada individualmente e a média total é considerada como entrada para a função ativação



Fonte: Autor

entre -1 e 1, assim como foi exposto na tabela 1. Um problema desse modelo simples é que, ao analisar uma frase como "O restaurante tem falta de: bons modos, boa comida, ambiente aceitável", ele pode interpretar "bom"e "aceitável"como sentimentos positivos. No entanto, ao ler a frase como um todo, é possível notar que o sentimento é negativo. Uma maneira de resolver esse problema é utilizando RNNs. Dessa forma, a frase é analisada em sequência, e, caso o mesmo exemplo seja utilizado, o vetor $a^{<i>}$ permitirá que a rede associe o "falta de"com a palavra "bom", levando em conta o contexto geral da frase.

2.4 Modelo de atenção

O modelo de atenção foi desenvolvido por Bahdanau [35] para a tradução de sentenças. Esse é um método que determina a importância relativa de cada componente em uma sequência em relação a outras componentes da mesma sequência. Em PLN, essa importância é representada através de pesos suaves [36], que são designados para cada palavra daquela sequência. De forma geral, a atenção codifica vetores chamados tokens de vetorização ao longo de uma sequência de largura fixa, que pode variar de dezenas a milhões de tokens em tamanho.

Um *token* é uma unidade discreta de texto, que pode ser uma palavra, parte de uma palavra, ou até mesmo um caractere, dependendo da forma de tokenização utilizada. No contexto do processamento de linguagem natural (PLN), a tokenização é o processo de dividir um texto em tokens, que são usados como entradas para modelos de machine learning.

Ao contrário dos pesos tradicionais, que são calculados no processo de retropropagação, os pesos suaves só existem na propagação direta, portanto, mudam a cada passo da entrada. Inicialmente, o mecanismo de atenção utilizava uma série de RNNs em sistemas de tradução, mas recentemente tem-se utilizado transformadores, desenvolvidos por oito pesquisadores do Google, utilizando o mecanismo de atenção proposto no artigo "*Attention is All You Need*"[37], que removeram a lenta sequência de RNNs e se apoiam mais em um esquema de atenção paralela.

2.5 Modelagem de Tópicos

2.5.1 TF-IDF

A modelagem de tópicos surgiu como um desdobramento natural dentro da área de recuperação de informação (*Information Retrieval*), uma vez que os pesquisadores buscavam maneiras eficazes de representar e organizar grandes volumes de documentos. Uma das primeiras técnicas desenvolvidas foi a TF-IDF (*term frequency–inverse document frequency*), baseada no trabalho de Spärck Jones [38], que introduziu o conceito de IDF. Ela definiu que a especificidade de um termo pode ser expressa como uma função inversa da quantidade de documentos em que ele aparece. Por exemplo, considerando as 37 peças de Shakespeare:

Word	**df**	**idf**
Romeo	1	1.57
salad	2	1.27
Falstaff	4	0.967
forest	12	0.489
battle	21	0.246
wit	34	0.037
fool	36	0.012
good	37	0
sweet	37	0

Tabela 2: Exemplo de valores de *df* (document frequency) e *idf* (inverse document frequency) para diferentes palavras.

O IDF é calculado da seguinte forma:

$$idf(t) = \ln\left(\frac{N}{df_t}\right), \tag{2.9}$$

em que N representa o número total de documentos e df_t o número de documentos em que o termo t aparece. Quanto maior o valor de IDF, mais relevante é a palavra para o corpus. Por exemplo, as palavras *Romeo* e *salad* possuem valores altos de IDF, o que indica que são mais informativas sobre o conteúdo dos documentos do que palavras como *good* ou *sweet*, que aparecem em praticamente todos os textos.

A métrica TF-IDF utiliza os valores de IDF como um fator de ponderação, conforme a fórmula:

$$tfidf(t,d) = tf(t,d) \cdot idf(t), \tag{2.10}$$

em que $tf(t,d)$ representa a frequência do termo t no documento d. Essa frequência pode ser calculada de várias maneiras, sendo uma das mais simples o número absoluto de ocorrências da palavra no documento. Utilizando esse peso, é possível associar uma importância relativa às palavras, desconsiderando aquelas que aparecem com muita frequência e, portanto, são menos informativas. Dessa maneira, reduzimos documentos de tamanhos variados para uma única matriz, em que as linhas representam as palavras, as colunas os documentos e os valores correspondem ao TF-IDF.

2.5.2 LSI

Embora o TF-IDF possua vantagens claras, como representar palavras relevantes para os documentos, essa técnica ainda apresenta certas limitações. Ainda são necessários muitos valores para descrever completamente um documento e, mesmo assim, não se obtém muitas informações sobre a estrutura estatística dos textos.

Para contornar essas limitações, foi desenvolvido o método de *Latent Semantic Indexing* (LSI) [39], que utiliza a técnica chamada *Singular Value Decomposition* (SVD), descrita pela seguinte equação:

$$A = U\Sigma V^T, \tag{2.11}$$

onde U é uma matriz $m \times k$, em que m é o número de palavras e k é o número de componentes latentes (ou dimensões) retidas após a decomposição. Componentes latentes são dimensões ocultas, que não são observadas diretamente nos dados originais, mas representam padrões subjacentes ou estruturas essenciais que explicam a variabilidade ou correlações presentes nos dados. Cada componente latente é formada por combinações lineares das variáveis originais.

Essa matriz U relaciona os termos a esses componentes latentes, que representam padrões matemáticos extraídos dos dados, mas que nem sempre correspondem a tópicos semanticamente interpretáveis.

A matriz Σ é diagonal e contém os valores singulares, que indicam a importância relativa de cada componente latente. Já V^T relaciona os documentos com esses mesmos componentes latentes. A matriz A representa a matriz original construída com o método TF-IDF.

A decomposição é realizada diretamente por métodos de álgebra linear, que calculam U, Σ e V^T sem processos iterativos de ajuste ou aprendizado. Dessa forma, U passa a representar uma associação entre palavras e componentes latentes, possibilitando uma redução dimensional e a extração de padrões relevantes nos dados.

Com essa técnica, é possível capturar certos aspectos da linguagem, como sinônimos e algumas formas de polissemia. No entanto, ainda há limitações, especialmente ao lidar com palavras polissemânticas, ou seja, palavras com múltiplos significados, uma vez que o modelo não considera diretamente o contexto em que os termos ocorrem.

Por exemplo, a palavra “banco” pode assumir tanto o sentido financeiro quanto o de assento. O modelo tende a agrupar palavras que coocorrem com “banco”, sugerindo que tratam do mesmo tópico. Assim, “banco” poderá ser associado simultaneamente a termos relacionados a instituições financeiras e a mobiliário. No entanto, sua representação vetorial será única, o que leva à ambiguidade e compromete a precisão semântica.

Além disso, ao lidar com matrizes extensas, o custo computacional da decomposição pode se tornar elevado. Por fim, um dos principais limites do modelo é que ele fornece apenas uma representação estatística dos textos, baseada em combinações lineares, sem incorporar uma modelagem probabilística ou uma representação semântica mais rica e contextualizada.

2.5.3 LDA

Um modelo que lida com os principais problemas citados até então é o *Latent Dirichlet Allocation* (LDA) [40]. Trata-se de um modelo generativo probabilístico para corpora de texto. O principal objetivo do LDA é identificar tópicos latentes presentes nos documentos, onde um tópico é definido como uma distribuição de probabilidade sobre um vocabulário fixo, ou seja, uma distribuição de palavras.

O LDA assume que cada documento é gerado a partir de uma mistura de tópicos, e que cada tópico, por sua vez, é uma distribuição sobre palavras. Assim, os documentos são modelados como combinações aleatórias desses tópicos latentes. A tarefa do LDA é inferir, a partir dos dados observados (as palavras dos documentos), tanto a distribuição de tópicos por documento quanto a distribuição de palavras por tópico.

Explicando de forma conceitual, ao analisar um documento, definido de antemão que existem $K = 3$ tópicos, o modelo atribui aleatoriamente tópicos para todas as palavras. Após isso, ele ajusta essas atribuições iterativamente. Com base nisso, aumenta a probabilidade de que essas palavras sejam atribuídas ao mesmo tópico latente. Esse processo se repete até atingir convergência estatística, ou seja, quando novas iterações não alteram de maneira significativa as distribuições.

Nota-se que o LDA não fornece tópicos determinísticos , todos os documentos são misturas de tópicos. Por conta da natureza aleatória da escolha dos tópicos para cada palavra, não há garantia de que todos os tópicos presentes nos textos serão identificados. Tópicos mais sutis, por exemplo, podem não ser detectados. Para lidar com esse problema, torna-se necessário aplicar o modelo com diferentes parâmetros e analisar seus resultados.

Suponha o seguinte exemplo com 5 documentos:

- D1: “O governo aprovou novas leis fiscais no congresso”
- D2: “O time venceu o campeonato após uma final emocionante”
- D3: “A economia cresceu 3% este trimestre, segundo o banco central”
- D4: “O jogador foi premiado como o melhor atacante da temporada”
- D5: “O ministro da fazenda discutiu reformas tributárias”

Note que o corpus envolve tópicos diversos. Ao aplicar o modelo LDA com $K = 3$, o modelo não conhece previamente a real distribuição de tópicos, apenas analisa padrões estatísticos. Inicialmente, atribui-se tópicos aleatórios para cada palavra , por exemplo, “governo” $\rightarrow$ tópico 1, “campeonato” $\rightarrow$ tópico 2, e assim por diante. Com o tempo, o modelo percebe que palavras como “governo”, “congresso”, “ministro” aparecem frequentemente em documentos semelhantes, sendo então agrupadas em um mesmo tópico. Palavras como “time”, “campeonato”, “jogador”, “temporada” são agrupadas em outro, e assim por diante.

Ao final, pode-se obter a seguinte distribuição:

- D1: 70% Tópico 1, 30% Tópico 3
- D2: 100% Tópico 2
- D3: 90% Tópico 3, 10% Tópico 1
- D4: 100% Tópico 2
- D5: 60% Tópico 1, 40% Tópico 3

Com os tópicos estimados como:

- Tópico 1 (Política): {“governo”, “ministro”, “leis”, “reformas”, “congresso”}
- Tópico 2 (Esportes): {“jogador”, “campeonato”, “time”, “temporada”, “final”}
- Tópico 3 (Economia): {“economia”, “banco”, “fiscal”, “cresceu”, “tributárias”}

O modelo LDA não é limitado à identificação de tópicos em texto. Ele pode ser aplicado em qualquer cenário onde haja uma estrutura latente a ser descoberta. Por exemplo, pode ser utilizado em processamento de imagens faciais, onde os elementos latentes poderiam ser nariz, boca, olhos etc.

2.5.4 BERTopic

O BERTopic [41] é um modelo mais atual e comumente utilizado na literatura nos dias de hoje. Seu diferencial em relação a outros métodos está no uso de representações semânticas avançadas dos textos, baseadas em modelos de linguagem profunda, como o BERT (Bidirectional Encoder Representations from Transformers).

Primeiramente, o BERTopic transforma os documentos, utilizando o BERT citado anteriormente, em vetores numéricos que representam seus significados de forma densa e contextualizada. Não se trata de uma contagem de palavras: esse vetor possui a capacidade de capturar nuances semânticas, como no exemplo de “Rei” e “Homem”. Após a vetorização dos documentos, o BERTopic organiza esses vetores por temas. No entanto, como os vetores possuem alta dimensionalidade, o agrupamento e a visualização se tornam complicados. Portanto, é necessário o uso de um algoritmo de redução de dimensionalidade.

O algoritmo utilizado pelo BERTopic é o “UMAP”. Esse algoritmo projeta os vetores em uma dimensão menor, preservando sua estrutura semântica. Com essa redução, torna-se possível identificar com mais facilidade os padrões de proximidade entre os discursos, ou seja, textos semelhantes permanecem próximos nesse espaço.

Com os discursos agora em baixa dimensão, o BERTopic aplica o algoritmo de clusterização denominado “HDBSCAN”. Essa técnica agrupa documentos que estão densamente distribuídos no espaço vetorial, ou seja, semanticamente semelhantes. Um dos pontos fortes do HDBSCAN é que não é necessário definir previamente quantos grupos (tópicos) devem ser encontrados, ao contrário do LDA, ele os determina automaticamente. Além disso, ele

permite que documentos muito diferentes de todos os demais sejam descartados (considerados ruído), o que melhora a qualidade dos tópicos gerados.

Após essa etapa, o BERTopic interpreta cada grupo, ou seja, descreve do que aquele conjunto de textos trata. Para isso, utiliza-se o "c-TF-IDF", uma técnica que considera cada grupo como se fosse um único "documento" e calcula as palavras mais representativas daquele conjunto, comparando com o vocabulário de todos os grupos. O resultado final é uma lista de palavras-chave que resume o conteúdo de cada tópico. Por exemplo, se um grupo contém textos que falam sobre filmes de terror, as palavras "suspense", "assustador" e "monstro" podem surgir como as mais relevantes.

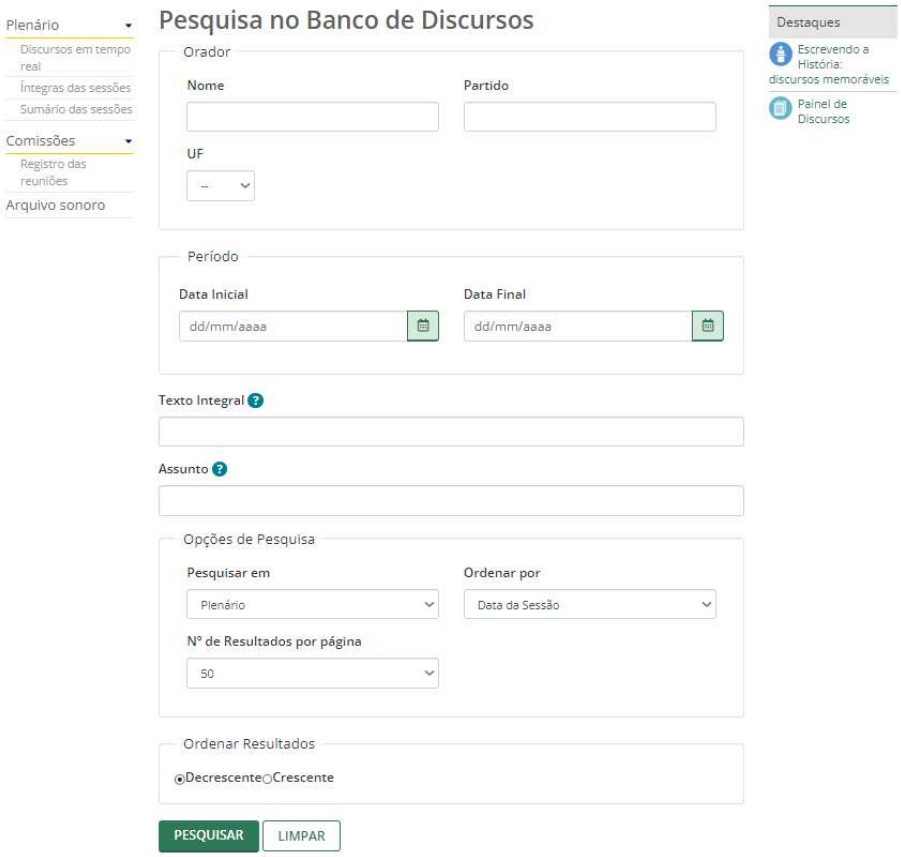
Como trabalha com modelos pré-treinados e técnicas robustas de agrupamento, o BERTopic consegue lidar bem com textos curtos, ambíguos ou que utilizam sinônimos, cenários nos quais métodos como o LDA frequentemente falham.

3 METODOLOGIA DE COLETA E PROCESSAMENTO DOS DADOS

3.1 Coleta dos dados

Os dados estavam disponíveis no site [42] da Câmara dos Deputados, onde é possível pesquisar os discursos por nome dos deputados, partido ou período de tempo desejado, como indicado na figura 12. Como o objetivo era obter a maior quantidade de discursos possível, a pesquisa foi realizada considerando apenas os períodos de tempo. O maior intervalo disponível ia de 1º de fevereiro de 1946 até a 2024 de forma crescente para que a coleta de dados seja do mais antigo até o mais atual, para permitir uma melhor análise da evolução dos discursos ao longo do tempo.

Figura 12 – Captura de tela do site da Câmara de Deputados com suas opções de filtragem na pesquisa



Fonte: Câmara dos Deputados [42]

Após selecionar o período desejado, uma nova página é carregada com os discursos organizados em uma tabela. Como pode-se ver abaixo, existem oito colunas na tabela, que são: Data, que indica o dia em que o discurso foi realizado; Sessão do Plenário, que informa o número da sessão; Fase do Plenário, que indica em qual etapa do Plenário o discurso foi proferido; Discurso, que encaminha para uma nova página contendo o texto transcrito do discurso; Sumário, que possui um ícone que expande a linha da tabela e apresenta, de forma resumida, o

conteúdo do discurso; Orador, que exibe o nome do autor acompanhado de seu respectivo partido e estado; Hora, que indica o horário em que o discurso foi adicionado ao banco de dados; e, por fim, Publicação.

Figura 13 – Captura de tela dos discursos selecionados a partir do filtro

Data	Sessão	Fase	Discurso	Sumário	Orador	Hora	Publicação
01/12/2000	222.2.51.O	PEQUENO EXPEDIENTE	□	□	ARTHUR VIRGÍLIO, PSDB-AM	*	DCD02/12/2000 PAG. 63964 □
20/12/2001	003.5.51.N	BREVES COMUNICAÇÕES	□	□	AGNALDO MUNIZ, PPS-RO	11h00	DCN21/12/2001 PAG. 22167 □
20/12/2001	003.5.51.N	ORDEM DO DIA	□	□	AIRTON CASCAVEL, PPS-RR	11h00	DCN21/12/2001 PAG. 22209 □
20/12/2001	003.5.51.N	BREVES COMUNICAÇÕES	□	□	AIRTON CASCAVEL, PPS-RR	11h00	DCN21/12/2001 PAG. 22169 □
20/12/2001	003.5.51.N	ORDEM DO DIA	□	□	AIRTON CASCAVEL, PPS-RR	11h00	DCN21/12/2001 PAG. 22214 □
20/12/2001	003.5.51.N	ORDEM DO DIA	□	□	AIRTON CASCAVEL, PPS-RR	11h00	DCN21/12/2001 PAG. 22229 □
20/12/2001	003.5.51.N	BREVES COMUNICAÇÕES	□	□	ALBERTO FRAGA, PMDB-DF	11h00	DCN21/12/2001 PAG. 22161 □
20/12/2001	003.5.51.N	ORDEM DO DIA	□	□	ALBERTO GOLDMAN, PSDB-SP	11h00	DCN21/12/2001 PAG. 22204 □
20/12/2001	003.5.51.N	ABERTURA	□	□	ALBERTO GOLDMAN, PSDB-SP	11h00	DCN21/12/2001 PAG. 22207 □
20/12/2001	003.5.51.N	ORDEM DO DIA	□	□	ALBERTO GOLDMAN, PSDB-SP	11h00	DCN21/12/2001 PAG. 22212 □
20/12/2001	003.5.51.N	ORDEM DO DIA	□	□	ANIVALDO VALE, PSDB-PA	11h00	DCN21/12/2001 PAG. 22216 □
20/12/2001	003.5.51.N	ORDEM DO DIA	□	□	ANIVALDO VALE, PSDB-PA	11h00	DCN21/12/2001 PAG. 22232 □

Fonte: Câmara dos Deputados [42]

O site apresenta limitações técnicas que tornam inviável a coleta de dados quando o período escolhido contém muitos discursos, devido ao longo tempo de carregamento das páginas. Para contornar esse problema, a coleta foi realizada em intervalos menores, limitados a 15 anos. Além disso, a URL do site possuía uma variável que permitia a expansão da quantidade máxima de discursos exibidos por página. Alterando essa variável, foi possível aumentar o limite original de 50 para 500 discursos por página, tornando a coleta de dados significativamente mais eficiente.

Utilizando a linguagem de programação *Python* e a biblioteca *Selenium*, foi desenvolvido um algoritmo para extrair as informações de data, sessão, fase, sumário e orador diretamente do HTML da página. Para a extração do sumário, o algoritmo expande todas as linhas da tabela e copia o conteúdo exibido. Com todas as informações coletadas, uma tabela é criada no *Python*, sendo que a informação da coluna "orador" é dividida em duas: uma coluna para o nome do autor e outra para o partido juntamente com o estado. Após a formatação, a tabela é exportada no formato *CSV*. Outra forma de extração desses dados foi realizada por Emanuelle Nascimento [43], ao invés de utilizar a biblioteca *Selenium* para a automatização da extração através de interações no navegador, foi utilizado a biblioteca *B4S*, que lida diretamente com o HTML da página.

3.2 Tratamento dos dados

Após a coleta de dados, foram gerados quatro arquivos *CSV*, cada um correspondendo a um intervalo específico. O primeiro abrange o período de 1º de fevereiro de 1946 até 31 de dezembro de 1963. O segundo cobre de 1º de janeiro de 1964 até 15 de novembro de 1985, escolhendo esse intervalo para permitir uma análise detalhada do período da ditadura militar. O terceiro compreende de 16 de novembro de 1985 até 31 de dezembro de 2000, e o último, de 1º de janeiro de 2001 até 10 de novembro de 2024. Com esses arquivos, foi realizada a concatenação utilizando *Python*, mesma linguagem utilizada para toda a limpeza e organização dos dados.

Por conta de quão antigos são os discursos, existem alguns erros de escrita: os nomes dos autores e dos partidos apresentam, por vezes, acentos errados ou ausentes, além de erros de digitação. Ademais, o padrão de escrita para o partido na coluna dos nomes não estava bem estabelecido. O correto seria: John Doe, JANE-CE, de acordo com a forma utilizada nos últimos 20 anos. Contudo, em alguns períodos, os estados eram separados por vírgulas, ou o partido aparecia com apenas a primeira letra maiúscula.

Para resolver o problema, foram removidos todos os acentos dos nomes dos autores, e as letras foram convertidas para minúsculas. Os partidos foram separados dos nomes dos autores, e as vírgulas que indicavam os estados foram substituídas por hifens, adequando-se ao padrão estabelecido. Os discursos começaram a ser adicionados ao site em outubro de 2000, sendo aproximadamente 500 mil discursos inseridos retroativamente a essa data.

Por conta disso, ocorreram alguns erros na associação de partidos aos autores. Deputados que atuaram durante a ditadura, por exemplo, foram, em alguns casos, vinculados a partidos aos quais se associaram após 1988, resultando na inclusão de partidos inexistentes à época nos dados referentes às décadas de 60 e 80. Como esses casos afetaram apenas um número limitado de autores, foi necessária uma pesquisa individual para corrigir essas associações.

No Plenário, diversas pessoas podem discursar, como professores universitários, psiquiatras, deputados de outros países e até ministros. No entanto, este trabalho se concentra exclusivamente nos discursos de deputados brasileiros ao longo dos anos, desconsiderando aqueles realizados por outros oradores. Além disso, a fase “Homenagem” do Plenário foi descartada, pois não está relacionada às proposições legislativas.

Após padronizar os discursos, é feito um tratamento para diminuir o número de palavras únicas nos textos, visto que cada palavra será processada individualmente. Ao reduzir essa quantidade, o tempo de processamento para cada discurso torna-se menor. Para isso, é utilizado o algoritmo conhecido como “lematização”, que determina o lema da palavra. Por exemplo, o verbo “andar” pode ser flexionado como “andou”, “andando” e “anda”. Cada uma dessas formas é considerada uma palavra distinta, o que acarreta maior tempo de processamento. Entretanto, o significado transmitido é o mesmo, de modo que, ao utilizar a lematização, essas palavras serão reduzidas ao seu lema. Esse algoritmo não se limita apenas a verbos, pois

também processa substantivos, artigos e adjetivos. Segue o exemplo:

- **Frase original (não lematizada):** Os deputados falaram sobre os novos projetos, discutindo e votaram as propostas.
- **Frase lematizada:** O deputado falar sobre o novo projeto, discutir e votar a proposta.

A aplicação desse algoritmo depende de uma consulta a um dicionário, ou seja, uma lista de palavras com suas respectivas formas flexionadas. Dessa forma, é realizada uma busca nesse dicionário para cada palavra. Uma vantagem que a lematização possui é que não há perda do contexto da frase, embora essa consulta individual aumente o tempo de processamento.

Após a lematização, é feito o processo de eliminação de palavras vazias, ou *stop words*, que são palavras que não carregam nenhum tipo de significado semântico. Essa lista de palavras vazias é personalizada dependendo dos dados e do propósito. No caso dos discursos parlamentares, por exemplo, palavras como “Deputado(a)”, “Deputados(as)”, “Senhores(as)”, “Senhor(a)” etc., que são formas de tratamento comuns em discursos, não carregam significado relevante, portanto seriam apenas processadas e aumentariam o tempo de execução individual de cada discurso. Assim, os discursos ficam da seguinte forma:

- **Antes da remoção das stop words:**
  “Senhor Presidente, nobres Deputados, venho hoje aqui falar sobre a importância do investimento em educação no nosso país.”
- **Após a remoção das stop words:**
  “falar importância investimento educação país”

Mesmo após a remoção das palavras vazias, é possível compreender o significado geral da frase, evidenciando que não há uma perda substancial de contexto semântico.

4 RESULTADOS

No capítulo passado, foi discutido o funcionamento do processamento de linguagem natural, da vetorização de palavras e da modelagem de tópicos. No capítulo atual, será apresentada a análise exploratória considerando todo o país, ao contrário do que foi feito por Wendel Costa [44] que realizou a analíse exploratória considerando os partidos do estado do Ceará, realizada sobre o conjunto de dados, destacando peculiaridades observadas, quantidades relevantes e se os resultados obtidos refletem expectativas com base em conhecimento prévio. Serão utilizados os modelos descritos anteriormente para analisar os discursos e avaliar sua precisão. Primeiramente, serão apresentados os resultados obtidos com um modelo consolidado na literatura há 22 anos, o *Latent Dirichlet Allocation* (LDA), seguido pela aplicação de um modelo mais recente, o *BERTopic*.

4.1 O Dado

Após a limpeza, os dados passaram a conter 936.460 discursos ao longo de 78 anos. Durante esse período, foram registrados 57 partidos e 6.555 deputados.

Partido	Deputados	Partido	Deputados	Partido	Deputados
ARENA	753	MDB	585	PMDB	1325
AVANTE	19	MTR	7	PRN	90
DEM	182	NOVO	10	PRONA	11
ED	1	PAN	3	PROS	56
PCB	32	PCDOB	111	PRP	30
PDC	106	PDS	511	PRS	8
PDT	409	PEN	11	PRT	13
PFL	641	PHS	24	PRTB	4
PL	347	PMB	26	PSB	309
PMN	33	PODE	52	PSC	101
PP	394	PPB	184	PSD	721
PPS	117	PR	205	PSDB	625
PRB	82	PRD	6	PSL	83
PTR	33	PT	607	PSOL	33
PV	70	PTB	679	PSP	107
REDE	9	PTN	47	PST	63
SOLIDARIEDADE	68	PTDOB	10	PSTU	12
UDN	315	SEM PARTIDO	117	UNIÃO	110
REP	84	REPUBLICANOS	84	—	—

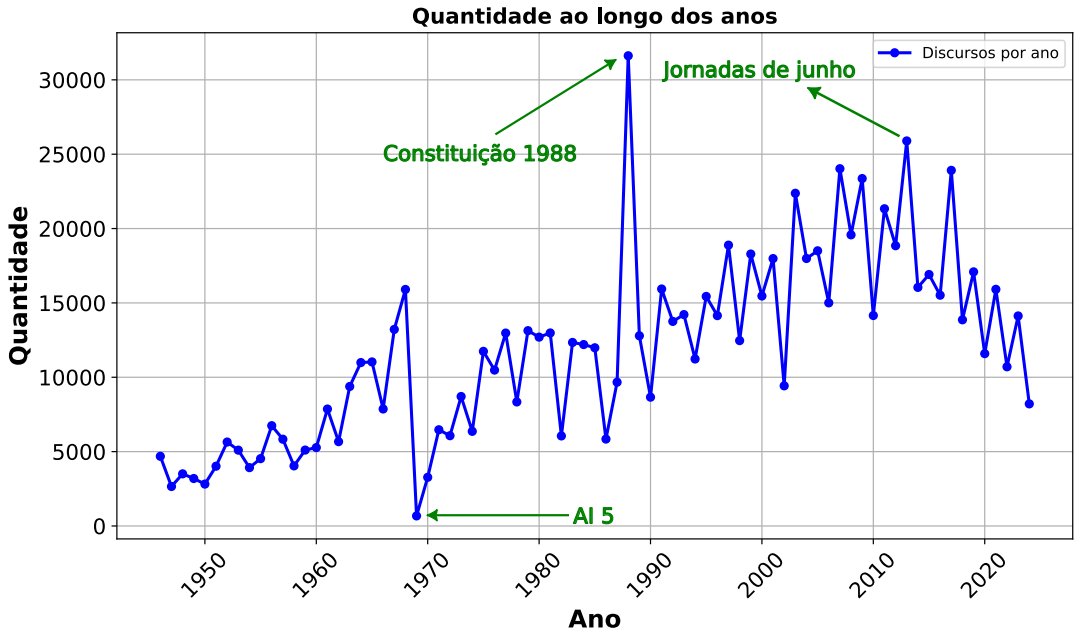
Tabela 3: Distribuição de deputados por partido.

A tabela 3 apresenta a distribuição de deputados em todos os partidos ao longo dos anos. É possível perceber que existe uma participação massiva de determinados partidos, enquanto, em contrapartida, há partidos com uma participação ínfima. Vale destacar que esta tabela não inclui todos os partidos que existiram ao longo dos anos. Por exemplo, o Partido da Juventude (PJ) não aparece no banco de dados com nenhum deputado tendo discurso durante os

períodos analisados, enquanto o Partido da Reconstrução Nacional (PRN), que anteriormente era o PJ, está presente. Isso reflete a dinâmica política e as mudanças nos nomes e nas siglas dos partidos ao longo do tempo.

Além disso, observa-se uma peculiaridade na distribuição dos discursos ao longo dos anos. Em anos eleitorais, há um aumento na quantidade de discursos, refletindo a intensificação das atividades parlamentares nesse período. Eventos históricos e políticos também deixam suas marcas na quantidade de discursos. Por exemplo, em 1969, nota-se uma queda significativa devido à instauração do AI-5 durante a ditadura militar.

Figura 14 – Quantidade de discursos ao longo dos anos.



Fonte: Autor

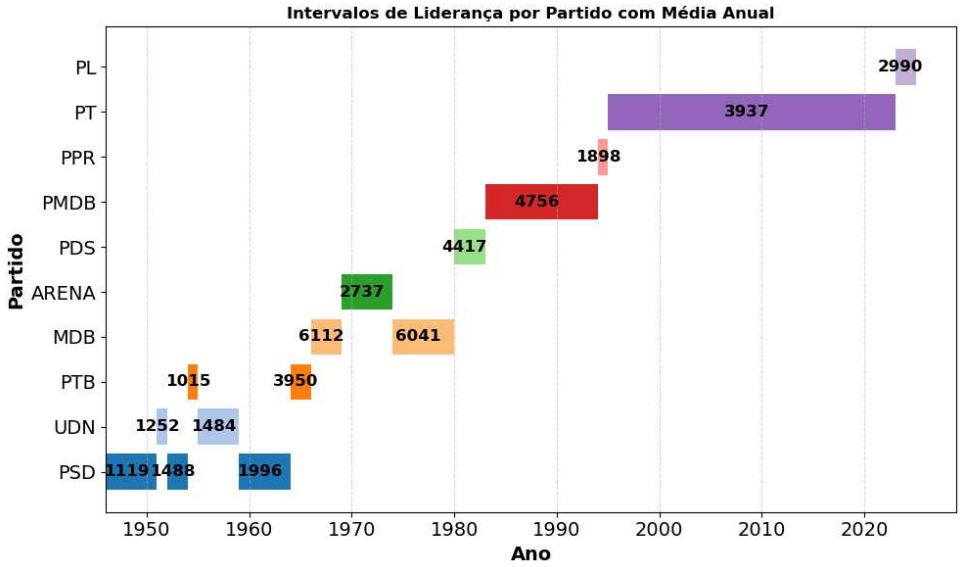
Outro destaque é o ano de 1988, quando ocorreu a promulgação da Constituição Federal, o que resultou em um expressivo aumento no número de discursos, conforme esperado. Já nos anos que antecederam, coincidiram com, e sucederam o impeachment da presidenta Dilma Rousseff, houve uma quantidade de discursos acima da média, evidenciando o impacto desse momento político nos debates parlamentares.

Também pode ser observado que, apesar de haver vários partidos com uma grande quantidade de deputados, a liderança em relação à quantidade de discursos flutuou entre poucos partidos, como pode ser verificado abaixo. Além disso, nota-se que os partidos que mais discursavam eram, em sua maioria, aqueles que eram ligados ao Presidente da República do respectivo período, refletindo a influência política e o alinhamento partidário durante os mandatos presidenciais.

Sabe-se que a quantidade de deputados de cada estado é proporcional à sua população. A partir de 1994, o número total de deputados federais foi fixado em 513. Antes dessa definição, não havia um número estabelecido, o que gerava flutuações consideráveis no total de deputados.

Devido à grande quantidade de nomes presentes no banco de dados, alguns erros

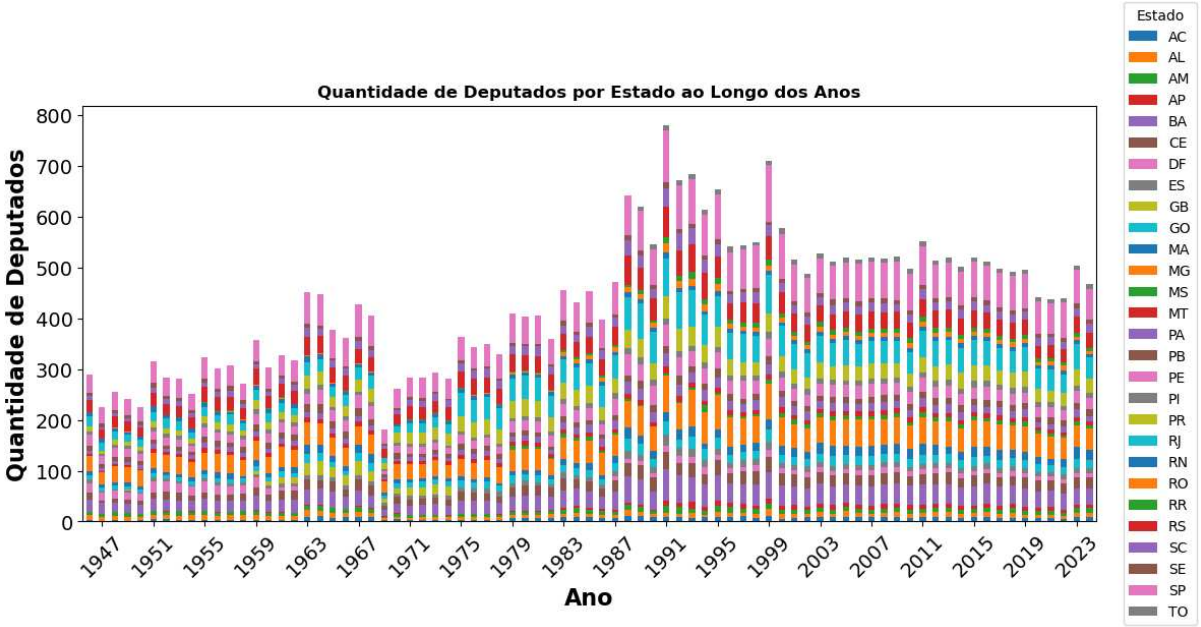
Figura 15 – Partidos com a maior quantidade de discursos por ano, onde cada cor representa um partido.



Fonte: Autor

de digitação podem ser observados, resultando, em alguns anos, em um número de deputados superior ao permitido. Esse problema é mais evidente na década de 1990, período em que os discursos ainda não eram publicados digitalmente, o que dificultava a padronização das informações.

Figura 16 – Número de deputados por estado ao longo dos anos, onde cada cor representa um estado brasileiro.



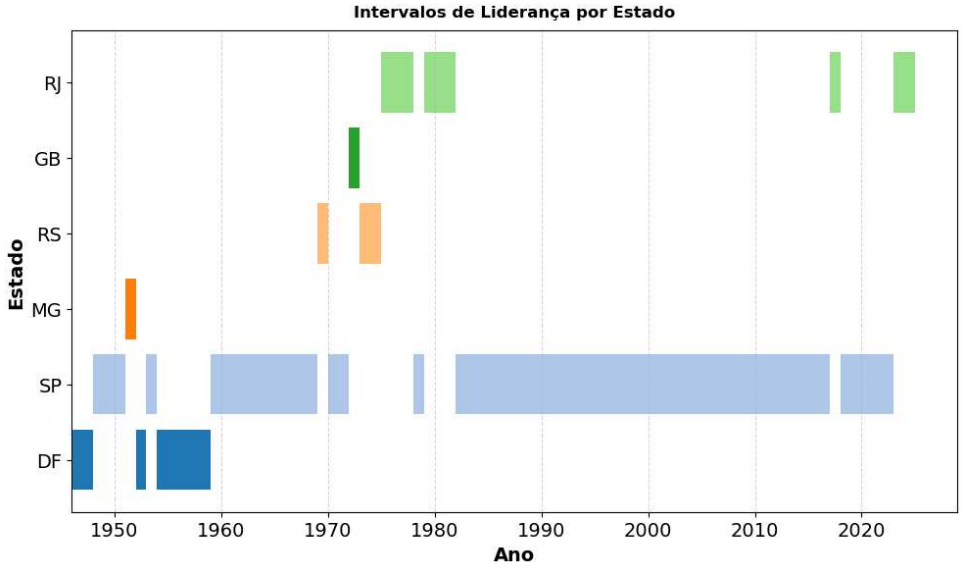
Fonte: Autor

Apesar disso, é possível perceber no gráfico 16 que, após os anos 2000, o limite de 513 deputados passou a ser observado com maior precisão, quando a publicação digital dos

discursos foi iniciada. No entanto, ainda existem flutuações em torno desse número, refletindo variações nos dados, que podem ser atribuídas a erros de digitação ou inconsistências no registro dos deputados ao longo do tempo.

Podemos observar a variação no número de deputados por estado no gráfico anterior, onde SP, a cor rosa, está predominando em relação aos números de deputados. Dada a grande diferença em relação aos demais, é esperado que os estados com maior número de representantes, como SP, liderem na quantidade de discursos ao longo dos anos. Esse padrão pode ser claramente visto no gráfico abaixo:

Figura 17 – Estado que liderou na quantidade de discursos ao longo dos anos



Fonte: Autor

4.2 LDA

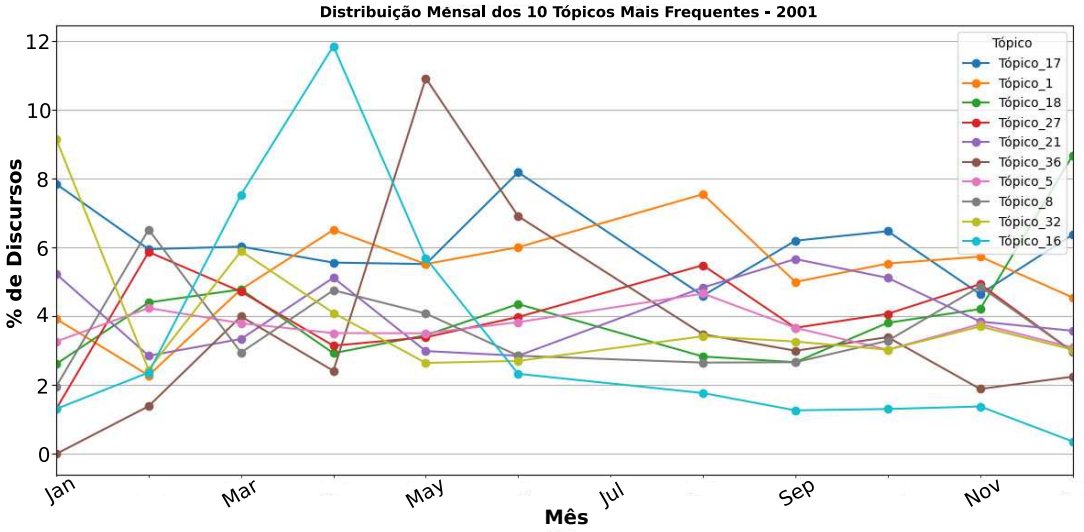
O LDA é um modelo utilizado para análise de propriedades latentes há 22 anos, sendo, portanto, amplamente consolidado na literatura. Um dos principais fatores que determinam a qualidade dos resultados obtidos pelo modelo é a combinação entre o pré-processamento realizado e a escolha dos hiperparâmetros.

No presente trabalho, optou-se por utilizar 40 tópicos, apresentar o ranking das 10 palavras mais frequentes de cada tópico e considerar apenas palavras que apareçam em no mínimo 150 discursos, mas não em mais de 98% dos discursos. Dessa forma, são filtradas tanto as palavras excessivamente comuns, que surgem em mais de 98% dos discursos, quanto as extremamente raras, que ocorrem em menos de 150 discursos. A escolha de 40 tópicos permite uma explicitação mais específica dos temas identificados. Reduzir o número de tópicos, os resultados tenderiam a agrupar os temas de forma mais generalizada. Os resultados dos discursos da câmara de 2001 se encontram na tabela 4.

O modelo foi, então, salvo e aplicado à lista de discursos para identificar o tópico predominante em cada um. Ao verificar a porcentagem de tópicos abordados mensalmente ao

longo de 2001, obteve-se a distribuição da figura 18.

Figura 18 – Distribuição mensal dos 10 tópicos mais populares em 2001



Fonte: Autor

É possível observar dois picos nos meses de abril e maio, nos tópicos 16 e 36, respectivamente. Isso indica um aumento na frequência de discursos que abordaram esses tópicos nesses períodos. De acordo com a Tabela 4, o tópico 36 está relacionado à temática de energia, enquanto o tópico 16 trata de corrupção. Em 2001, os níveis dos reservatórios de água estavam baixos devido a uma seca que estava ocorrendo, o que levou à necessidade de racionamento de energia e ocasionou apagões, conforme evidenciado na matéria do G1 [45]. Esses apagões ocorreram durante o mês de maio e simbolizaram uma crise elétrica no Brasil naquele período.

Já o Tópico 16, que apresenta um pico no mês de abril e trata sobre corrupção, reflete temas que marcaram o governo FHC durante seus mandatos. Houve, inclusive, uma tentativa de instaurar uma CPI para investigar esses casos, mas não foi obtido o número necessário de assinaturas, e a proposta foi arquivada em maio [46].

As assinaturas para a instauração dessa CPI começaram a ser recolhidas em abril de 2001, o que sugere que discussões nos plenários ocorreram abordando esse tema, explicando assim o pico observado no mês de abril, pois estavam faltando somente 9 assinaturas para a implementação [47].

Como foi dito anteriormente, o LDA realiza, inicialmente, uma distribuição aleatória de tópicos para as palavras dos documentos. Essa distribuição inicial pode ofuscar certos tópicos que não sejam tão predominantes. Entretanto, tópicos relevantes deverão ser detectados independentemente da *seed* utilizada. Dessa forma, diversos modelos foram gerados utilizando-se dos mesmos parâmetros mencionados anteriormente, mas com *seeds* diferentes, a fim de variar as distribuições iniciais e, assim, verificar os tópicos relevantes abordados em 2001.

Foram gerados cinco modelos baseados em cinco diferentes *seeds*, sendo 42 uma escolha comum em exemplos da literatura por motivos convencionais. Uma cópia do arquivo dos discursos foi usada para cada modelo, de forma a obter uma distribuição dos tópicos predo-

Tabela 4: Tópicos e palavras-chave dos discursos da câmara de 2001.

Tópico 0	**Tópico 1**	**Tópico 2**	**Tópico 3**	**Tópico 4**	**Tópico 5**	**Tópico 6**	**Tópico 7**
receber, orçamento, pensionista, reajuste, previdência, aposentado, real, em, mínimo, salário	se, rio, prefeito, população, obra, região, por, município, cidade, em	catarina, governo, produto, brasileiro, grosso, país, mato, brasil, sul, em	direito, tributário, renda, lei, trabalho, governo, imposto, trabalhador, projeto, em	esse, câmara, trabalho, informação, relatório, cpi, comunicação, por, comissão, em	paulo, por, crime, militar, violência, público, policial, polícia, segurança, em	2º, prisão, por, constituição, inciso, 1º, especial, lei, em, art	internacional, americano, estados, unidos, por, brasileiro, brasil, mundo, país, em
Tópico 8	**Tópico 9**	**Tópico 10**	**Tópico 11**	**Tópico 12**	**Tópico 13**	**Tópico 14**	**Tópico 15**
por, pronunciamento, aparte, agradeço, presidência, este, exo, em, mesa, exa	república, direito, justiça, público, nacional, tribunal, se, federal, por, em	recurso, desenvolvimento, por, nordestino, seca, governo, água, nordeste, região, em	por, pobreza, governo, família, educação, bolsa, criança, escola, em, programa	doença, brasileiro, esse, preço, população, medicamento, consumidor, por, saúde, em	parecer, lei, aprovar, recurso, por, projeto, em, orçamento, comissão, emenda	voto, por, dever, processo, imunidade, público, código, mandato, ética, em	dever, processo, público, sociedade, país, partido, se, por, político, em
Tópico 16	**Tópico 17**	**Tópico 18**	**Tópico 19**	**Tópico 20**	**Tópico 21**	**Tópico 22**	**Tópico 23**
público, querer, senador, por, país, denúncia, governo, cpi, corrupção, em	discussão, este, querer, líder, comissão, plenário, votar, votação, matéria, em	partido, matéria, por, urgência, bloco, votar, acordo, projeto, em, líder	painel, eletrônico, exa, requerimento, interno, ordem, em, mesa, sessão, regimento	estatuto, brasileiro, por, idoso, aprovar, este, lei, comissão, projeto, em	desenvolvimento, setor, mercado, economia, econômico, por, brasileiro, brasil, país, em	trabalho, receber, direito, por, pagar, fundo, empresa, governo, em, trabalhador	agrário, por, agricultor, governo, pequeno, terra, agricultura, produtor, rural, em
Tópico 24	**Tópico 25**	**Tópico 26**	**Tópico 27**	**Tópico 28**	**Tópico 29**	**Tópico 30**	**Tópico 31**
executivo, legislativo, matéria, votar, senado, câmara, este, nacional, congresso, em	se, congresso, esse, lei, militar, por, governo, em, provisório, medida	dívida, por, fernando, henrique, país, federal, servidor, em, público, governo	história, político, vida, país, brasil, brasileiro, este, homem, por, em	por, governador, pt, antonio, pmdb, carlos, senador, bahia, partido, em	comissão, neves, trabalho, querer, brasileiro, câmara, por, este, exa, em	pena, esse, direito, se, público, constituição, por, projeto, em, lei	financiamento, instituição, projeto, crédito, desenvolvimento, financeiro, recurso, em, amazônia, banco
Tópico 32	**Tópico 33**	**Tópico 34**	**Tópico 35**	**Tópico 36**	**Tópico 37**	**Tópico 38**	**Tópico 39**
vida, criança, humano, direito, país, brasil, por, social, mulher, em	projeto, vereador, público, serviço, municipal, prefeito, saneamento, em, município, água	ministério, por, governo, real, programa, social, milhão, recurso, saúde, em	esse, transporte, conta, ministério, justiça, público, tribunal, por, federal, em	país, energético, setor, por, crise, elétrico, empresa, governo, energia, em	governo, estudante, aluno, público, federal, ensino, professor, universidade, educação, em	jornal, por, orador, alagoa, anal, francisco, referir, janeiro, em, rio	prestar, josé, santo, se, médico, serviço, trabalho, profissional, por, em

minantes em cada discurso com base no respectivo modelo gerado.

Após isso, foi realizada a comparação entre os tópicos predominantes de cada conjunto de discursos. Por exemplo, suponha que o discurso 1, quando processado com o modelo baseado na *seed* 42 (denominado "modelo_42"), tenha o tópico 34 como predominante. Esse mesmo discurso, ao ser analisado com o modelo_1, pode apresentar o tópico 28 como predominante. Nesse caso, são comparadas as 10 palavras mais comuns de ambos os tópicos. Se houverem 7 ou mais palavras em comum, ou seja, 70% ou mais de similaridade, considera-se que esse tópico, agora nomeado, é um tópico predominante nos discursos analisados. A tabela 5 possui a comparação entre algumas das *seeds* utilizadas.

SEEDs	**Tópicos (SEED X)**	**Tópicos (SEED Y)**
SEED 1 vs SEED 2	70%: Rodovias federais, Legislação, Votação no Plenário, Politica Internacional 80%: Segurança pública, Petróleo, Acordo comercial, Agricultura Familiar 90%: Direitos do Trabalhador, Gestão Municipal 100%: Energia	70%: Acordo comercial, Legislação, Votação no Plenário, Rodovias federais 80%: Acordo comercial, Segurança pública, Agricultura Familiar, Agronegócio 90%: Gestão Municipal, Direitos do Trabalhador 100%: Energia
SEED 1 vs SEED 3	70%: Rodovias Federais, Segurança Pública, Energia, Violência contra mulher*, Questões sociais, Votação no Plenário, Sistema Financeiro 80%: Legislação, Corrupção, Política Institucional e Eleitoral, Salário servidores, Ensino superior 90%: Suporte financeiro familiar, Politica Internacional 100%: Direitos do Trabalhador, Gestão Municipal	70%: Direitos Humanos*, Votação Plenário, Rodovias Federais, Segurança Pública, Energia, Sistema Financeiro 80%: Ensino Superior, Corrupção, Política Institucional e Eleitoral, Salário servidores, Legislação 90%: Suporte financeiro familiar, Politica Internacional 100%: Gestão Municipal, Direitos do Trabalhador
SEED 42 vs SEED 1	70%: Politica Internacional, Saúde, Agronegócio, Corrupção, Suporte financeiro familiar, Rodovias Federais, Segurança pública, Legislação 80%: Energia, Educação superior 90%: Gestão Municipal, Salário servidores 100%: Direitos do trabalhador	70%: Legislação, Rodovias Federais, Segurança pública, Corrupção, Petróleo, Politica Internacional, Suporte Financeiro familiar, Saúde 80%: Educação superior, Energia 90%: Salário servidores, Gestão Municipal 100%: Direitos do Trabalhador
SEED 42 vs SEED 2	70%: Economia 80%: Seca Nordeste, Politica Internacional, Corrupção, Energia 90%: Agronegócio, Direitos do trabalhador, Gestão Municipal 100%: —	70%: Economia 80%: Acordo Comercial, Energia, Seca, Corrupção 90%: Gestão Municipal, Direitos do Trabalhador, Agronegócio 100%: —
SEED 42 vs SEED 3	70%: Saúde, Dívida Pública e Orçamento, Energia, Salário servidores, Economia 80%: Politica Internacional, Rodovias Federais, Segurança pública 90%: Educação Superior, Gestão Municipal 100%: Direitos do Trabalhador	70%: Orçamento, Saúde, Salário servidores, Energia, Economia 80%: Rodovias Federais, Politica Internacional, Segurança pública 90%: Gestão Municipal, Ensino Superior 100%: Direitos do Trabalhador
SEED 42 vs SEED 4	70%: Taxas para população, Dívida Pública e Orçamento, Agronegócio, Corrupção, Leis, Salário Servidores 80%: Politica Internacional, Energia, Suporte Financeiro familiar, Segurança Pública, Educação Superior, Seca 90%: Saúde, Direitos do Trabalhador 100%: —	70%: Salário Servidores, Orçamento, Corrupção, Taxas para população, Economia*, Estrutura Constitucional* 80%: Energia, Segurança Pública, Seca, Educação Base*, Politica Internacional, Educação Superior 90%: Saúde, Direitos do Trabalhador 100%: —

Tabela 5: Tópicos com mais de 70% de similaridade entre diferentes SEEDs

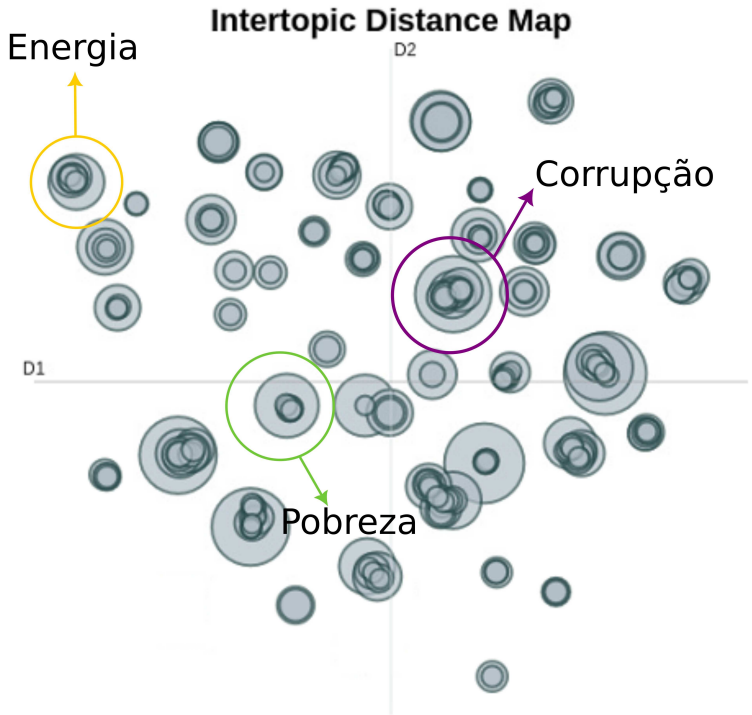
Pode-se notar na Tabela 5 que os tópicos "energia" e "corrupção" aparecem com

frequência, evidenciando, novamente, sua relevância em 2001. Observa-se ainda que o tópico "seca" também ocorre com frequência, o que está relacionado à crise energética enfrentada pelo Brasil em 2001, conforme comentado anteriormente.

4.3 BERTopic

Utilizando exatamente o mesmo pré-processamento, os discursos foram submetidos ao algoritmo BERTopic, mantendo-se os parâmetros padrão do modelo. Espera-se que os tópicos relevantes identificados anteriormente reapareçam, validando a consistência dos resultados. Uma das principais vantagens do BERTopic é que não é necessário definir previamente o número de tópicos, essa estimativa é realizada automaticamente pelo modelo. Como consequência, os tópicos identificados tendem a ser mais específicos e semanticamente coesos. A seguir, apresenta-se o gráfico da figura 20.

Figura 19 – Gráfico do modelo BERTopic. Existem 203 tópicos, e a ordem dos tópicos reflete sua relevância.

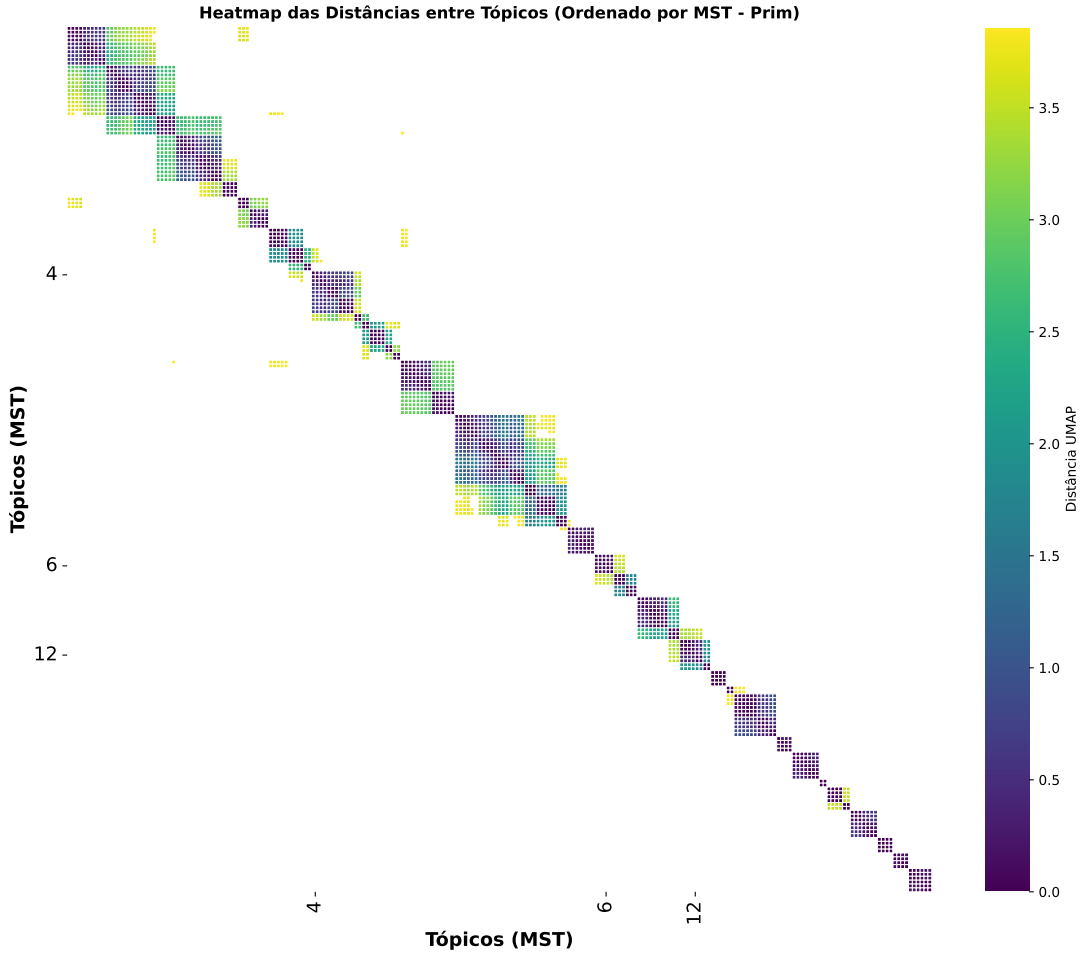


Fonte: Autor

O tópico 6 refere-se à temática da energia, refletindo a relevância dos apagões ocorridos em 2001 e o tópico 4 aborda corrupção. Já o tópico 12 aborda a seca enfrentada naquele ano. É importante destacar que a numeração dos tópicos indica sua relevância no conjunto de discursos analisados. Dessa forma, observa-se que os resultados obtidos com o BERTopic coincidem com aqueles encontrados por meio do LDA. No gráfico, os círculos representam os tópicos: quanto mais próximos entre si, mais relacionados semanticamente são, por outro lado,

quanto mais distantes, mais distintos são em relação ao conteúdo abordado. Entretanto existe uma grande quantidade de tópicos e uma grande quantidade de tópicos sobrepostos que seria possível agrupar para formar um grande tópico, com esse pensamento em mente foi calculada a *Minimum Spanning Tree* (MST) desses tópicos utilizando o algoritmo de Primm. Com a matriz MST montada, foi feito o plot do *heatmap* que é possível observar a seguir;

Figura 20 – Heatmap da matriz MST formada pelo gráfico do BERTopic



Fonte: Autor

Foi considerado um corte para manter somente as ligações mais fortes, com isso observamos que existem aglomerados de tópicos. Ao analisar o aglomerado do tópico de energia, nota-se que os tópicos que o compõe estão tratando da crise energética, distribuição de energia etc. Tópicos relacionados a energia. Para o tópico de corrupção nota-se que os tópicos tratados são discursos do Fernando Henrique, que estava ligado aos escândalos de corrupção. Por fim, o tópico que tratava sobre seca possui uma aproximação com tópicos ligados ao sertão e a pobreza.

5 CONCLUSÕES E PERSPECTIVAS

Por meio da aplicação dos algoritmos LDA e BERTopic, foi possível identificar e caracterizar os principais temas debatidos pelos parlamentares, bem como sua distribuição ao longo dos meses e a frequência de ocorrência desses tópicos no período analisado.

Os resultados obtidos com o LDA mostraram que temas como educação, saúde, economia, meio ambiente e segurança pública mantêm presença constante no repertório parlamentar, com variações de intensidade associadas a conjunturas específicas. Além disso, observou-se que crises políticas e eventos marcantes, como o racionamento de energia elétrica e os escândalos de corrupção, coincidem com picos temáticos e aumento no volume de discursos.

Já a aplicação do BERTopic permitiu uma segmentação temática mais refinada e sensível ao contexto, identificando tópicos emergentes e menos frequentes com maior precisão, como debates sobre seca, direitos sociais e temas regionais. A análise também evidenciou como determinados tópicos se agrupam semanticamente..

Em síntese, este estudo contribui para a compreensão da atividade discursiva na Câmara dos Deputados, demonstrando como métodos de Processamento de Linguagem Natural podem ser aplicados para mapear a dinâmica temática do parlamento brasileiro. As descobertas oferecem subsídios relevantes para estudos de ciência política, comunicação e comportamento legislativo. Como desdobramentos futuros, propõe-se aprofundar a análise da relação entre temas discursivos e posicionamentos ideológicos, bem como explorar a integração com redes de coautoria, votações e proposições legislativas.

Um dos resultados encontrados foi a existência de uma dinâmica entre os tópicos relevantes na sociedade e aqueles abordados pelos deputados, como observado nos casos do apagão, da corrupção e da seca. Isso indica que há uma interação social entre a sociedade e a Câmara dos Deputados. Lidar com dinâmicas sociais por meio de métodos da física já é uma abordagem adotada há alguns anos, como exemplificado no artigo *"Statistical Physics of Social Dynamics"* [48], que discute ferramentas para tratar diferentes tipos de dinâmicas sociais, como dinâmicas de opinião, culturais, hierárquicas, entre outras.

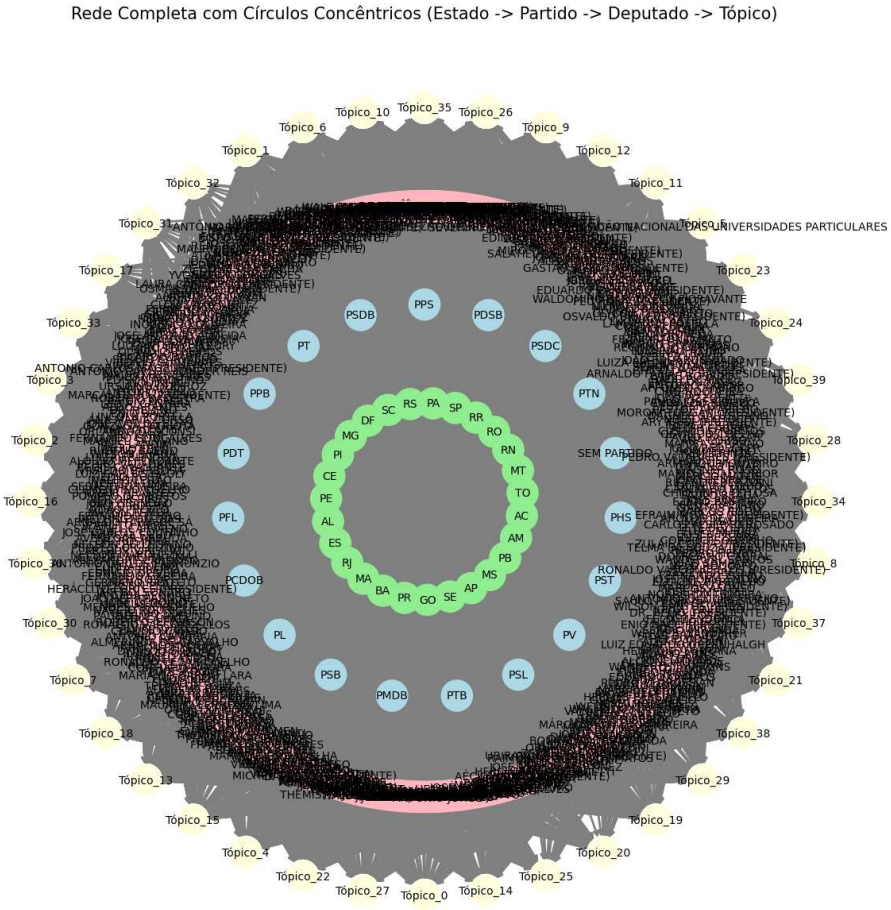
Dessa forma, buscar quantificar essas dinâmicas utilizando métodos físicos se mostra um caminho viável, ainda que de forma puramente qualitativa.

Uma outra maneira de relacionar os dados seria por meio de grafos de conhecimento, também conhecidos como *Knowledge Graphs*, que são estruturas de dados que representam informações através de entidades (como pessoas, lugares ou objetos) conectadas por relacionamentos significativos. Dessa forma, forma-se uma rede em que cada entidade corresponde a um nó, e as relações entre elas são representadas por arestas rotuladas. Essa abordagem facilita a interpretação dos dados, permite inferências e é amplamente utilizada em motores de busca, assistentes virtuais e sistemas de recomendação.

Aplicando esse método para relacionar estados, partidos, deputados e tópicos, é

possível construir uma rede, como pode ser vista na figura 21.

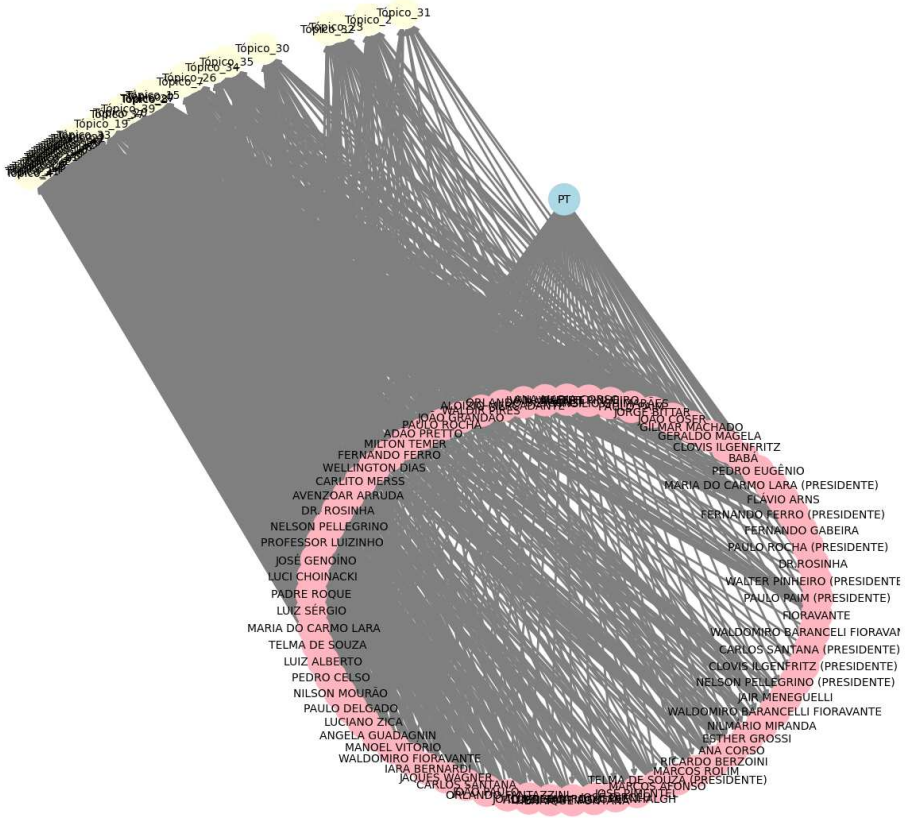
Figura 21 – Grafo de conhecimentos dos discursos da Câmara em 2001.



Fonte: Autor

Devido a grande quantidade de deputados e discursos nota-se que o grafo é extremamente denso e nenhum informação sobre sua estrutura é interpretável, mas é possível filtrar pelos nós, analisando somente o partido dos trabalhadores (PT) têm-se na figura 22.

Figura 22 – Grafo de conhecimentos dos discursos da Câmara em 2001.



Fonte: Autor

REFERÊNCIAS

[1] PONTUAL, H. D. *Uma breve história das Constituições do Brasil*. 2013. Dísponivel em: https://www.senado.gov.br/noticias/especiais/constituicao25anos/historia-das-constituicoes.htm. Acesso em: 15 dez. 2024.

[2] CASA CIVIL. *Emendas Constitucionais*. 2024. Dísponivel em: http://www.planalto.gov.br/ccivil_03/constituicao/Emendas/Emc/quadro_emc.htm. Acesso em: 15 dez. 2024.

[3] CASA CIVIL. *Constituição da República Federativa do Brasil de 1988*. 2024. Dísponivel em: http://www.planalto.gov.br/ccivil_03/constituicao/Emendas/Emc/quadro_emc.htm. Acesso em: 15 dez. 2024.

[4] DUARTE, G. D. *Conjutura atual em OSPB*. 10. ed. *[S.l.]*: Lê, 1992. 138–142 p.

[5] DUARTE, G. D. *Conjutura atual em OSPB*. 10. ed. *[S.l.]*: Lê, 1992. 161 p.

[6] CÂMARA DOS DEPUTADOS. *Deputados*. 2024. Dísponivel em: https://www2.camara.leg.br/transparencia/acesso-a-informacao/copy_of_perguntas-frequentes/deputados. Acesso em: 15 dez. 2024.

[7] SENADO FEDERAL. *Como funciona a eleição de deputados federais e estaduais*. 2018. Dísponivel em: https://www12.senado.leg.br/noticias/materias/2018/10/01/como-funciona-a-eleicao-de-deputados-federais-e-estaduais. Acesso em: 15 dez. 2024.

[8] BARBOSA, A. J. *O Poder Legislativo no Brasil*. 2016. Dísponivel em: https://www12.senado.leg.br/noticias/materias/2018/10/01/como-funciona-a-eleicao-de-deputados-federais-e-estaduais. Acesso em: 15 dez. 2024.

[9] SUPREMO TRIBUNAL DE JUSTIçA. *Tribunais superiores*. Dísponivel em: https://www12.senado.leg.br/noticias/materias/2018/10/01/como-funciona-a-eleicao-de-deputados-federais-e-estaduais. Acesso em: 15 dez. 2024.

[10] SUPREMO TRIBUNAL DE JUSTIçA. *Poder Judiciário Brasileiro*. Dísponivel em: https://international.stj.jus.br/pt/Poder-Judiciario-Brasileiro. Acesso em: 15 dez. 2024.

[11] CÂMARA DOS DEPUTADOS. *PL 3559/2012*. 2012. Dísponivel em: https://international.stj.jus.br/pt/Poder-Judiciario-Brasileiro. Acesso em: 15 dez. 2024.

[12] CÂMARA DOS DEPUTADOS. *Maio Amarelo: Lei Seca completa 10 anos; autor da norma avalia conquistas e desafios*. 2018. Dísponivel em: https://www.camara.leg.br/radio/programas/537030-maio-amarelo-lei-seca-completa-10-anos-autor-da-norma-avalia-conquistas-e-desafios/. Acesso em: 16 fev. 2025.

[13] CÂMARA DOS DEPUTADOS. *O papel das Comissões*. Dísponivel em: https://www2.camara.leg.br/atividade-legislativa/comissoes/o-papel-das-comissoes. Acesso em: 16 dez. 2024.

[14] CÂMARA DOS DEPUTADOS. *Conheça a tramitação de projetos de lei*. Dísponivel em: https://www.camara.leg.br/noticias/573454-SAIBA-MAIS-SOBRE-A-TRAMITACAO-DE-PROJETOS-DE-LEI. Acesso em: 16 dez. 2024.

[15] CÂMARA DOS DEPUTADOS. *Propostas legislativas*. Dísponivel em: https://www.camara.leg.br/buscaportalcontextoBusca=BuscaProposicoespagina=1. Acesso em: 16 dez. 2024.

[16] CÂMARA DOS DEPUTADOS. *Lei de Acesso à Informação (2011)*. 2011. Dísponivel em: https://www2.camara.leg.br/legin/fed/lei/2011/lei-12527-18-novembro-2011-611802-norma-pl.html. Acesso em: 16 dez. 2024.

[17] BRITO, A. C. M. *Using complex networks to analyze the Brazilian Chamber of Deputies*. Dissertação (Dissertação de Mestrado) — Universidade de São Paulo, 2020.

[18] AMANCIO, T. T. J. D. R. Brazilian political study: with topics analysis and complex networks. In: SPRINGER NATURE. New Trends in Database and Information Systems: ADBIS 2024 Short Papers, Workshops, Doctoral Consortium and Tutorials, Bayonne, France, August 28–31, 2024, *Proceedings*. *[S.l.]*, 2024. v. 2186, p. 130.

[19] MOREIRA, D. Com a palavra os nobres deputados: ênfase temática dos discursos dos parlamentares brasileiros. *Dados*, v. 63, n. 1, p. 1–37, 2020.

[20] GRAVES, A. *et al*. A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 31, n. 5, p. 855–868, 2009.

[21] LI, X.; WU, X. *Constructing Long Short-Term Memory based Deep Recurrent Neural Networks for Large Vocabulary Speech Recognition*. 2015.

[22] YAO, K. *et al*. Recurrent neural networks for language understanding. In: *Interspeech*. *[S.l.*: *s.n.]*, 2013. p. 2524–2528.

[23] KARPATHY, A. *The Unreasonable Effectiveness of Recurrent Neural Networks*. 2015. Dísponivel em: http://karpathy.github.io/2015/05/21/rnn-effectiveness/. Acesso em: 5 fev. 2024.

[24] BASODI, S. *et al*. Gradient amplification: An efficient way to train deep neural networks. *Big Data Mining and Analytics*, TUP, v. 3, n. 3, p. 196–207, 2020.

[25] BOULILA, W. *et al*. *Weight Initialization Techniques for Deep Learning Algorithms in Remote Sensing: Recent Trends and Future Perspectives*. 2021.

[26] HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural Computation*, v. 9, n. 8, p. 1735–1780, 11 1997. ISSN 0899-7667.

[27] HOCHREITER, S.; SCHMIDHUBER, J. Lstm can solve hard long time lag problems. In: *Proceedings of the 10th International Conference on Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 1996. (NIPS'96), p. 473–479.

[28] GERS, F. A.; SCHMIDHUBER, J.; CUMMINS, F. Learning to forget: Continual prediction with lstm. *Neural Computation*, v. 12, n. 10, p. 2451–2471, 10 2000. ISSN 0899-7667.

[29] KARIM, F. *et al*. Lstm fully convolutional networks for time series classification. *IEEE Access*, Institute of Electrical and Electronics Engineers (IEEE), v. 6, p. 1662–1669, 2018. ISSN 2169-3536.

[30] WU, Y. *et al*. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[31] CHO, K. *et al*. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[32] SU, Y.; KUO, C.-C. J. On extended long short-term memory and dependent bidirectional recurrent neural network. *Neurocomputing*, Elsevier BV, v. 356, p. 151–161, set. 2019. ISSN 0925-2312.

[33] MIKOLOV, T. *et al*. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[34] PENNINGTON, J.; SOCHER, R.; MANNING, C. GloVe: Global vectors for word representation. In: MOSCHITTI, A.; PANG, B.; DAELEMANS, W. (Ed.). *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, 2014. p. 1532–1543.

[35] BAHDANAU, D.; CHO, K.; BENGIO, Y. *Neural Machine Translation by Jointly Learning to Align and Translate*. 2016.

[36] RATHI, R. N.; MUSTAFI, A. The importance of term weighting in semantic understanding of text: A review of techniques. *Multimedia Tools and Applications*, v. 82, n. 7, p. 9761–9783, 2023.

[37] VASWANI, A. *et al*. Attention is all you need. *Advances in neural information processing systems*, v. 30, 2017.

[38] JONES, K. S. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, Emerald Group Publishing Limited, v. 28, n. 1, p. 11–21, 1972.

[39] DEERWESTER, S. *et al*. Indexing by latent semantic analysis. *Journal of the American society for information science*, Wiley Online Library, v. 41, n. 6, p. 391–407, 1990.

[40] BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. *Journal of machine Learning research*, v. 3, n. Jan, p. 993–1022, 2003.

[41] GROOTENDORST, M. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*, 2022.

[42] CÂMARA DOS DEPUTADOS. *Sessões e Reuniões*. 2024. Dísponivel em: https://www2.camara.leg.br/atividade-legislativa/discursos-e-notas-taquigraficas. Acesso em: 6 dez. 2024.

[43] NASCIMENTO, E. N. do *et al*. Automatização da coleta de discursos políticos de parlamentares brasileiros. *Universidade de Fortaleza – Programa de Iniciação Científica*, 2022.

[44] COSTA, W. A. J.; OLIVEIRA, E. A. de. Caracterização de discursos políticos através de aprendizado de máquina. *Universidade de Fortaleza – Programa de Iniciação Científica PIBIC*, 2021.

[45] G1. *Brasil sofreu 'apagão' em 2001 por seca nos reservatórios; relembre*. Dísponivel em: https://g1.globo.com/economia/noticia/2023/08/15/brasil-sofreu-apagao-em-2001-por-seca-nos-reservatorios-relembre.ghtml. Acesso em: 18 jul. 2025.

[46] JORNAL DO SENADO. *Sem as assinaturas necessárias, CPI da Corrupção é arquivada*. Dísponivel em: https://www2.senado.leg.br/bdsf/bitstream/handle/id/498279/2001-05-11.pdf?sequence=1. Acesso em: 18 jul. 2025.

[47] MADUENO, D. *CPI da corrupção depende de 9 deputados*. Dísponivel em: https://www1.folha.uol.com.br/fsp/brasil/fc2504200112.htm. Acesso em: 18 jul. 2025.

[48] CASTELLANO, C.; FORTUNATO, S.; LORETO, V. Statistical physics of social dynamics. *Reviews of Modern Physics*, American Physical Society (APS), v. 81, n. 2, p. 591–646, maio 2009. ISSN 1539-0756.