Universidade de Brasília - UnB

Faculdade de Ciências e Tecnologias em Engenharia - FCTE

Engenharia de Software

Sumarização Extrativa de Comentários no Contexto de Big Data: Um Estudo de Caso sobre o App Gov.Br

Autor: Henrique Amorim Costa Melo

Orientador: Prof. Dr. Glauco Vitor Pedrosa

Brasília, DF

2025

Henrique Amorim Costa Melo

Sumarização Extrativa de Comentários no Contexto de Big Data: Um Estudo de Caso sobre o App Gov.Br

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

Universidade de Brasília - UnB

Faculdade de Ciências e Tecnologias em Engenharia - FCTE

Orientador: Prof. Dr. Glauco Vitor Pedrosa

Brasília, DF

2025

Henrique Amorim Costa Melo
Sumarização Extrativa de Comentários no Contexto de Big Data: Um Estudo de Caso sobre o App Gov.Br/ Henrique Amorim Costa Melo. – Brasília, DF, 2025-
64 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Glauco Vitor Pedrosa

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade de Ciências e Tecnologias em Engenharia - FCTE , 2025.

1. Processamento de Linguagem Natural. 2. Sumarização de Conteúdo. I. Prof. Dr. Glauco Vitor Pedrosa. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Sumarização Extrativa de Comentários no Contexto de Big Data: Um Estudo de Caso sobre o App Gov.Br

CDU 02:141:005.6

Henrique Amorim Costa Melo

Sumarização Extrativa de Comentários no Contexto de Big Data: Um Estudo de Caso sobre o App Gov.Br

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

Trabalho aprovado. Brasília, DF, 19 de fevereiro de 2025:

Prof. Dr. Glauco Vitor Pedrosa
Orientador

Prof. Dr. Bruno César Ribas
Convidado 1

Prof. Dr. John Lenon Cardoso Gardenghi
Convidado 2

Brasília, DF
2025

Agradecimentos

Gostaria de expressar minha profunda gratidão a todos que contribuíram para a realização deste trabalho. Agradeço imensamente à minha família e amigos pelo apoio incondicional e pela paciência durante todo o processo. Meu agradecimento especial aos meus professores, cujas orientações e ensinamentos foram fundamentais para o desenvolvimento deste projeto. Por fim, um agradecimento especial ao meu orientador, Glauco Vitor Pedrosa, por sua orientação valiosa, suporte contínuo e por compartilhar seu conhecimento e experiência ao longo desta jornada.

Resumo

Este trabalho apresenta um experimento focado na sumarização de um grande volume de textos curtos, especificamente comentários de usuários do app Gov.br. O objetivo principal é reduzir o conjunto massivo de comentários a um conjunto menor de textos representativos, facilitando a análise e a tomada de decisões. Para isso, foram aplicadas técnicas de Processamento de Linguagem Natural (PLN) e sumarização extrativa baseada em grafos, utilizando o algoritmo LexRank, precedidas por uma etapa de filtragem de comentários baseada em características linguísticas para remover textos irrelevantes e redundantes, garantindo resumos concisos e representativos dos comentários dos usuários do aplicativo Gov.br. Os resultados obtidos demonstram uma abordagem prática para a análise de grandes volumes de textos curtos, possibilitando que os desenvolvedores priorizem as demandas mais relevantes e implementem melhorias de forma mais eficiente.

Palavras-chaves: Processamento de Linguagem Natural. Sumarização Extrativa de Textos. Big Data. Comentários de Usuários. Aplicativo Móvel

Abstract

This work presents an experiment focused on summarizing a large volume of short texts, specifically user comments from the Gov.br app. The main objective is to reduce the massive set of comments to a smaller set of representative texts, facilitating analysis and decision-making. To achieve this, Natural Language Processing (NLP) techniques and graph-based extractive summarization were applied, using the LexRank algorithm, preceded by a comment filtering stage based on linguistic characteristics to remove irrelevant and redundant texts, ensuring concise and representative summaries of user comments from the Gov.br application. The results obtained demonstrate a practical approach to analyzing large volumes of short texts, enabling developers to prioritize the most relevant demands and implement improvements more efficiently.

Key-words: Natural Language Processing. Extractive Text Summarization. Big Data. User Comments. Mobile Application.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

APP	Aplicativo
API	*Application Programming Interface*
SEO	*Search Engine Optimization*
PLN	Processamento de Linguagem Natural
TF	*Term Frequency*
IDF	*Inverse Document Frequency*
DUC	*Document Understanding Conference*
2FA	*Two Factor Authentication*
SVM	*Support Vector Machine*

Sumário

1 Introdução

1.1 Contextualização

A área de análise de dados massivos, popularmente conhecida como *Big Data*, surgiu em resposta ao rápido crescimento na quantidade de informações geradas em diversas fontes, como redes sociais, dispositivos móveis, sensores e transações digitais (BRYANT et al., 2016). Esse fenômeno trouxe à tona a necessidade de novas abordagens e tecnologias para coletar, armazenar e processar dados em larga escala (CHEN; MAO; LIU, 2014).

Diferentemente das abordagens tradicionais, que lidam com conjuntos de dados menores e estruturados, o *Big Data* lida com dados que variam em volume, velocidade e variedade, o que demanda soluções inovadoras, como o uso de computação em nuvem, aprendizado de máquina e algoritmos de mineração de dados. Essas tecnologias permitem que empresas e organizações extraiam *insights* valiosos que podem impulsionar a tomada de decisões estratégicas, otimizar processos e identificar tendências em tempo real. O impacto do *Big Data* é evidente em diversas áreas, incluindo a saúde, os negócios e o setor público, oferecendo novas possibilidades para a inovação e a eficiência operacional (MAYER-SCHöNBERGER; CUKIER, 2013).

1.2 Motivação e Desafios

Dada a grande quantidade de dados gerados, surge um desafio significativo: como analisar e interpretar eficientemente esse volume massivo de informações? A análise manual é impraticável devido ao tempo e aos recursos necessários. Portanto, a criação de ferramentas automatizadas para processar e resumir esses dados é essencial para uma análise eficaz e eficiente (MANYIKA et al., 2011).

A sumarização automática, por exemplo, oferece uma solução eficiente, condensando os dados e destacando os pontos mais relevantes, o que facilita a interpretação rápida e precisa das opiniões dos usuários (SOUZA; OLIVEIRA; PEREIRA, 2020). Dessa forma, as equipes de desenvolvimento e suporte podem focar sua atenção nos problemas mais críticos e nas sugestões mais frequentes, sem perder tempo em *feedbacks* irrelevantes ou repetitivos.

Além de economizar tempo e recursos, a sumarização de comentários permite uma análise mais objetiva e estruturada das informações. Ela ajuda a identificar padrões e tendências que poderiam passar despercebidos em uma leitura manual, fornecendo *insights* valiosos sobre a experiência dos usuários. Isso se torna ainda mais crucial em ambientes

onde a interação com o público é intensa e contínua, como em serviços públicos digitais. Ao priorizar as demandas mais recorrentes e os problemas mais urgentes, a sumarização contribui para a melhoria contínua do serviço oferecido, garantindo que as necessidades dos usuários sejam atendidas de forma ágil e eficiente.

Entretanto, a sumarização de textos, especialmente em contextos como o de comentários de usuários, enfrenta diversos desafios (NENKOVA; MCKEOWN, 2011; RADEV et al., 2004), tais como:

- Heterogeneidade e Variedade de Textos: Os comentários dos usuários podem variar muito em termos de estilo, tamanho e linguagem. Alguns podem ser curtos e objetivos, enquanto outros são longos e detalhados. Além disso, o vocabulário e a gramática podem ser informais ou mal estruturados, o que dificulta a criação de resumos precisos e consistentes.

- Identificação de Informações Relevantes: Nem todos os elementos de um comentário são igualmente importantes. O desafio está em determinar automaticamente quais partes são essenciais e quais são irrelevantes ou redundantes, especialmente quando há opiniões subjetivas, críticas vagas ou elogios genéricos.

- Qualidade da Sumarização: Um grande desafio é garantir que os resumos automáticos mantenham coerência e fluidez, especialmente quando gerados a partir de várias fontes ou comentários fragmentados. Muitas vezes, os resumos podem ser desorganizados ou omitir informações importantes devido à dificuldade de interpretar corretamente o texto original.

1.3 Estudo de Caso

Neste trabalho, o foco está na aplicação de técnicas de sumarização automática para analisar os comentários dos usuários do aplicativo **Gov.Br** (GOV..., 2024), amplamente utilizado pela população brasileira para acesso a serviços governamentais. Diariamente, a plataforma recebe um volume expressivo de comentários, abrangendo desde elogios e sugestões até críticas e reclamações. Segundo (PEDROSA et al., 2023), o principal desafio enfrentado pelas equipes responsáveis por esses dados é processar, analisar e extrair informações relevantes de maneira eficiente. Com o crescimento contínuo do número de usuários e interações, a análise individual de cada comentário torna-se inviável, o que pode retardar a identificação de problemas críticos e a implementação de melhorias essenciais.

Além da quantidade massiva de comentários, a variedade e a relevância das informações são fatores que adicionam complexidade ao problema. Muitos usuários fazem

publicações não relacionadas ao funcionamento do aplicativo ou comentários fora de contexto, tornando a análise mais desafiadora. Dessa forma, torna-se fundamental um processo de filtragem eficiente, capaz de identificar rapidamente os comentários mais pertinentes para os desenvolvedores. Sem essa filtragem, a presença excessiva de feedbacks irrelevantes pode mascarar questões críticas que demandam solução imediata.

Outro aspecto essencial a ser considerado é a necessidade de resposta ágil aos comentários mais relevantes. O governo e suas instituições devem manter-se alinhados às expectativas e necessidades dos cidadãos, garantindo que problemas identificados sejam tratados de forma eficaz. A demora na análise desses dados pode resultar na perda de oportunidades de aprimoramento ou no agravamento de falhas que poderiam ser corrigidas precocemente. Assim, o desenvolvimento de uma ferramenta automatizada de sumarização é essencial para destacar as informações mais relevantes, assegurando que ações corretivas e melhorias sejam implementadas de maneira oportuna e eficiente.

1.4 Objetivos

1.4.1 Objetivo Geral

O objetivo geral deste trabalho foi aplicar uma metodologia automatizada para a sumarização de comentários de usuários do aplicativo Gov.Br, gerando um conjunto mais conciso e representativo a fim de facilitar a análise das informações pelos desenvolvedores. O universo dos comentários a serem considerados foram aqueles com notas baixas (de 1 a 3) atributidas pelos usuários na Google Play Store.

1.4.2 Objetivos Específicos

Para atingir o objetivo geral do trabalho, foram definidos os seguintes objetivos específicos:

- Implementar um processo de filtragem de comentários baseado em características linguísticas para remover textos irrelevantes, redundantes ou de baixa qualidade.
- Aplicar a técnica de sumarização extrativa utilizando o algoritmo LexRank para gerar resumos a partir dos comentários filtrados.
- Criar um pipeline de processamento automatizado, integrando as etapas de coleta de dados, filtragem e sumarização em um fluxo de trabalho eficiente, capaz de processar grandes volumes de comentários em tempo hábil.

- Propor melhorias baseadas nos resultados obtidos, identificando possíveis ajustes ou aprimoramentos na ferramenta para otimizar a precisão e a relevância dos comentários sumarizados.

1.4.3 Estrutura da Monografia

Esta monografia está organizada em capítulos, conforme descrito a seguir:

- **Capítulo 2:** Aborda os principais conceitos relacionados à sumarização automática de textos, incluindo as técnicas de ranqueamento baseadas em grafos, como o LexRank, e a importância da filtragem de comentários antes da sumarização.
- **Capítulo 3:** Descreve as ferramentas e metodologias utilizadas no desenvolvimento da pesquisa, incluindo o processo de coleta de dados, a etapa de filtragem de comentários e a aplicação do LexRank para a geração dos sumários.
- **Capítulo 4:** Apresenta os resultados obtidos a partir da aplicação das técnicas propostas, analisando a eficácia da filtragem de comentários, a qualidade dos resumos gerados pelo LexRank e discute as limitações do método.
- **Capítulo 5:** Resume as principais contribuições do trabalho, destacando os avanços proporcionados pela implementação da filtragem e do LexRank na geração de sumários automáticos. Também sugere direções para trabalhos futuros, explorando melhorias no processo de filtragem e avaliação dos resumos.

2 Referencial Teórico

Este capítulo tem como objetivo apresentar os fundamentos teóricos relacionados às técnicas empregadas no desenvolvimento deste trabalho. Serão abordados os princípios da sumarização de textos, bem como uma explicação detalhada sobre os algoritmos utilizados.

2.1 Sumarização de Textos

A quantidade de informações e documentos disponíveis online aumentou significativamente, demandando avanços na área de sumarização de textos (ALLAHYARI et al., 2017). A sumarização é definido como um texto que, a partir de um ou mais documentos originais, transmite as informações mais relevantes, com uma extensão significativamente menor que o texto original. O processo de sumarização de textos visa produzir um resumo conciso e coerente, preservando o conteúdo e o significado essencial da fonte original. Nos últimos anos, diversas abordagens foram desenvolvidas e amplamente aplicadas em diferentes domínios, incluindo motores de busca e sites de notícias, que utilizam resumos para otimizar a navegação e a extração de conhecimento (ALLAHYARI et al., 2017).

A sumarização de textos apresenta desafios significativos, pois, ao contrário dos humanos, que leem um texto integralmente para compreendê-lo antes de elaborar um resumo destacando os pontos principais, os computadores não possuem o mesmo nível de compreensão linguística e contextual. Esse fator torna a tarefa de sumarização automática complexa e não trivial. Desde a década de 1950, pesquisas fundamentais foram realizadas, como o trabalho pioneiro de Luhn (THE. . . , 1958), que propôs um método para extrair trechos com base na frequência de palavras e frases, desconsiderando termos comuns de alta ocorrência. Esse método deu início a uma linha de pesquisa dedicada ao aprimoramento das técnicas de sumarização.

Em razão das limitações das técnicas abstrativas, que ainda enfrentam desafios como a correta modelagem semântica e a fluidez textual, a maior parte dos sistemas atuais de sumarização adota métodos extrativos. Assim, este trabalho concentra-se na sumarização extrativa, oferecendo uma visão geral das abordagens mais amplamente utilizadas nessa categoria. Ao longo do texto, serão exploradas técnicas de representação de tópicos, bases de conhecimento, impacto do contexto na sumarização e métodos de avaliação, proporcionando uma compreensão detalhada sobre o estado da arte na área de sumarização de textos.

2.1.1 Sumarização Extrativa e Abstrativa

A sumarização automática pode ser classificada em duas abordagens principais: *extrativa* e *abstrativa* (CAJUEIRO et al., 2023).

A **sumarização extrativa** consiste na seleção direta de sentenças ou trechos significativos do texto original para compor o resumo. Nesse método, as partes mais importantes são identificadas e extraídas sem modificações, resultando em um resumo que é essencialmente uma coleção de segmentos do próprio documento fonte. Técnicas comuns para determinar a relevância das sentenças incluem a análise da frequência de termos, métricas de similaridade e algoritmos baseados em grafos, como o TextRank (ALLAHYARI et al., 2017). Embora a sumarização extrativa seja mais simples de implementar e possa produzir resumos coerentes, ela está limitada ao conteúdo existente no texto original e pode não capturar a essência de forma tão eficaz quanto a abordagem abstrativa (NASCIMENTO, 2023).

Por outro lado, a **sumarização abstrativa** busca gerar novos textos que transmitam as informações essenciais do documento original, possivelmente utilizando uma linguagem diferente. Essa abordagem requer uma compreensão profunda do conteúdo, permitindo a reescrita das ideias principais em frases novas e concisas. Modelos de aprendizado profundo, especialmente aqueles baseados em redes neurais e transformadores, como o BART e o T5, têm sido empregados para essa tarefa, demonstrando avanços significativos na geração de resumos mais naturais e informativos (CAO, 2022). No entanto, a sumarização abstrativa enfrenta desafios como a preservação da precisão factual e a manutenção da coerência textual (NASCIMENTO, 2023).

2.1.2 Comparação entre as Abordagens

Enquanto a sumarização extrativa tende a ser mais direta e menos propensa a erros de geração, ela pode resultar em resumos que carecem de fluidez e coesão, uma vez que apenas concatena partes do texto original. A sumarização abstrativa, embora mais complexa e computacionalmente intensiva, tem o potencial de produzir resumos mais naturais e compreensíveis, capturando a essência do conteúdo de maneira mais eficaz (NASCIMENTO, 2023). Contudo, devido à sua complexidade, modelos abstrativos podem introduzir informações não presentes no texto original ou distorcer fatos, exigindo mecanismos robustos de controle de qualidade (OLIVEIRA; COSTA, 2021).

2.2 Características Linguísticas para Filtragem de Comentários

A análise de comentários de usuários enfrenta o desafio de identificar quais contribuições são realmente relevantes para a avaliação e aprimoramento de serviços ou pro-

dutos. Para abordar essa questão, estudos como o de (AUTOR(ES), 2022) propõem a extração de características linguísticas que auxiliam na distinção entre comentários úteis e não úteis. Essas características são organizadas em cinco dimensões principais: estilística, legibilidade, léxico, sentimento e conteúdo.

- **Dimensão Estilística:** Captura aspectos relacionados ao estilo dos comentários, como o número de palavras e sentenças. Por exemplo, a métrica de comprimento do comentário (*review-length*) avalia o número de palavras, enquanto o comprimento da sentença (*sentence-length*) considera o número de sentenças. Comentários mais longos podem fornecer informações detalhadas, mas também podem conter redundâncias. No contexto da sumarização, entender essas características é crucial para identificar trechos que agregam valor ao resumo final.

- **Dimensão de Legibilidade:** Avalia a facilidade de leitura dos comentários, considerando a presença de palavras complexas e aplicando métricas de legibilidade, como os índices de *Flesch* e *Dale-Chall*. A legibilidade indica o nível de compreensão necessário para interpretar o texto. Comentários com alta legibilidade são mais propensos a serem compreendidos corretamente pelos modelos de sumarização, garantindo que as informações essenciais sejam capturadas de forma eficaz.

- **Dimensão Léxica:** Refere-se às características do vocabulário utilizado nos comentários, incluindo a contagem de substantivos, verbos e adjetivos, além da diversidade léxica, que é a proporção de palavras únicas em relação ao total de palavras do comentário. Uma alta diversidade léxica pode indicar um comentário mais informativo, o que é benéfico para a geração de resumos que abrangem diferentes aspectos do conteúdo original (CAJUEIRO et al., 2023).

- **Dimensão de Sentimento:** Reflete as opiniões dos usuários, analisando a polaridade do comentário (positiva, negativa ou neutra) e a presença de palavras que expressam sentimentos. A análise de sentimentos é essencial para a sumarização, pois permite que o resumo mantenha o tom geral das opiniões dos usuários, garantindo que perspectivas positivas e negativas sejam representadas adequadamente (CAO, 2022).

- **Dimensão de Conteúdo:** Captura características textuais com base em técnicas de mineração de texto, como a frequência de termos (*tf-idf*) e a presença de palavras relacionadas à qualidade. Essas características ajudam a distinguir comentários informativos daqueles menos úteis, permitindo que o processo de sumarização selecione as informações mais relevantes para compor um resumo coeso e informativo (NASCIMENTO, 2023).

A incorporação dessas características linguísticas no pré-processamento dos comentários antes da sumarização pode melhorar significativamente a qualidade dos resumos gerados. Ao filtrar e classificar os comentários com base nessas dimensões, é possível focar nos conteúdos mais relevantes e informativos, otimizando o processo de sumarização. Modelos de classificação, como a Máquina de Vetores de Suporte (*Support Vector Machine* - SVM), têm sido utilizados para essa finalidade, demonstrando eficácia na categorização de textos com base em características linguísticas (JOACHIMS, 1998a).

Ao aplicar essas técnicas de filtragem e classificação, o processo de sumarização se torna mais eficiente, garantindo que os resumos reflitam com precisão as informações mais relevantes presentes nos comentários dos usuários.

2.3 Relação entre Ranqueamento, Sumarização e Filtragem

A sumarização automática de textos é uma tarefa essencial no processamento de linguagem natural, visando condensar informações mantendo a essência do conteúdo original. Nesse contexto, a filtragem e o ranqueamento desempenham papéis cruciais para aprimorar a qualidade dos resumos gerados.

2.3.1 Filtragem

A filtragem consiste na remoção de elementos irrelevantes ou redundantes do texto, como ruídos, informações supérfluas ou dados que não contribuem significativamente para o entendimento do conteúdo. Ao aplicar a filtragem antes da sumarização, garante-se que apenas as partes mais relevantes do texto sejam consideradas, o que melhora a coerência e a coesão do resumo final.

2.3.2 Ranqueamento

Após a etapa de filtragem, é necessário determinar quais sentenças ou trechos do texto são mais importantes para compor o resumo. Para isso, utilizam-se algoritmos de ranqueamento baseados em grafos, como o PageRank, o TextRank e o LexRank. Esses algoritmos avaliam a relevância das sentenças com base em sua centralidade e nas conexões com outras sentenças no grafo, permitindo a seleção das mais representativas para o resumo (MIHALCEA; TARAU, 2004b; ERKAN; RADEV, 2004a).

2.3.3 Integração entre Filtragem, Ranqueamento e Sumarização

A integração das etapas de filtragem e ranqueamento no processo de sumarização é fundamental para a obtenção de resumos de alta qualidade. Inicialmente, a filtragem remove informações irrelevantes, reduzindo o ruído no texto. Em seguida, o ranqueamento

identifica as sentenças mais centrais e relevantes, que são então selecionadas para compor o resumo final. Essa abordagem combinada assegura que o resumo seja conciso, coerente e contenha as informações mais importantes do texto original.

2.3.4 Benefícios da Filtragem e do Ranqueamento na Sumarização

A aplicação combinada de filtragem e ranqueamento no processo de sumarização oferece diversos benefícios:

- **Redução de Ruído:** A filtragem elimina informações irrelevantes, aumentando a clareza do resumo.
- **Seleção de Informações Relevantes:** O ranqueamento identifica as sentenças mais importantes, assegurando que o resumo contenha as informações essenciais.
- **Coerência e Coesão:** A integração dessas etapas contribui para a produção de resumos mais coerentes e coesos, facilitando a compreensão do conteúdo pelo leitor.

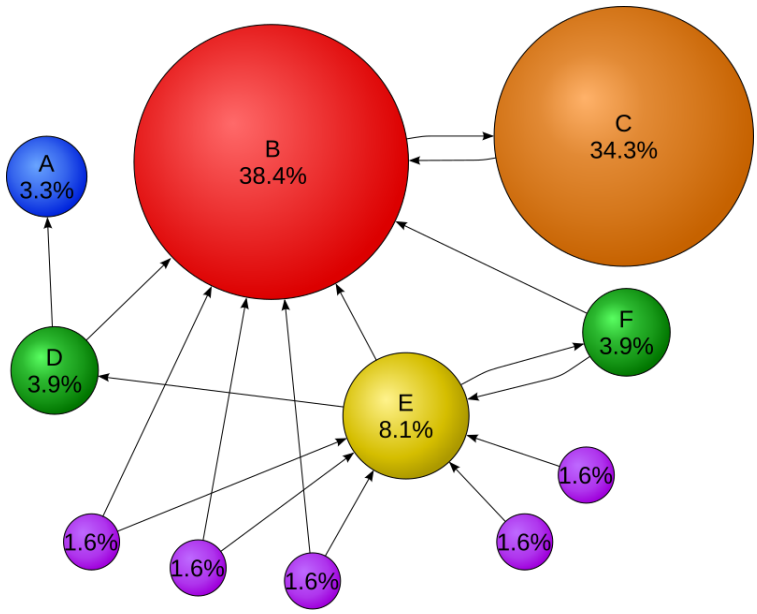
Dessa forma, a relação entre ranqueamento, sumarização e filtragem é intrínseca e complementar, sendo essencial para a geração de resumos automáticos de alta qualidade.

2.4 PageRank

O PageRank é um algoritmo desenvolvido por Larry Page e Sergey Brin, fundadores do Google, para avaliar a importância de páginas na web. A ideia central é introduzir o conceito de *autoridade* de uma página, derivado exclusivamente da estrutura topológica da web, independentemente do conteúdo da página. Esse conceito é análogo ao de citações na literatura científica, onde a relevância de um artigo é frequentemente avaliada pelo número de vezes que é citado por outros trabalhos (PAGE et al., 1999).

No contexto do PageRank, a autoridade de uma página p depende do número de *hyperlinks* que apontam para ela, chamados de *backlinks*, bem como da autoridade das páginas q que referenciam p diretamente. Em outras palavras, uma página é considerada importante se for citada por outras páginas que também são consideradas importantes. Além disso, o PageRank considera que citações seletivas (ou seja, quando uma página q linka para p de maneira específica e não para várias outras páginas simultaneamente) contribuem mais para a autoridade de p do que citações uniformes.

Figura 1 – Ilustração de Autoridade



Fonte: (WIKIPEDIA, 2024)

2.4.1 Modelo Matemático do PageRank

O PageRank é uma forma de medir a importância de uma página na web, levando em conta a estrutura dos links entre as páginas. A ideia central é que uma página é considerada importante se muitas outras páginas importantes apontam para ela.

Matematicamente, o PageRank de uma página p, denotado como $PR(p)$, é calculado usando a seguinte fórmula (PAGE et al., 1999):

$$PR(p) = \frac{1-d}{N} + d \sum_{q \in L(p)} \frac{PR(q)}{C(q)}$$

Onde:

- d é o fator de amortecimento (geralmente definido como 0,85), que representa a probabilidade de um usuário continuar clicando em links.
- N é o número total de páginas na web.
- $L(p)$ é o conjunto de páginas que possuem links para a página p.
- $PR(q)$ é o PageRank da página q.
- $C(q)$ é o número de links de saída da página q.

A fórmula indica que o PageRank de uma página p é composto por um valor base $\frac{1-d}{N}$, que representa a probabilidade de iniciar uma nova navegação em uma página

aleatória, e pela soma do PageRank das páginas que apontam para p, normalizada pelo número de links de saída dessas páginas.

Para resolver o PageRank para todas as páginas, utiliza-se uma abordagem iterativa. Inicialmente, todas as páginas recebem o mesmo valor de PageRank. A cada iteração, o valor de $PR(p)$ é atualizado com base nos valores das páginas que apontam para ela. O processo continua até que o valor do PageRank convirja para um valor estável.

$$PR^{(t)} = \frac{1-d}{N} + dWPR^{(t-1)}$$

Nesta equação:

- W é a matriz de transição, onde $W_{ij} = \frac{1}{C(j)}$ se a página j tem um link para a página i, e $W_{ij} = 0$ caso contrário.
- $PR^{(t)}$ é o vetor de PageRank na iteração t.

Esse processo iterativo garante que o PageRank final seja a distribuição estacionária do sistema, ou seja, a solução para a qual o PageRank converge após várias iterações.

Além disso, ao considerar grafos com pesos, onde cada link pode ter uma "força"diferente, o cálculo do PageRank é ajustado para incorporar esses pesos. Nesse caso, a matriz de transição W é modificada para refletir os pesos dos links entre as páginas:

$$W_{ij} = \frac{w_{ij}}{\sum_k w_{kj}}$$

onde w_{ij} representa o peso do link de j para i.

Essa abordagem permite que o PageRank seja utilizado em uma variedade de contextos, além da web, como na análise de redes sociais e em bioinformática (PAGE et al., 1999).

2.4.2 Vantagens

- **Independência do Conteúdo:** O PageRank baseia-se exclusivamente na estrutura de links da web, independentemente do conteúdo textual das páginas. Isso possibilita que páginas com pouco conteúdo relevante ainda possam alcançar uma alta classificação se forem amplamente referenciadas por outras páginas importantes.
- **Resistência a Manipulações:** O PageRank é relativamente resistente a manipulações, como técnicas de SEO enganosas. A autoridade de uma página é determinada pela qualidade e quantidade de links que ela recebe, o que dificulta a manipulação artificial de sua classificação.

- **Simplicidade e Eficiência Computacional:** O algoritmo é simples de implementar e computacionalmente eficiente, podendo ser calculado por métodos iterativos básicos, como o método das potências. Essa eficiência é especialmente vantajosa para a análise de grandes volumes de dados (LANGVILLE; MEYER, 2006).

- **Versatilidade de Aplicação:** Embora originalmente desenvolvido para classificar páginas da web, o PageRank pode ser adaptado para outras situações, como a sumarização de comentários em aplicativos. No contexto de análise de comentários de usuários, o PageRank pode ser utilizado para identificar quais comentários são mais influentes ou relevantes, com base na estrutura de interações e referências entre os comentários. Isso permite focar nos *feedbacks* mais significativos e amplamente discutidos, auxiliando na priorização e compreensão das opiniões dos usuários.

Diante do grande volume de dados e da necessidade de identificar rapidamente os elementos mais relevantes, o PageRank destaca-se por sua robustez e confiabilidade.

2.5 TextRank

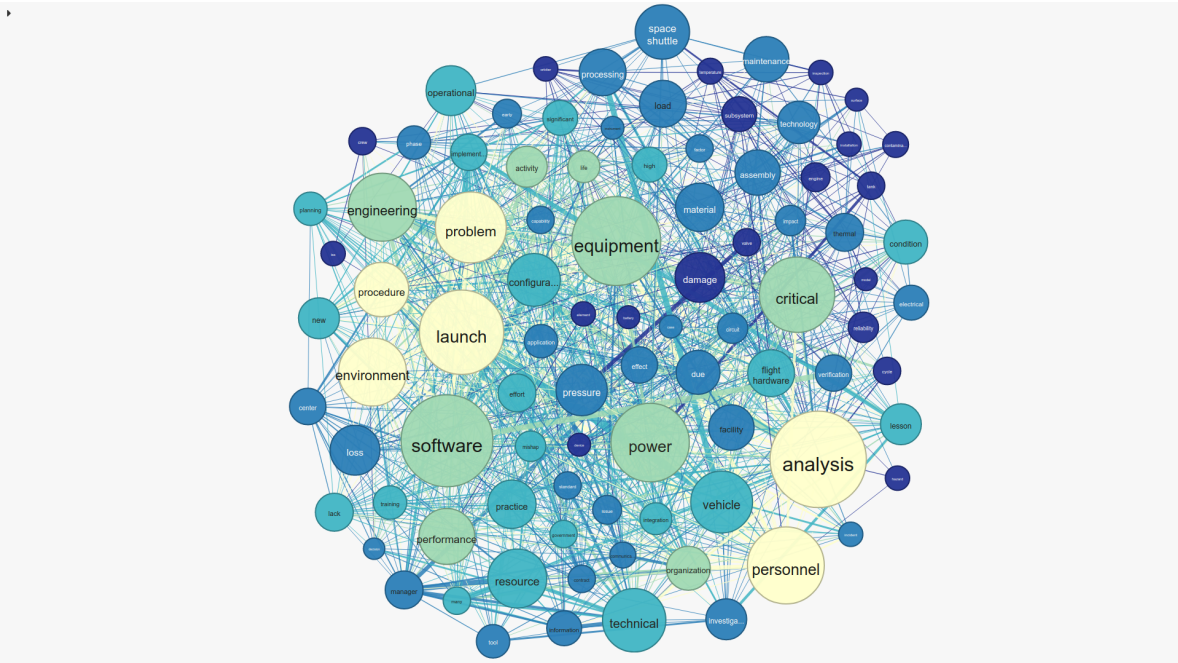
Algoritmos de ranqueamento baseados em grafos, como o HITS de Kleinberg (KLEINBERG, 1999) ou o PageRank do Google (PAGE et al., 1998), têm sido amplamente utilizados em análises de citações, redes sociais e na estrutura de links da Web. Esses algoritmos representam um marco na tecnologia de busca na Web, ao fornecerem um mecanismo de ranqueamento de páginas que se baseia no conhecimento coletivo dos arquitetos da Web, ao invés de se limitar à análise de conteúdo individual de cada página.

Aplicando uma abordagem semelhante a grafos lexicais ou semânticos extraídos de documentos de linguagem natural, resulta em um modelo de ranqueamento que pode ser aplicado em diversas tarefas de processamento de linguagem natural (PLN). Nesse contexto, o conhecimento extraído de um texto completo é utilizado para tomar decisões de ranqueamento ou seleção em nível local. Métodos de ranqueamento orientados para textos podem ser aplicados em tarefas como a extração automatizada de palavras-chave, sumarização extrativa e desambiguação de sentidos das palavras (MIHALCEA; TARAU, 2004a).

O TextRank é diretamente derivado do PageRank, um algoritmo inicialmente desenvolvido para classificar páginas da web com base em sua importância dentro da estrutura de links da internet. Assim como o PageRank, o TextRank utiliza um modelo baseado em grafos para determinar a relevância de elementos específicos, mas, ao invés de aplicar isso a páginas da web, o TextRank é voltado para a análise de textos.

Enquanto o PageRank avalia a importância de uma página com base no número e na qualidade dos links que apontam para ela, o TextRank adapta essa ideia para trabalhar

Figura 2 – Grafo de Palavras



Fonte: (GRAPHAWARE, 2017)

com palavras ou frases dentro de um texto. Em um grafo do TextRank, os nós representam unidades textuais, como palavras ou sentenças, e as arestas entre os nós indicam as relações entre essas unidades, frequentemente baseadas na co-ocorrência dentro de uma determinada janela de contexto.

Diferentemente do PageRank, que é utilizado para medir a relevância de páginas na web, o TextRank é utilizado para identificar elementos relevantes em um texto, como palavras-chave ou sentenças importantes. Essa diferença crucial permite que o TextRank seja aplicado em uma variedade de tarefas de processamento de linguagem natural, como sumarização de textos e extração de palavras-chave, aproveitando a estrutura do texto para derivar significado e relevância de uma maneira que não depende diretamente do conteúdo, mas sim das relações entre as partes do texto.

2.5.1 Modelo Matemático do TextRank

A ideia central do modelo é similar ao conceito de votação. Quando um vértice aponta para outro, ele está essencialmente votando naquele vértice, e o número de votos recebidos contribui para a importância do mesmo.

Formalmente, seja $G = (V, E)$ um grafo direcionado, onde V é o conjunto de vértices (nós) e E é o conjunto de arestas (conexões) entre os vértices. Para um dado vértice $v_i \in V$, seja $In(v_i)$ o conjunto de vértices que apontam para v_i (predecessores), e $Out(v_i)$ o conjunto de vértices para os quais v_i aponta (sucessores).

A pontuação de um vértice v_i no TextRank é dada pela seguinte fórmula básica

(MIHALCEA; TARAU, 2004a):

$$S(v_i) = (1 - d) + d \sum_{v_j \in In(v_i)} \frac{S(v_j)}{|Out(v_j)|}$$

Onde:

- $S(v_i)$ é a pontuação do vértice v_i.
- d é o fator de amortecimento (ou *damping factor*), que geralmente é definido como 0,85, e controla a probabilidade de seguir os links no grafo.
- $In(v_i)$ é o conjunto de vértices que apontam para v_i.
- $Out(v_j)$ é o número de arestas saindo do vértice v_j.

Para incluir pesos nas arestas do grafo, a fórmula é ajustada para considerar a força da conexão entre os vértices. Seja w_{ji} o peso da aresta que conecta o vértice v_j ao vértice v_i. A fórmula modificada é (MIHALCEA; TARAU, 2004a):

$$S(v_i) = (1 - d) + d \sum_{v_j \in In(v_i)} \frac{w_{ji} \cdot S(v_j)}{\sum_{v_k \in Out(v_j)} w_{jk}}$$

Neste caso:

- w_{ji} representa o peso da aresta entre v_j e v_i, refletindo a importância ou força da conexão.
- A soma no denominador $\sum_{v_k \in Out(v_j)} w_{jk}$ normaliza a contribuição de $S(v_j)$ com base no peso total das arestas saindo de v_j.

O algoritmo começa com valores arbitrários atribuídos a cada nó do grafo e itera até que a convergência abaixo de um determinado limiar seja alcançada. Ao final, cada vértice recebe uma pontuação que reflete sua importância dentro do grafo, sendo que o uso de pesos permite um refinamento na análise da relevância dos vértices.

2.5.2 Extração de Palavras-Chave com TextRank

O TextRank é uma técnica eficiente para a extração de palavras-chave de um texto, identificando as palavras ou frases mais representativas de um documento em linguagem natural. A extração de palavras-chave com TextRank envolve os seguintes passos:

1. **Construção do Grafo:**

- As unidades a serem classificadas são sequências de uma ou mais palavras, que formam os vértices de um grafo textual. Qualquer relação definida entre duas palavras pode ser uma conexão útil (aresta) adicionada entre dois vértices.
- Utiliza-se uma relação de coocorrência, controlada pela distância entre as ocorrências das palavras: dois vértices são conectados se as palavras correspondentes coocorrerem dentro de uma janela de N palavras, onde N pode variar de 2 a 10 palavras (MIHALCEA; TARAU, 2004a).

2. **Filtragem Sintática:**
   - As palavras adicionadas ao grafo podem ser filtradas sintaticamente, selecionando apenas aquelas que pertencem a certas classes gramaticais. Por exemplo, pode-se considerar apenas substantivos e adjetivos para inclusão no grafo, e assim desenhar arestas baseadas apenas nas relações estabelecidas entre essas classes.
3. **Tokenização e Anotação:**
   - Primeiramente, o texto é tokenizado e anotado com as classes gramaticais – uma etapa de pré-processamento necessária para aplicar os filtros sintáticos.
   - Para evitar o crescimento excessivo do grafo, são consideradas apenas palavras individuais como candidatas a serem adicionadas ao grafo. Palavras-chave compostas são eventualmente reconstruídas na fase de pós-processamento.
4. **Execução do Algoritmo de Ranqueamento:**
   - Após a construção do grafo (grafo não direcionado e não ponderado), o algoritmo de ranqueamento é executado por várias iterações até convergir – geralmente entre 20 e 30 iterações, com um limiar de 0,0001.
   - Uma vez obtida a pontuação final para cada vértice no grafo, os vértices são ordenados em ordem decrescente de sua pontuação, e os mais bem classificados são retidos para pós-processamento (MIHALCEA; TARAU, 2004a).
5. **Pós-Processamento:**
   - Durante o pós-processamento, todas as palavras selecionadas como palavras-chave em potencial pelo algoritmo TextRank são marcadas no texto.
   - Sequências de palavras-chave adjacentes são combinadas em uma palavra-chave composta. Por exemplo, em um texto sobre "O aplicativo é muito ruim", se "Aplicativo"e "Ruim"forem selecionados como palavras-chave potenciais pelo TextRank, como são adjacentes, elas seriam combinadas em uma única palavra-chave "Aplicativo ruim".

2.6 LexRank

Nos últimos anos, o processamento de linguagem natural (PLN) tem evoluído significativamente, com muitos problemas sendo abordados por técnicas estatísticas robustas. Entre essas técnicas, métodos baseados em grafos vêm ganhando destaque, sendo utilizados em tarefas como *clustering* de palavras, onde o foco é especificamente em sumarização extrativa com o objetivo de produzir um resumo de documentos sobre um tema comum, mas não especificado.

A sumarização extrativa produz resumos ao selecionar um subconjunto das sentenças dos documentos originais. Isso contrasta com a sumarização abstrativa, na qual a informação no texto é reformulada. Embora resumos produzidos por humanos tipicamente não sejam extrativos, a maioria das pesquisas em sumarização hoje se concentra nesse tipo, devido à complexidade das técnicas abstrativas, que envolvem representação semântica, inferência e geração de linguagem natural.

2.6.1 Centralidade de Sentenças e Sumarização Baseada em Centroides

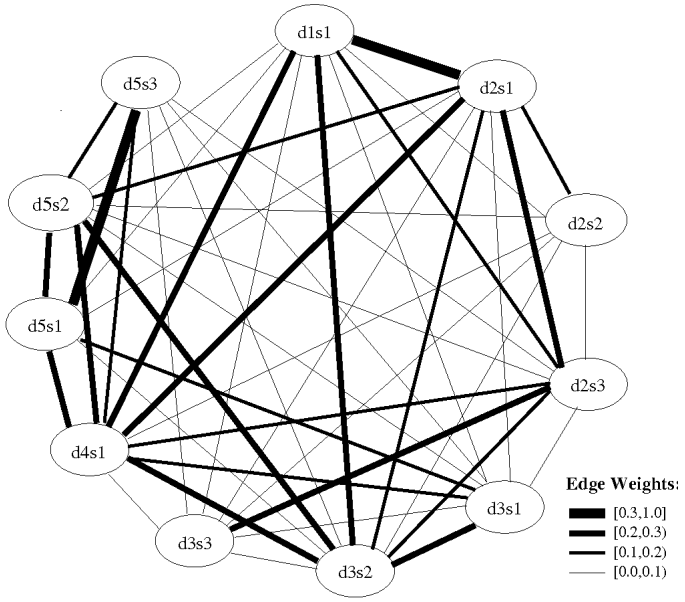
A centralidade de uma sentença é frequentemente definida em termos da centralidade das palavras que ela contém. Uma forma comum de avaliar a centralidade das palavras é utilizando o centroide do *cluster* de documentos em um espaço vetorial. O centroide de um *cluster* é um pseudo-documento que consiste em palavras que possuem pontuações TF-IDF acima de um determinado limiar, onde *TF* é a frequência de uma palavra no *cluster*, e *IDF* a frequência inversa do documento. Na sumarização baseada em centroides, as sentenças que contêm mais palavras do centroide do *cluster* são consideradas centrais. Esse método tem se mostrado promissor, resultando em sistemas de sumarização de múltiplos documentos amplamente utilizados (ERKAN; RADEV, 2004b).

2.6.2 Centralidade Baseada em Grafos

A centralidade de uma sentença em um *cluster* de documentos pode ser vista como uma medida de sua importância com base na similaridade com outras sentenças no *cluster*. Para definir a similaridade entre duas sentenças, utilizamos o modelo de *bag-of-words* (Wikipedia, 2024), representando cada sentença como um vetor em um espaço N-dimensional, onde N é o número de palavras possíveis na língua alvo. A similaridade entre duas sentenças é então definida pelo cosseno entre os vetores correspondentes (ERKAN; RADEV, 2004b).

$$\text{cosseno-idf}(x, y) = \frac{\sum_{w \in x,y} \text{tf}_{w,x} \cdot \text{idf}_w \cdot \text{tf}_{w,y} \cdot \text{idf}_w}{\sqrt{\sum_{x_i \in x} (\text{tf}_{x_i,x} \cdot \text{idf}_{x_i})^2} \cdot \sqrt{\sum_{y_i \in y} (\text{tf}_{y_i,y} \cdot \text{idf}_{y_i})^2}} \tag{2.1}$$

Figura 3 – Similaridades de cosseno em grafo



Fonte: (ERKAN; RADEV, 2004b)

2.6.2.1 Centralidade de Grau

Uma abordagem simples para avaliar a centralidade de uma sentença é contar o número de outras sentenças similares a ela. Definimos a centralidade de grau de uma sentença como o grau do nó correspondente no grafo de similaridade. No entanto, o limiar de similaridade utilizado para construir esse grafo pode influenciar significativamente os resultados, com limiares baixos incluindo similaridades fracas e limiares altos potencialmente eliminando relações relevantes(ERKAN; RADEV, 2004b).

2.6.2.2 Centralidade por Autovalores e LexRank

No método de centralidade de grau, cada aresta é tratada como um voto para determinar o valor central de cada nó, porém, essa abordagem democrática pode ser problemática em comentários, onde nem todas as relações têm a mesma importância. No contexto da sumarização, isso significa que sentenças irrelevantes podem acabar sendo incluídas no resumo. Para resolver esse problema, propomos ponderar os votos com base na centralidade dos nós votantes. Esse conceito é formalizado pela equação(ERKAN; RADEV, 2004b):

$$p(u) = \sum_{v \in adj[u]} \frac{p(v)}{deg(v)} \tag{2.2}$$

onde $p(u)$ é a centralidade do nó u, $adj[u]$ é o conjunto de nós adjacentes a u, e $deg(v)$ é o grau do nó v. Esta formulação pode ser vista como uma adaptação do algoritmo PageRank, onde a centralidade de cada nó é distribuída entre seus vizinhos, considerando a importância de cada nó vizinho.

2.6.3 Implementação do LexRank

Para garantir que o grafo de similaridade seja irredutível e aperiódico, adicionamos uma pequena probabilidade de salto para qualquer nó no grafo, permitindo que um caminhante aleatório escape de componentes periódicos ou desconectados. A equação resultante é(ERKAN; RADEV, 2004b):

$$p(u) = \frac{d}{N} + (1-d) \sum_{v \in adj[u]} \frac{p(v)}{deg(v)} \tag{2.3}$$

onde N é o número total de nós no grafo e d é um fator de amortecimento. Em termos de matriz, essa equação pode ser expressa como:

$$\mathbf{p} = [d\mathbf{U} + (1-d)\mathbf{B}]^{\mathrm{T}} \mathbf{p} \tag{2.4}$$

onde $\mathbf{U}$ é uma matriz quadrada com todos os elementos iguais a $1/N$.

2.6.4 Comparação com Outras Abordagens

O LexRank, ao ponderar a centralidade das sentenças com base na centralidade de suas sentenças adjacentes, supera métodos baseados em centroides, especialmente em *clusters* de documentos onde há muita variação temática ou ruído. Os resultados mostram que, em testes com os dados das Conferências de Entendimento de Documentos (DUC), o LexRank apresenta um desempenho superior à abordagem de centroides em diversos casos.

2.6.5 Conclusão

O LexRank oferece uma abordagem robusta para sumarização de textos baseada em grafos, permitindo a identificação das sentenças mais centrais em um *cluster* de documentos. Ao incorporar conceitos de centralidade de autovalores, o LexRank é capaz de superar as limitações de métodos mais simples, como a centralidade de grau, fornecendo resumos mais precisos e representativos do conteúdo original.

2.7 Técnicas Modernas de Sumarização

Com o avanço das redes neurais e dos modelos de linguagem, surgiram técnicas modernas de sumarização que superam abordagens tradicionais. Entre essas técnicas, destacam-se modelos baseados em transformadores, como o BERTSUM, o T5 e o PEGASUS.

2.7.1 BERTSUM

O BERTSUM é uma extensão do modelo BERT, projetada para tarefas de sumarização. Ele incorpora camadas adicionais de transformadores inter-sentença para capturar relações entre sentenças, permitindo a geração de resumos mais coerentes e coesos (LIU; LAPATA, 2019).

2.7.2 T5

O modelo T5 (Text-to-Text Transfer Transformer) adota uma abordagem unificada, convertendo todas as tarefas de processamento de linguagem natural em um formato de entrada-texto para saída-texto. Essa flexibilidade permite que o T5 seja aplicado a diversas tarefas, incluindo a sumarização, alcançando resultados de ponta (RAFFEL et al., 2020).

2.7.3 PEGASUS

O PEGASUS foi especificamente projetado para a sumarização abstrativa. Durante seu pré-treinamento, sentenças importantes são mascaradas no documento de entrada e o modelo é treinado para gerá-las como uma sequência de saída, semelhante a um resumo extrativo. Essa abordagem resulta em um desempenho superior em múltiplas tarefas de sumarização (ZHANG et al., 2020).

2.7.4 Comparativo entre Abordagens Baseadas em Grafos e Deep Learning

As técnicas tradicionais de sumarização, como o LexRank e o TextRank, utilizam algoritmos baseados em grafos para identificar sentenças centrais em um texto. Embora sejam métodos mais simples em comparação com modelos de deep learning, eles apresentam diversas vantagens práticas.

Modelos baseados em transformadores, como o PEGASUS, o T5 e o BERTSUM, oferecem resultados superiores na sumarização abstrativa, capturando a semântica profunda do texto e gerando resumos mais naturais e contextualizados (ZHANG et al., 2020). No entanto, essas técnicas possuem desafios significativos:

- **Alto custo computacional**: Modelos de deep learning exigem grande poder de processamento para treinamento e inferência, tornando sua aplicação inviável em cenários com recursos limitados.
- **Dependência de grandes volumes de dados**: Redes neurais precisam de grandes conjuntos de dados rotulados para alcançar bom desempenho, o que pode ser um entrave na aplicação prática.

- **Baixa interpretabilidade**: Enquanto métodos baseados em grafos, como o LexRank, permitem uma análise clara de como as sentenças foram selecionadas, os modelos de deep learning funcionam como caixas-pretas, dificultando a explicação dos resultados.
- **Complexidade na implementação**: Implementar e ajustar modelos de deep learning requer conhecimento especializado e tempo considerável para experimentação e otimização.

Por outro lado, as abordagens baseadas em grafos oferecem uma alternativa eficiente e prática, especialmente para a sumarização extrativa de comentários curtos. O LexRank, utilizado neste trabalho, apresenta vantagens importantes:

- **Baixo custo computacional**: O LexRank pode ser executado rapidamente em conjuntos de dados grandes sem necessidade de GPUs ou servidores potentes.
- **Facilidade de implementação**: Sua implementação em bibliotecas como *networkx* e *lexrank* é direta e requer menos ajustes em comparação com modelos de deep learning.
- **Robustez para textos curtos**: Comentários de usuários geralmente são curtos e diretos, tornando a sumarização extrativa uma abordagem suficiente para capturar informações relevantes sem necessidade de reformulação textual.
- **Maior interpretabilidade**: Como o ranqueamento é baseado na estrutura do grafo de similaridade entre sentenças, os critérios para a seleção de frases são transparentes e fáceis de justificar.

Dessa forma, apesar dos avanços proporcionados pelos modelos de deep learning, o LexRank foi escolhido por oferecer um equilíbrio ideal entre eficiência computacional, simplicidade de implementação e interpretabilidade, tornando-se uma solução viável para a sumarização automática de comentários de usuários no contexto deste estudo.

2.8 Filtragem Baseada em Características Linguísticas

A filtragem de comentários antes da sumarização desempenha um papel fundamental na melhoria da qualidade dos resumos gerados. A remoção de textos irrelevantes ou com baixa informatividade permite que o processo de sumarização se concentre nas sentenças mais representativas. A biblioteca **textstat** é utilizada para calcular métricas de legibilidade, permitindo a filtragem de textos mal estruturados ou difíceis de compreender (BANSAL, 2024).

2.8.1 Biblioteca textstat para Avaliação da Legibilidade

A biblioteca *textstat* é uma ferramenta amplamente utilizada para a análise de legibilidade de textos. Seu funcionamento baseia-se em métricas estatísticas que quantificam a complexidade textual, permitindo a identificação de textos de difícil compreensão ou que apresentem baixa informatividade. Essa abordagem é essencial para a filtragem de comentários, visto que textos excessivamente curtos, mal estruturados ou com vocabulário inadequado podem comprometer a qualidade da sumarização.

2.8.1.1 Métricas de Legibilidade

A *textstat* fornece diversas métricas para avaliação da complexidade textual. As principais utilizadas na filtragem são:

- **Índice de Flesch** (*Flesch Reading Ease*): Mede a facilidade de leitura do texto com base no número médio de sílabas por palavra e palavras por sentença. Quanto maior a pontuação, mais fácil é a leitura do texto (FLESCH, 1948).
- **Índice de Dale-Chall**: Analisa a proporção de palavras complexas no texto, utilizando uma lista pré-definida de palavras fáceis para calcular a dificuldade de leitura (DALE; CHALL, 1948).
- **Contagem de sentenças e palavras**: Permite filtrar textos excessivamente curtos, que podem não conter informações suficientes para uma sumarização eficaz.

Essas métricas são aplicadas de maneira eficiente, permitindo que a filtragem ocorra antes da etapa de sumarização, garantindo que apenas textos legíveis e informativos sejam utilizados como entrada para o LexRank.

2.8.2 Justificativa da Escolha

A seleção dessa biblioteca foi motivada por suas características que facilitam a integração com o LexRank:

- **Eficiência computacional**: É leve e otimizada, permitindo a análise rápida de grandes volumes de textos sem necessidade de processamento intensivo, ao contrário de modelos baseados em redes neurais (MANNING; RAGHAVAN; SCHüTZE, 2008).
- **Compatibilidade com LexRank**: O LexRank opera sobre a estrutura de similaridade entre sentenças. A filtragem baseada em características linguísticas melhora essa estrutura ao remover sentenças com ruído, garantindo que a análise de similaridade seja aplicada apenas sobre textos relevantes (ERKAN; RADEV, 2004a).

- **Facilidade de implementação**: Possui suporte nativo para manipulação de texto em Python e podem ser integradas ao pipeline de pré-processamento sem necessidade de etapas adicionais de treinamento ou ajuste fino de hiperparâmetros (PEDREGOSA et al., 2011).
- **Melhoria na representatividade do resumo**: A remoção de comentários genéricos ou com baixo conteúdo informativo reduz a redundância no conjunto de sentenças consideradas pelo LexRank, aprimorando a diversidade dos resumos gerados (JOACHIMS, 1998b).

3 Materiais e Métodos

Este capítulo apresenta as ferramentas e metodologias empregadas para a sumarização de comentários do aplicativo “Gov.Br”, detalhando as tecnologias utilizadas e os procedimentos realizados. Conforme abordado anteriormente, as técnicas de sumarização aplicadas envolvem os algoritmos LexRank, TextRank e PageRank. A linguagem de programação Python (Python, 2024) foi escolhida para a implementação, devido à sua vasta coleção de bibliotecas voltadas ao processamento de linguagem natural e manipulação de dados, o que facilitou tanto a coleta dos comentários quanto o uso dos algoritmos necessários.

3.1 Classificação e Plano Metodológico

Metodologicamente, este trabalho é classificado como um estudo de caso experimental. Ele se concentra na aplicação de técnicas de Processamento de Linguagem Natural (PLN) e algoritmos de sumarização de texto em um conjunto específico de dados: os comentários de usuários do aplicativo Gov.Br. As principais características que definem sua classificação são:

- Estudo de Caso: O foco está em um cenário real e específico (comentários do aplicativo Gov.Br), onde há uma necessidade de processar e resumir grandes volumes de dados textuais. O estudo de caso tem como objetivo analisar e otimizar a interação dos usuários com o app por meio da análise automatizada de feedbacks.
- Abordagem Experimental: O trabalho aplica um pré-processando utilizando técnicas de filtragem e também a sumarização com o algoritmo LexRank. Esse caráter experimental envolve a implementação de soluções automatizadas para a coleta, filtragem e sumarização dos comentários e a validação da eficácia dessas técnicas em fornecer resumos representativos e informativos.
- Metodologia Quantitativa: Embora se trate de um estudo qualitativo em termos de análise de textos, a metodologia adotada utiliza dados numéricos e algoritmos para gerar resumos, o que insere o trabalho em uma abordagem quantitativa no processamento e avaliação dos resultados.

Dessa forma, metodologicamente, o trabalho combina aspectos de estudo de caso e experimentação com um foco na aplicação de técnicas quantitativas de PLN .

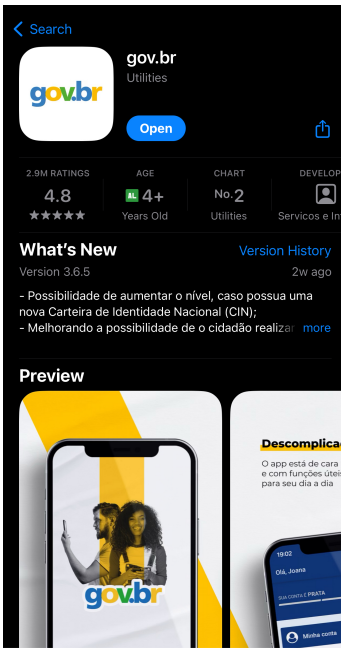
3.2 Fonte de Dados

3.2.1 Gov.Br

Para a obtenção dos comentários que foram utilizados no experimento, foi escolhida a base de dados do aplicativo "Gov.Br" (GOV. . . , 2020). Este aplicativo, desenvolvido pelo governo brasileiro, visa centralizar e facilitar o acesso dos cidadãos aos serviços oferecidos por diferentes órgãos governamentais.

A escolha do "Gov.br" como fonte de dados se deve ao seu impacto significativo na vida dos brasileiros, além de ser um dos aplicativos mais utilizados, com milhões de downloads e avaliações. Essas características tornam os comentários dos usuários uma rica fonte de informações, ideal para a aplicação e teste dos algoritmos de sumarização. No momento da coleta, o aplicativo contava com uma nota média de 4,8 estrelas e aproximadamente 2,9 milhões de avaliações na *App Store*, destacando-se por sua popularidade e relevância.

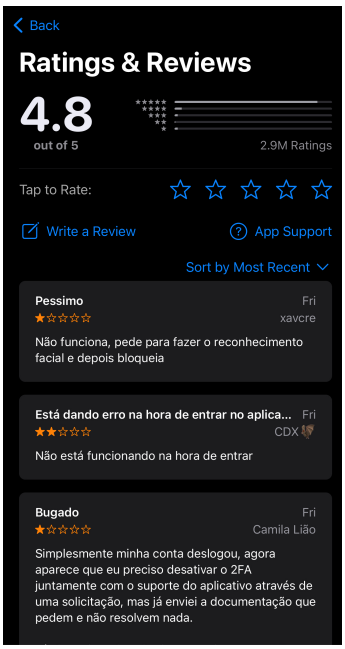
Figura 4 – Página da aplicação (Fonte: Autor)



3.2.2 Coleta de Dados

Para realizar a coleta dos comentários utilizados na sumarização dos textos, foi desenvolvido um script em Python que fez uso da biblioteca `google-play-scraper` (JoMingyu, 2019), a qual permite a extração de avaliações de usuários de aplicativos disponíveis na *Google Play Store* (GOOGLE, 2024). Além disso, as bibliotecas `Matplotlib` (HUNTER, 2007), `numpy` (NumPy, 2006) e `pandas` (Pandas, 2009) foram utilizadas para tornar esse processo mais visual.

Figura 5 – Avaliações da aplicação (Fonte: Autor)



O processo de extração foi conduzido de forma a obter as 200 avaliações mais recentes do aplicativo *Gov.br* para cada nível de avaliação (variando de 1 a 3 estrelas), totalizando assim 600 comentários. Esse número foi escolhido, pois limita o espaço amostral a comentários mais recentes, já que comentários mais antigos podem não ser mais relevantes. Como o objetivo da aplicação é ajudar os desenvolvedores a identificar os principais pontos de melhoria, foi limitado a nota dos comentários a 3 estrelas, já que normalmente comentários com menor nota contém mais críticas do que os de nota mais alta. Após a coleta, os dados foram estruturados em uma tabela e exportados para um arquivo de planilha no formato `.csv`. Cada linha da planilha resultante contém a avaliação numérica, o usuário responsável, o texto do comentário e a data em que a avaliação foi realizada.

Figura 6 – Planilha de comentários

Nota	Autor	Data	Comentário
1	Alex Sandro da Silva Silva	31/08/2024 17:21	Ruim demais
1	EVANDRO VIANA BARROSO	31/08/2024 17:14	Por favor arrumem esse aplicativo horrível pois estou sem ter acesso ao aplicativo da carteira de trabalho digital ok Pois só estar aparecendo um nome de duas etapas que tem que desativar isso estar aparecendo desde de
1	Alexandre Ferreira	31/08/2024 17:11	aplicativo não está mais funcionando como antes fica dando erro
1	Elisete Viana	31/08/2024 17:11	Cada vez pior,, prefiro a carteira de papel aff
1	Luiz YoShida	31/08/2024 17:10	Horrível.
1	Louise Alcântara	31/08/2024 16:49	Simplesmente troquei de celular e o app não quer mais abrir, ponho CPF da erro. Como vou ter acesso a minha carteira de trabalho? Meu FGTS??! Etc .. estou desesperada. Dando erro de captha sempre que ponho CPF.
1	Vicente Chuco neto	31/08/2024 16:35	Não é bom não
1	Bruna Kern Donamore	31/08/2024 16:33	O reconhecimento facial esta MUITO AQUÉM do que deveria ser, nunca consegui entra por ele. Agora da erro ate p entrar pelo banco, estou tendo que assinar os documentos e reconhecer firma no cartório, como os incas
1	Mariana Torres	31/08/2024 16:28	Esse é o pior app que já usei na vida. Já precisei de tirar RG online e não consegui, perdi um momento de emergência por causa disso. Preciso tirar minha Carteira de Trabalho Digital e mais uma vez fiquei na mão...
1	Elisângela Muniz	31/08/2024 15:59	Péssimo
1	Lucineia Silva Ferreira Domingues	31/08/2024 15:50	Ha vários dias tenho tentado acesso a minha conta e dá erro.O que houve????
1	Felipe Martins	31/08/2024 15:47	Não consigo acessar o gov sempre da erro no login. (ERL0000600) e preciso acessar com urgência
1	ɱαιиα ɱσяαєs	31/08/2024 15:16	não consigo entrar no site nem pelo app sempre dá erro ERL0032600
1	Roberto Bento	31/08/2024 15:16	Péssimo, reconhecimento facial não funciona.
1	Claudio Ribeiro	31/08/2024 15:15	Não está conectado
1	Edson Madeira	31/08/2024 14:50	Preciso ajuda
1	Queops Freitas	31/08/2024 14:40	Nunca tive problema de acesso a minha conta gov mas agora quando ponho o CPF pra acesse, aparece a mensagem : sistema de captcha indisponível no momento (ERL0000600), n faço ideia do q seja, concertem isso.
1	Caio Genesi	31/08/2024 14:40	péssimo aplicativo! antes era compatível com meu celular redmi note 11 e agora não é mais
1	Maria G C Thuler	31/08/2024 14:30	Captchar não funciona. Nunca aceitam a foto do documento para redefir a senha e aquela palhaçada de tirar foto segurando um papel com a data do dia e com a foto do documento na mesma foto que nunca aceitam tamb
1	Edson Aparecido da Silva Silva	31/08/2024 14:27	Olá boa tarde estava até bom o app mas ficou travando agora que preciso de vincular de novo não funciona
1	TERESA MELO	31/08/2024 14:22	Péssimo. Pede para reconhecer figuras e por mais que se clique diz que não esta disponível. Preciso agendar pericia e somente pelo aplicativo. Vou ficar com falta?
1	Alfredo Kohler	31/08/2024 13:56	Simplesmente não funciona , não reconhece nem meu CPF cadastrado a anos. Voltamos a Idade da pedra no País dos sonhos
1	lucxsz	31/08/2024 13:54	nunca vi aplicativo tão ruim, difícil acessibilidade, não sei como que integram o sistema de login com o gov, é péssimo para entrar em vários aplicativos, péssimo mesmo, toda hora sistema fora do ar, complicado usar isso.
1	Alyson Maranhão	31/08/2024 13:40	Não consigo fazer o login pelo aplicativo e muito menos o reconhecimento facial.
1	Alone Oficial BR	31/08/2024 13:36	Tanto dinheiro público jogado fora tô tentando resolve um problema para meu pai neste lixo e não funciona direito.
1	Guilherme Sena de Aguiar	31/08/2024 13:03	Não consigo acessar devido ao modo desenvolvedor, sou desenvolvedor de jogos e uso meu celular como ferramenta de testes e o app implica em desativar o modo desenvolvedor.
1	Luísa Ferreira	31/08/2024 12:21	tenho nada a falar sobre este aplicativo

Fonte: Autor

3.3 Filtragem de Comentários

Antes da etapa de sumarização, é necessário realizar uma filtragem dos comentários para remover textos irrelevantes, redundantes ou de baixa qualidade. Esse processo melhora a coerência e a representatividade do conjunto de dados, garantindo que apenas comentários relevantes sejam utilizados na geração dos resumos.

3.3.0.1 Ferramenta Utilizada

A filtragem foi implementada utilizando a biblioteca *textstat*, que permite a extração de métricas linguísticas e a análise estatística dos comentários. Ela foi utilizada para calcular índices de legibilidade, como o índice de Flesch e o índice de Dale-Chall, permitindo a remoção de textos com baixa compreensibilidade (BANSAL, 2024).

3.3.0.2 Processo de Filtragem

O processo de filtragem ocorre em três etapas principais:

1. **Pré-processamento 1**: Remoção de comentários que possuem um número de palavras não satisfatório.
2. **Pré-processamento 2**: Remoção de caracteres especiais, normalização do texto e tokenização.
3. **Avaliação de legibilidade**: Aplicação do *textstat* para calcular índices de legibilidade e descartar textos que não atingem um limite mínimo de compreensão.

A filtragem garante que apenas os comentários mais relevantes sejam utilizados na etapa de sumarização, melhorando a qualidade dos resumos gerados.

3.4 Sumarização de Comentários com LexRank

A sumarização de comentários tem como objetivo reduzir a quantidade de texto, preservando as informações mais relevantes. O método utilizado neste trabalho é o **LexRank**, uma abordagem baseada em grafos que identifica as sentenças mais centrais do conjunto de dados por meio de medidas de similaridade lexical (ERKAN; RADEV, 2004a).

O LexRank constrói um grafo onde cada nó representa uma sentença e as arestas entre os nós são ponderadas de acordo com a similaridade entre elas. A centralidade de cada sentença no grafo é então utilizada para selecionar as frases mais representativas para compor o sumário.

3.4.1 Processo de Sumarização

O processo de sumarização ocorre em três etapas principais:

1. **Construção do grafo**: Cada comentário é representado como um nó em um grafo, e as conexões entre os nós são definidas com base na similaridade entre os textos.
2. **Cálculo de centralidade**: Aplicação do algoritmo LexRank para determinar quais comentários possuem maior importância dentro do conjunto analisado.
3. **Seleção das sentenças mais relevantes**: As sentenças com maior grau de centralidade são escolhidas para compor o sumário final.

A aplicação dessas etapas permite obter um resumo coerente, representativo e não redundante dos comentários filtrados.

4 Resultados

Nesta seção, são apresentados os resultados experimentais da sumarização extrativa aplicada aos comentários do aplicativo Gov.Br. O objetivo da análise é avaliar o impacto da variação de diferentes parâmetros disponíveis no código, mantendo os demais fixos nos valores padrão. A análise comparativa permite compreender como cada parâmetro influencia a qualidade da filtragem e da sumarização.

Os parâmetros fixos são:

- **count_por_nota**: número de comentários por nota (200).
- **nota_maxima**: nota máxima considerada na análise (3).
- **summary_size**: tamanho máximo dos sumários gerados (5).

Os parâmetros variáveis manterão um valor intermediário pra cada cenário enquanto o parâmetro analisado é alterado, dentre eles temos:

- **summary_threshold**: valor intermediário 0.5.
- **flesch_threshold**: valor intermediário 50.
- **min_words**: valor intermediário 8.

4.1 Impacto do summary_threshold

O parâmetro **summary_threshold** define o limiar mínimo de similaridade para que uma sentença seja considerada relevante na sumarização pelo LexRank. Um valor menor permite que mais sentenças sejam incluídas, enquanto um valor maior restringe a seleção às sentenças mais centrais. Esse valor pode variar de 0 a 1.

4.1.1 Limiar baixo (0.05)

Com **summary_threshold** = 0.05, um maior número de sentenças foi incluído nos resumos, resultando em textos mais extensos. Esse ajuste favorece uma cobertura maior dos comentários, mas pode reduzir a concisão dos sumários.

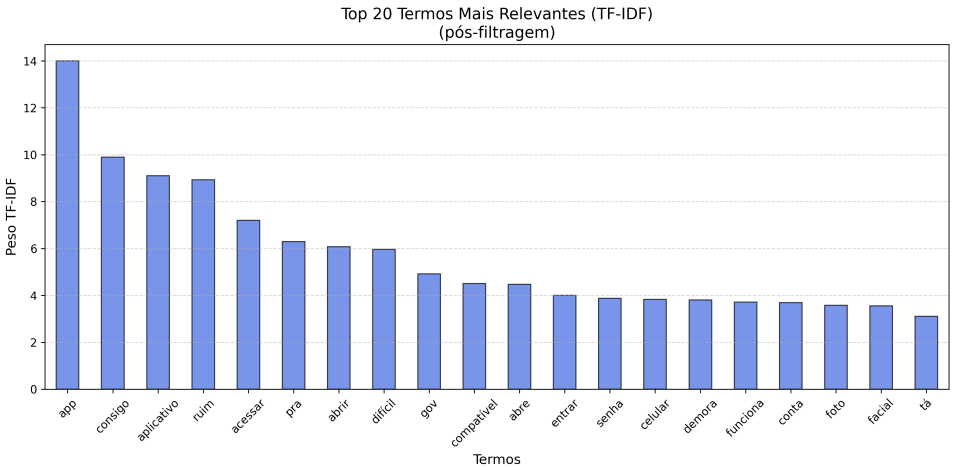


Figura 7 – Principais termos identificados pelo TF-IDF para **summary_threshold** = 0.05.

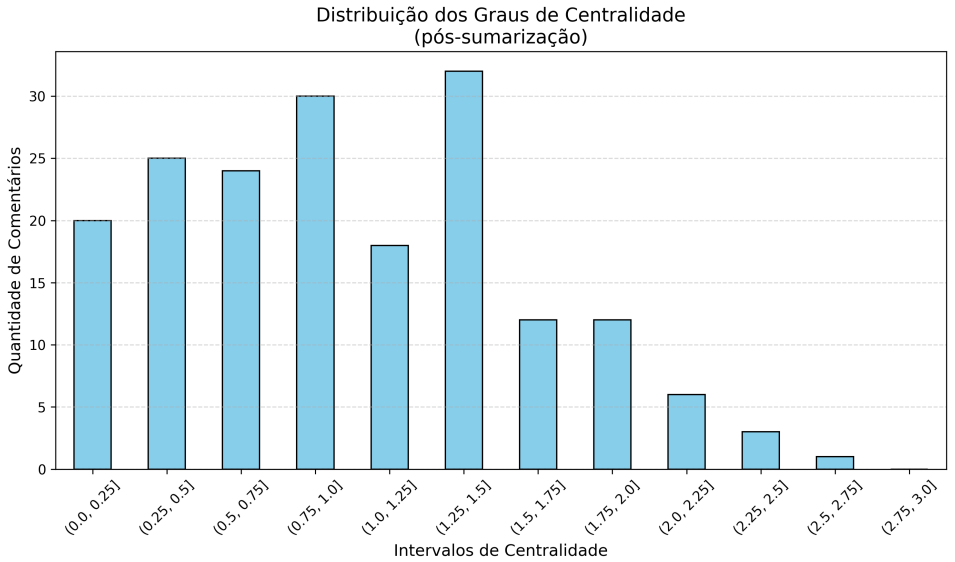


Figura 8 – Distribuição dos scores dos comentários antes e depois da filtragem para **summary_threshold** = 0.05.

Sumário	**Texto**
1	Não consigo baixar em meu celular. não aparece nem Playstore tá uma m já tô bem loco e fala que o app não é compatível com o Cel .e pior que tenho que ter esse gov pra acessar todos os app do governo tá difícil.
2	App parou de funcionar no meu smartphone há muito tempo. Não consigo baixar mais, mensagem de incompatibilidade com o dispositivo. Eu tinha validação em 2 etapas e não consigo resolver nada que preciso porque sempre pede pra entrar pelo gov. Mas não tenho acesso. Aff
3	Tenho o app a um tempo , mas agora não consigo abrir. Pois tem uma mensagem que diz que meu celular não é compatível. Até o ano passado era. Como vou usar?
4	Pelo amor de Deus, como um aplicativo importantíssimo não e compatível com os celulares? E outra cheio de problemas dentro app, o meu não consigo cadastrar a mais de 8 messes.
5	Tá difícil resolver as coisas não vai de jeito nenhum o código de acesso saiu do meu app não consigo resolver a ideia e boa juntar tudo num appas e muito burocrático da certo não.

Tabela 1 – Sumários gerados para **summary_threshold** = 0.05.

4.1.2 Limiar intermediário (0.5)

Com **summary_threshold** = 0.5, um bom número de sentenças foi incluído nos resumos, para equilibrar textos mais extensos com a concisão dos sumários.

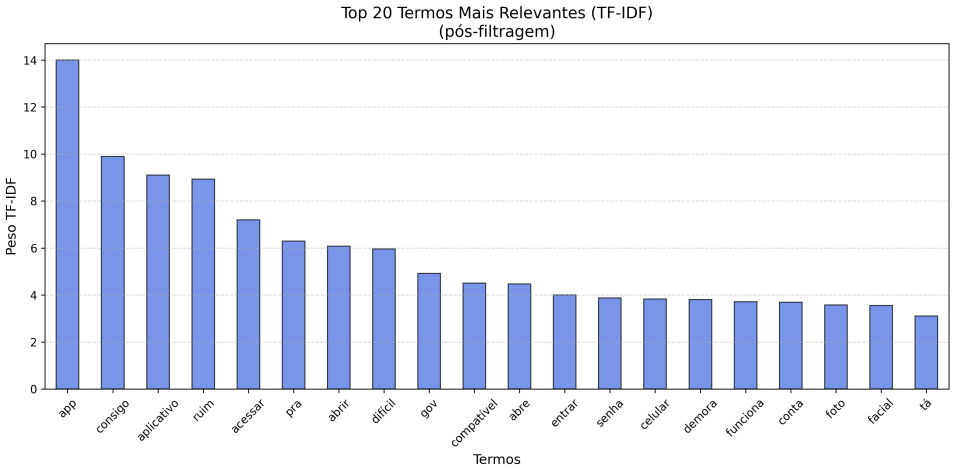


Figura 9 – Principais termos identificados pelo TF-IDF para **summary_threshold** = 0.5.

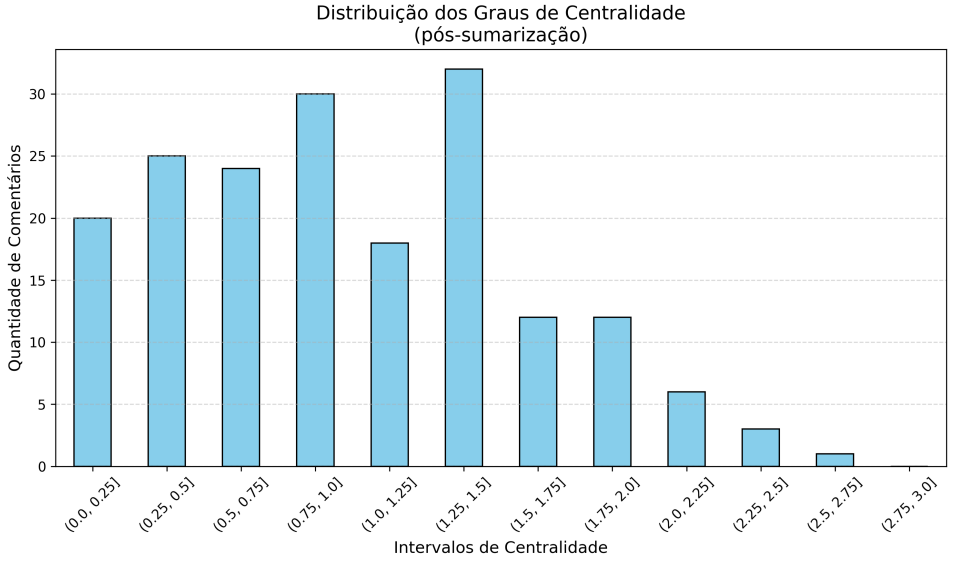


Figura 10 – Distribuição dos scores dos comentários antes e depois da filtragem para **summary_threshold** = 0.5.

Sumário	Texto
1	Não consigo acessar meu app do gov.br.
2	tá muito ruim pra acessar o app.
3	Não consigo abrir o app
4	muito bom o aplicativo
5	É um app palhaçada. Lhe obrigam a instalar o app, talvez uma forma de controlar onde você vai. Agora exigem 2 etapas e o app... E exigem que o modo desenvolver esteja desativado. Lixo de app

Tabela 2 – Sumários gerados para **summary_threshold** = 0.5.

4.1.3 Limiar Alto (0.9)

Com **summary_threshold** = 0.9, um menor número de sentenças foi incluído nos resumos, priorizando mais a concisão dos sumários.

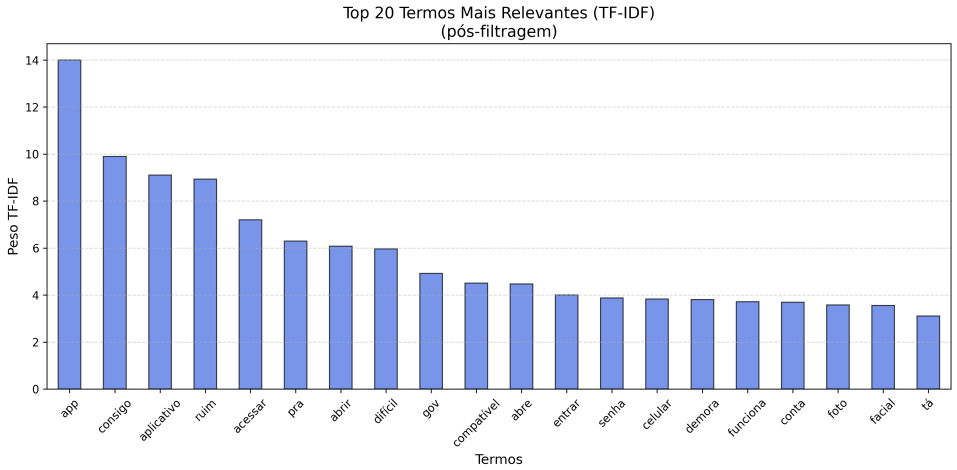


Figura 11 – Principais termos identificados pelo TF-IDF para **summary_threshold** = 0.9.

Sumário	Texto
1	Ridículo ter que sair do app pra fazer login no browser.
2	Muito ruim esse aplicativo
3	Muito pouco intuitivo pessoas mais idosas tem sérias dificuldades com o mesmo
4	Não consigo ter acesso ao aplicativo pois aparece uma mensagem dizendo que meu aparelho não é compatível, gostaria de uma orientação
5	Não estou conseguindo fazer login no Gov.br com a minha conta Nubank. Me ajudem!

Tabela 3 – Sumários gerados para **summary_threshold** = 0.9.

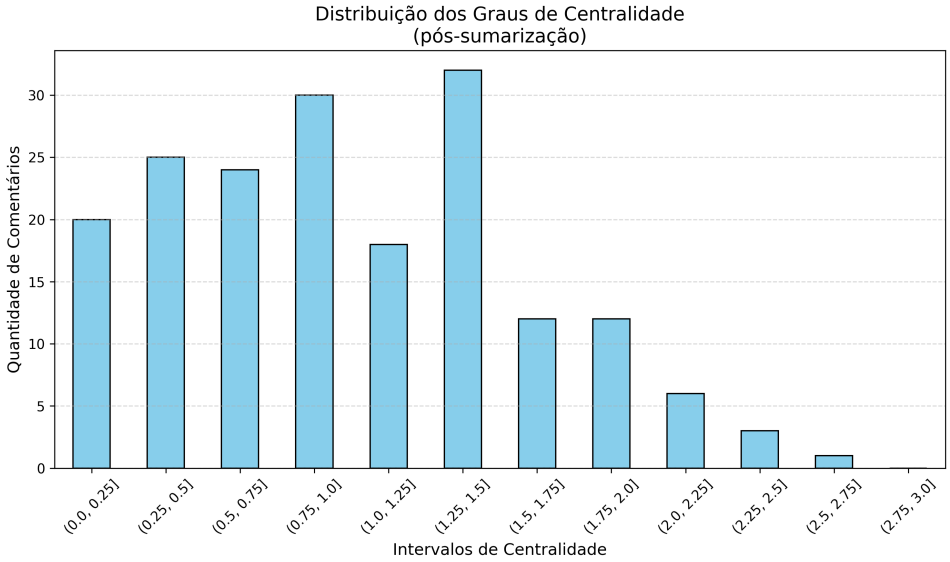


Figura 12 – Distribuição dos scores dos comentários antes e depois da filtragem para **summary_threshold** = 0.9.

4.2 Impacto do flesch_threshold

O parâmetro **flesch_threshold** define o índice mínimo de legibilidade exigido para que um comentário seja considerado. Valores mais baixos permitem a inclusão de textos mais complexos e menos compreensíveis, enquanto valores mais altos filtram conteúdos de difícil leitura. Esse valor varia de 0 a 100.

4.2.1 Limiar baixo (20)

Com **flesch_threshold** = 20, foram mantidos comentários de menor legibilidade, resultando em textos mais difíceis de compreender.

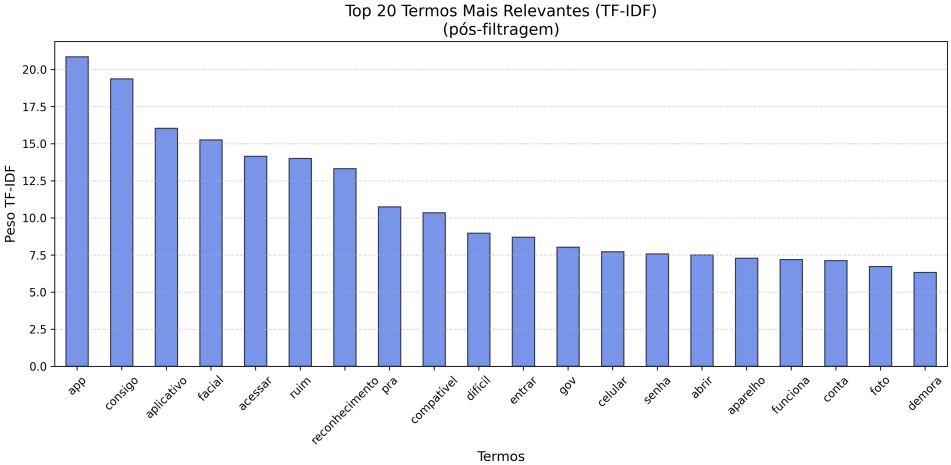


Figura 13 – Principais termos identificados pelo TF-IDF para **flesch_threshold** = 20.

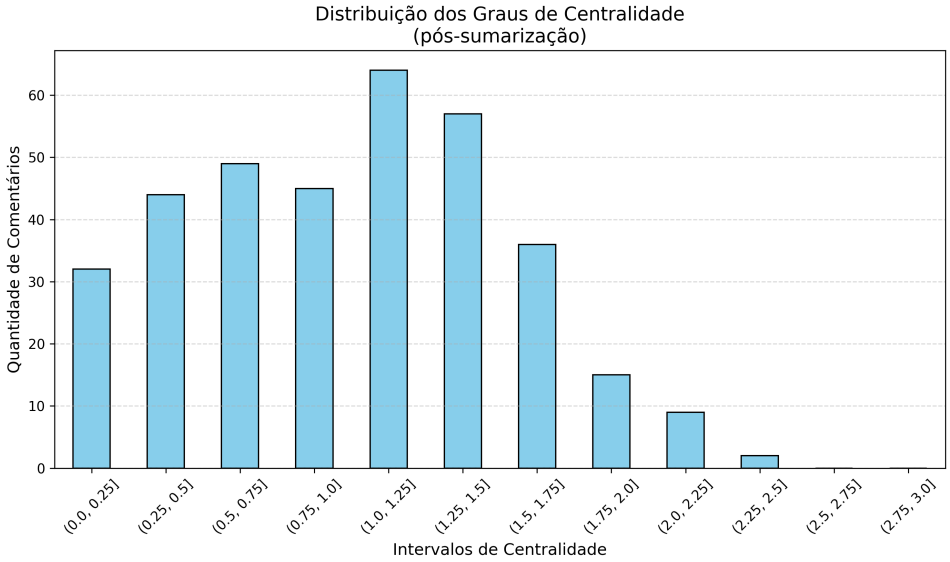


Figura 14 – Distribuição dos scores dos comentários antes e depois da filtragem para **flesch_threshold** = 20.

Sumário	**Texto**
1	não dá para fazer o reconhecimento facial
2	muito ruim o sistema de reconhecimento facial
3	Não consigo acessar meu app do gov.br
4	Muito ruim não consigo fazer o login com reconhecimento facial
5	O reconhecimento facial é muito lento.

Tabela 4 – Sumários gerados para **flesch_threshold** = 20.

4.2.2 Limiar intermediário (50)

Com **flesch_threshold** = 50, houve um equilíbrio entre a inclusão de comentários de legibilidade moderada e a remoção de textos excessivamente complexos.

Sumário	**Texto**
1	Não consigo acessar meu app do gov.br
2	tá muito ruim pra acessar o app.
3	Não consigo abrir o app
4	muito bom o aplicativo
5	É um app palhaçada. Lhe obrigam a instalar o app, talvez uma forma de controlar onde você vai. Agora exigem 2 etapas e o app... E exigem que o modo desenvolver esteja desativado. Lixo de app

Tabela 5 – Sumários gerados para **flesch_threshold** = 20.

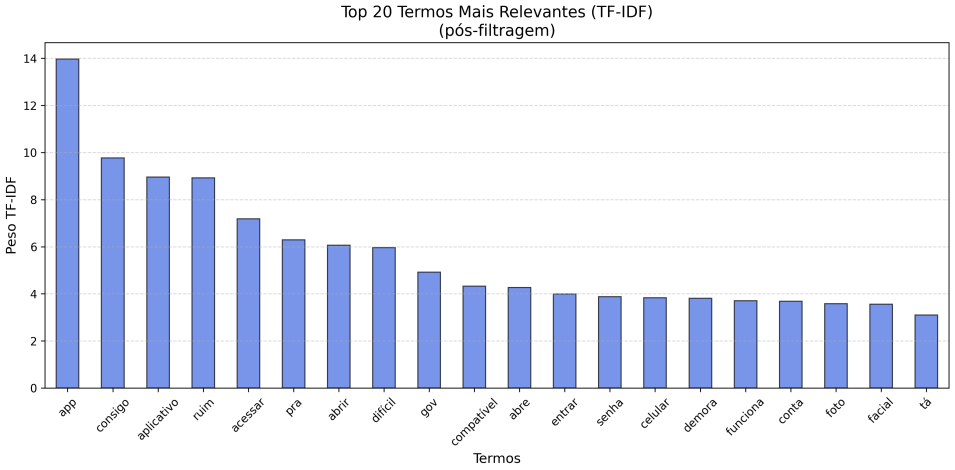


Figura 15 – Principais termos identificados pelo TF-IDF para **flesch_threshold** = 50.

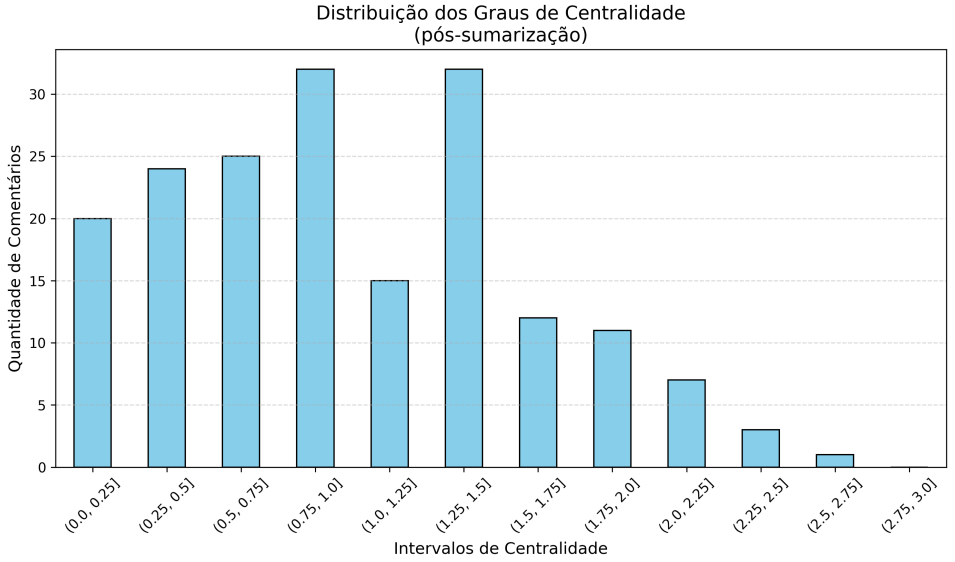


Figura 16 – Distribuição dos scores dos comentários antes e depois da filtragem para **flesch_threshold** = 50.

4.2.3 Limiar alto (80)

Com **flesch_threshold** = 80, apenas os comentários mais simples e fáceis de entender foram mantidos.

4.3 Impacto do min_words

O parâmetro **min_words** define a quantidade mínima de palavras que um comentário deve ter para ser considerado na sumarização. Um valor menor permite a inclusão de comentários mais curtos, enquanto um valor maior filtra mensagens breves e prioriza textos mais desenvolvidos.

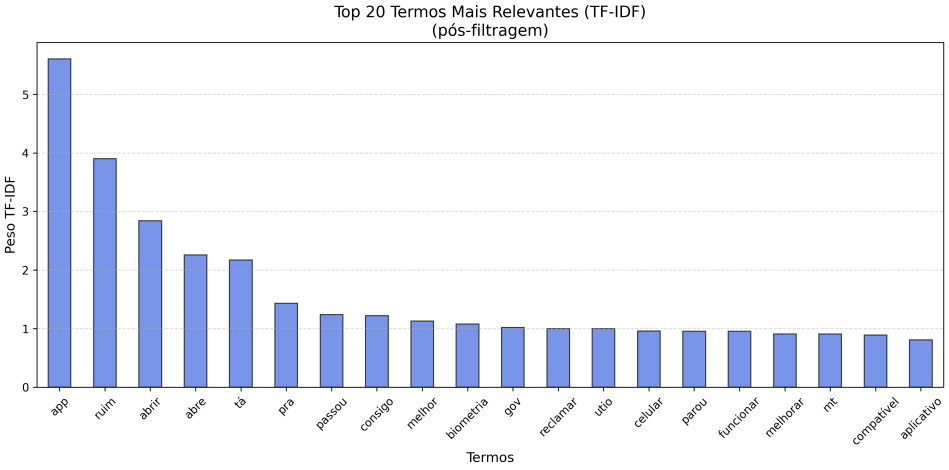


Figura 17 – Principais termos identificados pelo TF-IDF para **flesch_threshold** = 80.

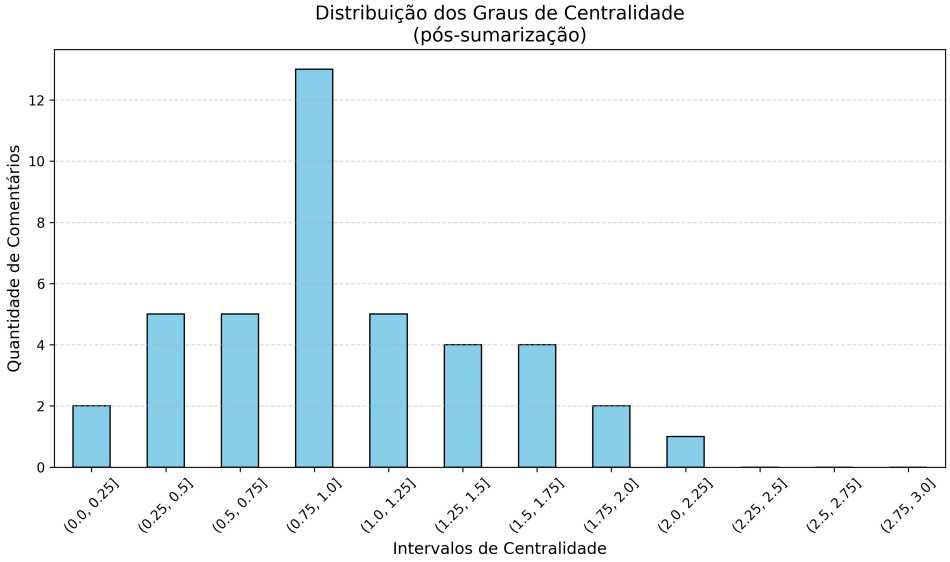


Figura 18 – Distribuição dos scores dos comentários antes e depois da filtragem para **flesch_threshold** = 80.

Sumário	**Texto**
1	tá muito ruim pra acessar o app.
2	Não consigo abrir o app
3	Era muito bom agora ele não abrir não sei porque.
4	Bem ruim o app péssimo
5	muito ruim esse app

Tabela 6 – Sumários gerados para **flesch_threshold** = 80.

4.3.1 Valor baixo (3)

Com **min_words** = 3, comentários extremamente curtos foram mantidos, resultando em sumários mais fragmentados e possivelmente menos informativos.

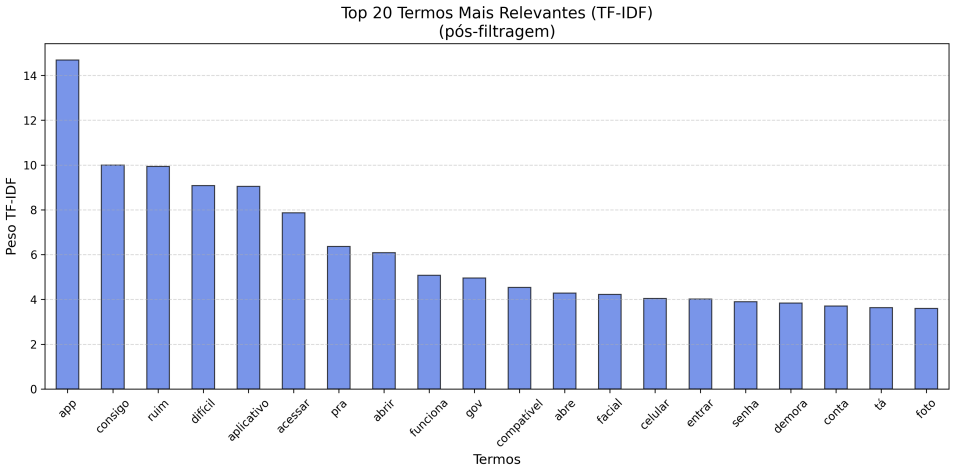


Figura 19 – Principais termos identificados pelo TF-IDF para **min_words** = 3.

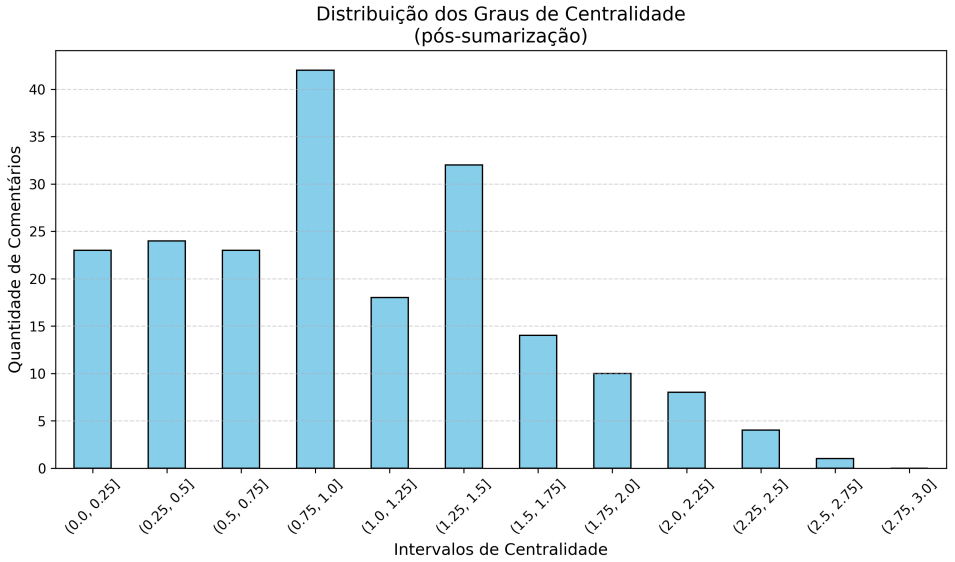


Figura 20 – Distribuição dos scores dos comentários antes e depois da filtragem para **min_words** = 3.

Sumário	Texto
1	Não consigo acessar meu app do gov.br
2	Muito difícil
3	tá muito ruim pra acessar o app.
4	É um app palhaçada. Lhe obrigam a instalar o app, talvez uma forma de controlar onde você vai. Agora exigem 2 etapas e o app... E exigem que o modo desenvolver esteja desativado. Lixo de app
5	Não consigo abrir o app

Tabela 7 – Sumários gerados para **min_words** = 3.

4.3.2 Valor intermediário (8)

Com **min_words** = 8, houve um equilíbrio entre a inclusão de comentários mais curtos e a filtragem de mensagens excessivamente breves.

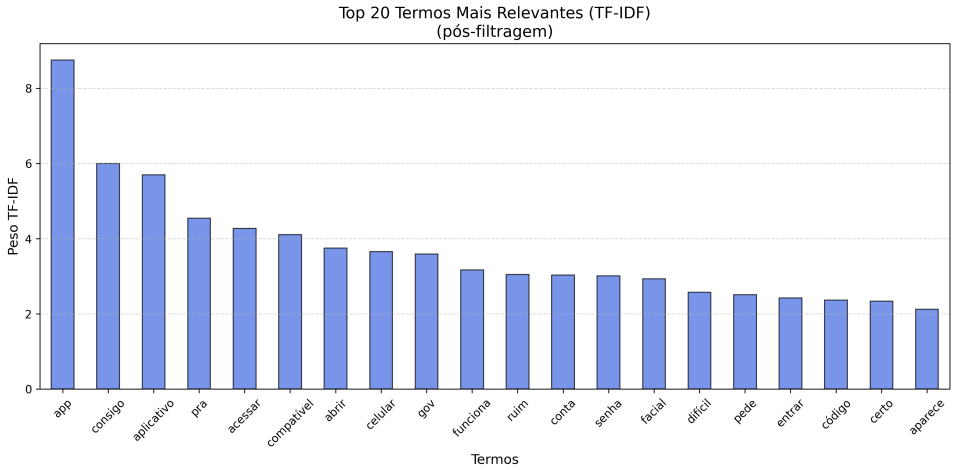


Figura 21 – Principais termos identificados pelo TF-IDF para **min_words** = 8.

Sumário	Texto
1	Simplesmente o app não funciona mais no meu celular, fala que a versão não é mais compatível.
2	Ridículo ter que sair do app pra fazer login no browser.
3	É um app palhaçada. Lhe obrigam a instalar o app, talvez uma forma de controlar onde você vai. Agora exigem 2 etapas e o app... E exigem que o modo desenvolver esteja desativado. Lixo de app
4	Não sei se a culpa é do aplicativo, mas, não consigo entrar na minha conta Gov. Só acessa até digita o CPF, e roda até cai, não chega a pedir senha.
5	Essa desgra... toda vez só trava ...toda vez e nescessário fazer cadastro e a facial nem sempre funciona lixo

Tabela 8 – Sumários gerados para **min_words** = 8.

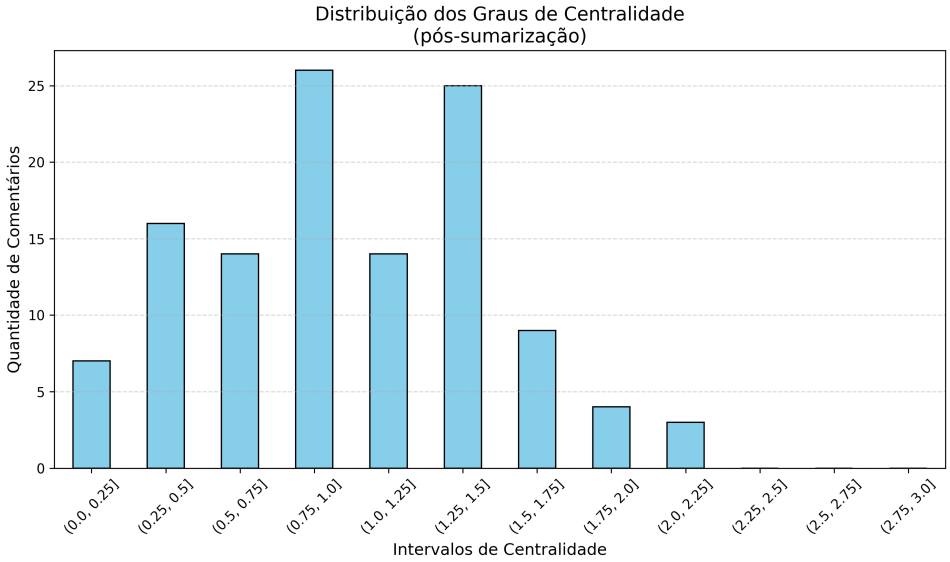


Figura 22 – Distribuição dos scores dos comentários antes e depois da filtragem para **min_words** = 8.

4.3.3 Valor alto (13)

Com **min_words** = 13, apenas comentários mais detalhados foram mantidos, resultando em sumários mais descritivos e menos fragmentados.

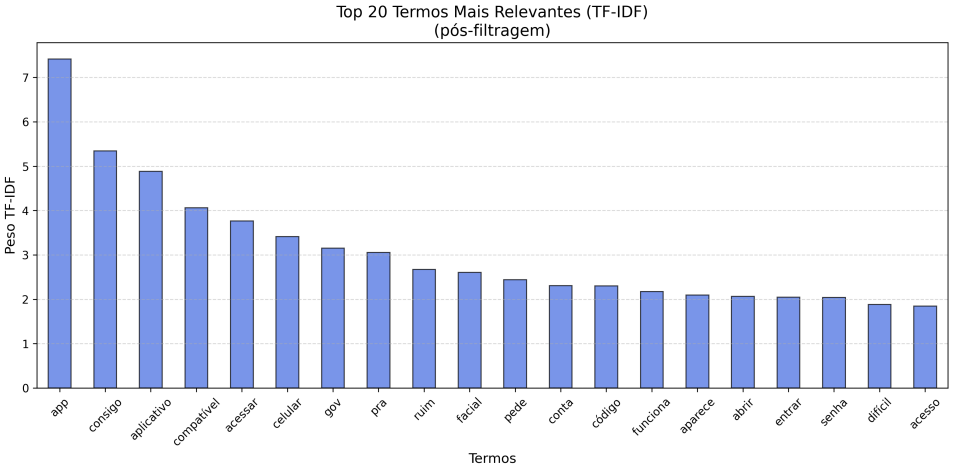


Figura 23 – Principais termos identificados pelo TF-IDF para **min_words** = 13.

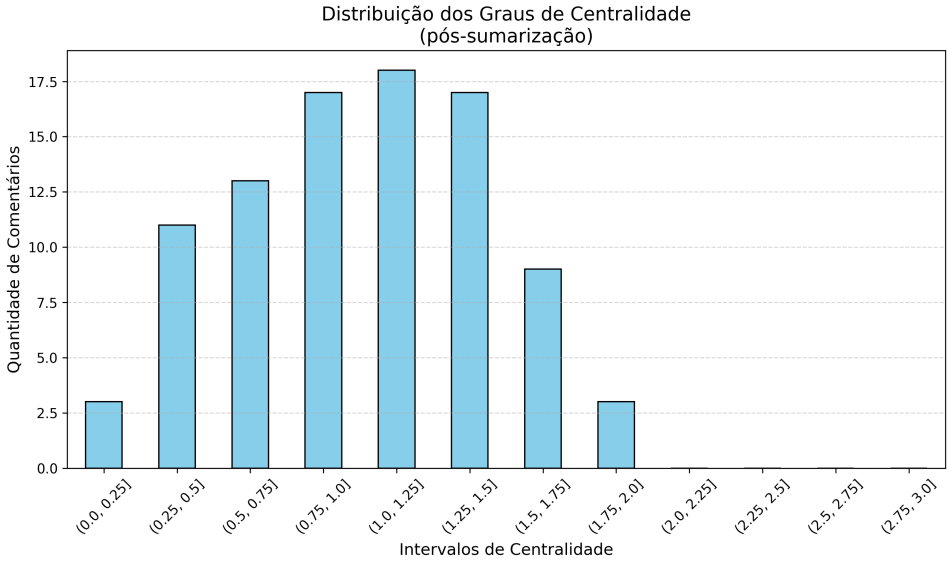


Figura 24 – Distribuição dos scores dos comentários antes e depois da filtragem para **min_words** = 13.

Sumário	**Texto**
1	Simplesmente o app não funciona mais no meu celular, fala que a versão não é mais compatível.
2	Não consigo abrir meu app minha senha desapareceu não sei o que fazer.
3	Cadê o Código de Segurança, não chega no meu número? TI do governo preguiçosos. Não consegue fazer um app que presta.
4	tô tentando anxar os dados do médico do meu filho que é especial na carteira de passe livre e não consigo
5	Não sei se a culpa é do aplicativo, mas, não consigo entrar na minha conta Gov. Só acessa até digita o CPF, e roda até cai, não chega a pedir senha.

Tabela 9 – Sumários gerados para **min_words** = 13.

4.4 Resultados com Parâmetros Otimizados

Com base na análise experimental, os valores dos parâmetros foram ajustados para priorizar a inclusão de textos mais informativos e representativos. Os valores otimizados são:

- **summary_threshold** = 0.05: Permite que mais sentenças sejam incluídas na sumarização, garantindo uma melhor cobertura dos comentários.
- **flesch_threshold** = 45: Filtra comentários excessivamente complexos, mantendo aqueles com uma legibilidade moderada.
- **min_words** = 8: Exclui comentários muito curtos e irrelevantes sem restringir excessivamente o espaço amostral.

4.4.1 Filtragem de Comentários

A filtragem inicial eliminou comentários redundantes, irrelevantes ou de baixa qualidade. Os critérios utilizados foram:

- Índice de legibilidade de Flesch inferior a 45.
- Comentários com menos de 10 palavras.
- Comentários com nota superior a 3 estrelas.

A tabela 10 mostra exemplos de comentários que foram mantidos e de comentários que foram removidos pelo processo de filtragem.

Status	**Comentário**
Removido	Ótimo app.
Removido	Muito ruim, travando toda hora.
Removido	Não gostei
Removido	App excelente.
Mantido	O aplicativo é bom, mas há problemas com a autenti- cação e demora no carregamento das páginas.
Mantido	Após a última atualização, a funcionalidade de login ficou instável, causando erros frequentes.
Mantido	Seria útil se houvesse mais informações sobre os serviços oferecidos dentro do próprio app.

Tabela 10 – Exemplos de comentários removidos e mantidos após a aplicação dos critérios de filtragem.

4.4.2 Análise das Características dos Comentários

A seguir, são apresentados gráficos que mostram a distribuição das características dos comentários processados após a aplicação dos parâmetros otimizados.

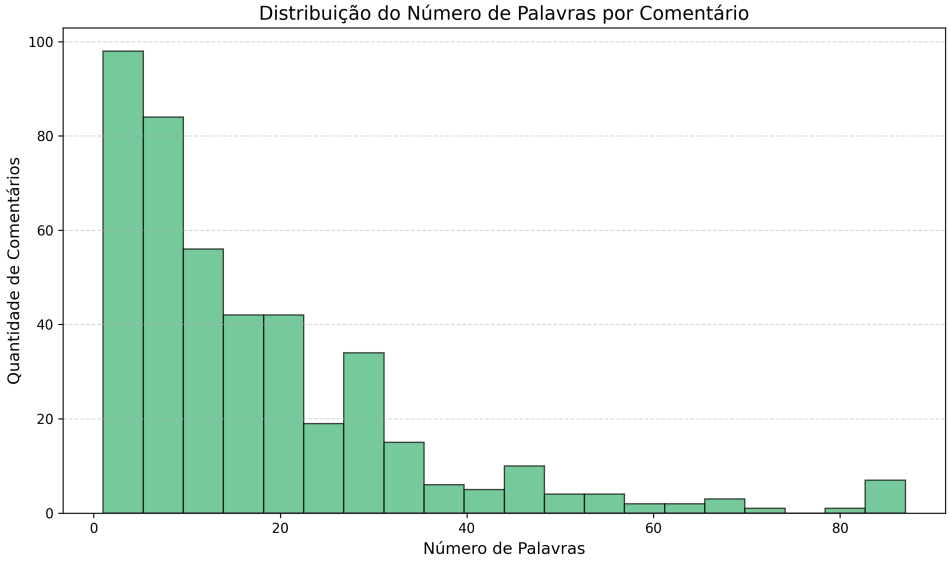


Figura 25 – Distribuição da quantidade de palavras nos comentários.

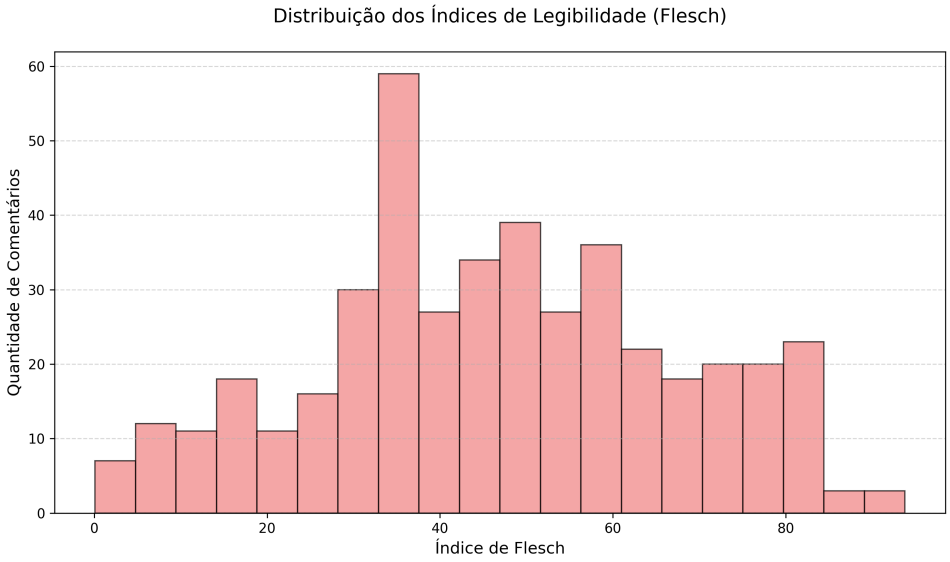


Figura 26 – Distribuição dos índices de Flesch.

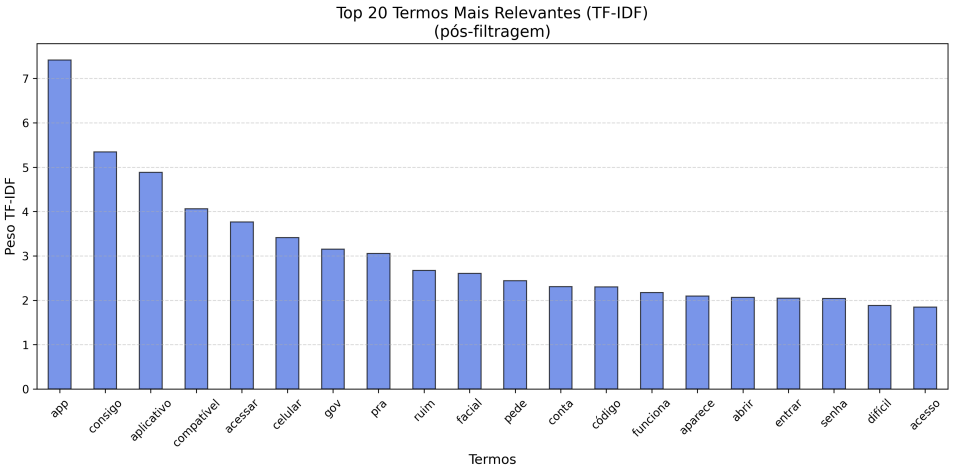


Figura 27 – Principais termos identificados pelo TF-IDF após a filtragem e sumarização.

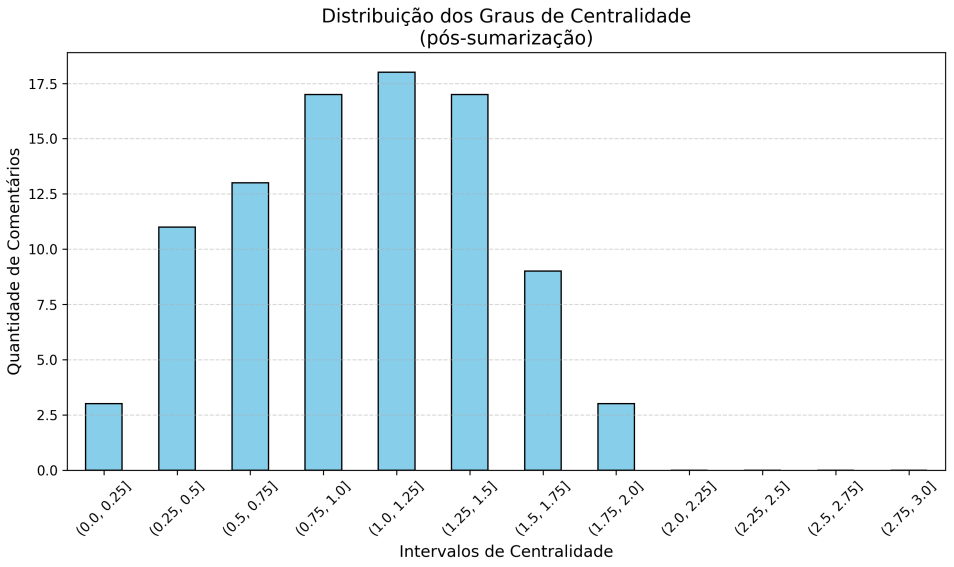


Figura 28 – Distribuição dos scores de centralidade dos comentários após a filtragem.

4.4.3 Sumários Finais Gerados

Foi aplicada a sumarização extrativa com LexRank, que selecionou sentenças mais representativas dos comentários filtrados. O tamanho do sumário foi limitado a 8 sentenças para ter uma melhor amostragem. A Tabela 11 apresenta os sumários finais gerados após a aplicação dos parâmetros otimizados.

Sumário	**Texto**
1	App parou de funcionar no meu smartphone há muito tempo. Não consigo baixar mais, mensagem de incompatibilidade com o dispositivo. Eu tinha validação em 2 etapas e não consigo resolver nada que preciso porque sempre pede pra entrar pelo gov. Mas não tenho acesso. Aff
2	Ridiculo, eu preciso do aplicativo e quando vou instalar descubro que meu celular é incompatível. Sendo que é um celular que foi lançado em 2023/2024. Meu celular é um Xiaomi Redmi Note 12S, não faz sentido o app estar incompatível sendo que as atualizações antigas funcionava de boa (e não não posso baixar atualização antiga pq o app não funciona ele pede para atualizar e não abre)
3	Não consigo baixar em meu celular. não aparece nem Playstore tá uma m já tô bem loco e fala que o app não é compatível com o Cel .e pior que tenho que ter esse gov pra acessar todos os app do governo tá difícil....
4	Tenho o app a um tempo , mas agora não consigo abrir. Pois tem uma mensagem que diz que meu celular não é compatível. Até o ano passado era. Como vou usar?
5	Tá difícil resolver as coisas não vai de jeito nenhum o código de acesso saiu do meu app não consigo resolver a ideia e boa juntar tudo num appas e muito burocrático da certo não..
6	Está falando q o app não é mais compatível com o aparelho. Aparelho é um redimi note 12s, já está todo atualizado e feito todos os procedimentos para poder abrir o app, esse erro veio agora com a virada de ano. Não consigo acesso mais pelo celular...
7	Estão dizendo que o app não é mais compatível com o meu dispositivo. Quer dizer que eu vou ter que comprar um novo celular só pra poder abrir o app?
8	Pelo amor de Deus, como um aplicativo importantíssimo não e compatível com os celulares? E outra cheio de problemas dentro app, o meu não consigo cadastrar a mais de 8 messes.

Tabela 11 – Sumários finais gerados com os parâmetros otimizados.

4.4.4 Problemas Identificados

A partir dos sumários finais apresentados na Tabela 11, os principais problemas relatados pelos usuários do aplicativo Gov.Br foram identificados:

- **Incompatibilidade do aplicativo**: Um grande número de usuários relatou que o aplicativo se tornou incompatível com seus dispositivos, incluindo modelos relativa-

mente recentes. Essa incompatibilidade impede o acesso a serviços essenciais e força os usuários a procurarem alternativas ou realizarem procedimentos presenciais.

- **Problemas na atualização**: Alguns relatos indicam que versões anteriores do aplicativo funcionavam corretamente, mas após as atualizações, o acesso foi bloqueado. Além disso, não há possibilidade de utilizar versões antigas, pois o aplicativo exige a atualização para continuar funcionando.
- **Impossibilidade de instalação**: Muitos usuários mencionaram que o aplicativo simplesmente não aparece na Play Store ou exibe uma mensagem de incompatibilidade no momento da instalação, sem justificativas técnicas aparentes.
- **Dependência excessiva do Gov.br**: O bloqueio de acesso ao aplicativo impacta não apenas sua própria usabilidade, mas também o acesso a outros serviços governamentais que dependem da autenticação via Gov.br.
- **Falta de suporte e orientação**: Alguns usuários relataram dificuldades para obter suporte ou instruções claras sobre como resolver os problemas de compatibilidade e acesso.

Os resultados indicam que a falta de compatibilidade do aplicativo com determinados dispositivos é o principal obstáculo enfrentado pelos usuários. A impossibilidade de instalar ou acessar o Gov.Br compromete serviços digitais essenciais e gera frustração significativa. Melhorias no suporte técnico, comunicação clara sobre requisitos de sistema e flexibilização das restrições de compatibilidade são medidas necessárias para mitigar esses problemas.

5 Conclusão

Este trabalho apresentou um estudo sobre a aplicação de técnicas de sumarização extrativa para a análise de comentários de usuários do aplicativo Gov.Br. A abordagem proposta integrou um processo de filtragem baseado em características linguísticas, seguido pela aplicação do algoritmo LexRank para a geração de resumos automáticos. O objetivo principal foi desenvolver uma ferramenta que facilitasse a análise de grandes volumes de comentários, permitindo a identificação eficiente das informações mais relevantes.

Além da implementação da ferramenta, este estudo também explorou e explicou conceitos fundamentais do campo de Processamento de Linguagem Natural (PLN), oferecendo uma base teórica sólida para a compreensão dos métodos utilizados. No referencial teórico, foram discutidos algoritmos de ranqueamento baseados em grafos, como o PageRank, LexRank e TextRank, suas relações e aplicações na extração de informações relevantes de textos. Além disso, abordagens complementares, como a análise de características linguísticas e técnicas estatísticas, foram apresentadas para contextualizar a importância da filtragem de dados antes da sumarização. Dessa forma, o trabalho não apenas propôs uma solução prática para a análise de comentários, mas também serviu como um material de referência sobre os fundamentos teóricos da sumarização automática.

A implementação da filtragem demonstrou um impacto significativo na qualidade dos resumos gerados. A remoção de comentários irrelevantes, redundantes ou de baixa informatividade garantiu que o LexRank operasse sobre um conjunto de dados mais coeso e representativo. O uso de métricas de legibilidade, como o índice de Flesch, permitiu descartar textos com baixo grau de compreensão. Esses mecanismos contribuíram para uma melhoria na representatividade dos sumários, reduzindo a presença de informações triviais e enfatizando conteúdos mais substanciais.

Os resultados indicaram que a combinação da filtragem com a sumarização baseada em grafos oferece uma solução eficiente para o processamento de grandes quantidades de comentários. O LexRank demonstrou ser uma alternativa viável para a geração de sumários sem a necessidade de modelos complexos baseados em redes neurais, apresentando um equilíbrio entre simplicidade de implementação, baixo custo computacional e interpretabilidade. A análise dos resumos gerados confirmou a capacidade do algoritmo de selecionar as sentenças mais centrais e informativas, proporcionando um panorama conciso das principais reclamações e sugestões dos usuários.

Entretanto, algumas limitações foram identificadas. A abordagem extrativa, apesar de eficiente, não reestrutura o conteúdo dos comentários, podendo gerar sumários que carecem de fluidez textual. Além disso, a definição de parâmetros, como os limiares de

legibilidade e de relevância, ainda demanda ajustes experimentais para melhor adaptação ao domínio específico dos dados analisados. Outra limitação observada é a dependência exclusiva da similaridade lexical entre frases para a construção do grafo do LexRank, o que pode resultar na perda de informações relevantes que não compartilham termos idênticos, mas que possuem proximidade semântica.

Como direções para trabalhos futuros, sugere-se a exploração de técnicas mais avançadas de filtragem, como a incorporação de análise semântica e classificação supervisionada para aprimorar a seleção dos comentários relevantes. Além disso, a avaliação da qualidade dos resumos poderia ser enriquecida com a participação de especialistas humanos, permitindo uma comparação mais aprofundada entre os resultados automáticos e a interpretação manual dos dados. Outra possibilidade de aprimoramento envolve a experimentação com abordagens híbridas, combinando sumarização extrativa e abstrativa para melhorar a coesão textual dos sumários gerados.

Além disso, uma expansão interessante seria a adaptação da ferramenta para diferentes domínios e tipos de textos, como avaliações de produtos em e-commerce, comentários de redes sociais ou relatos de experiências em portais governamentais. A personalização dos critérios de filtragem para diferentes contextos pode tornar o modelo mais versátil e adequado a cenários específicos. Outra possibilidade é a incorporação de técnicas de aprendizado profundo para refinar a etapa de pré-processamento e seleção de sentenças, tornando a filtragem mais precisa e adaptável a padrões linguísticos mais complexos.

Dessa forma, este trabalho contribui para a área de Processamento de Linguagem Natural ao demonstrar a viabilidade da sumarização extrativa aplicada a grandes volumes de comentários, oferecendo uma abordagem prática e eficiente para a análise de feedbacks em plataformas digitais. Além disso, ao apresentar e discutir conceitos essenciais de PLN e sumarização automática, o estudo serve como um referencial técnico para futuras pesquisas na área.

Referências

ALLAHYARI, M. et al. Text summarization techniques: A brief survey. *International Journal of Advanced Computer Science and Applications*, The Science and Information Organization, v. 8, n. 10, 2017. Disponível em: <http://dx.doi.org/10.14569/IJACSA.2017.081052>. Citado 2 vezes nas páginas 18 e 19.

AUTOR(ES). Título do artigo. In: *Anais da Conferência*. [S.l.]: IEEE, 2022. p. 123–456. Citado na página 20.

BANSAL, S. *textstat - Python package for text statistics*. 2024. Disponível em: <https://github.com/shivam5992/textstat>. Citado 2 vezes nas páginas 33 e 39.

BRYANT, R. et al. Big data: Challenges and opportunities. *Journal of Computer Information Systems*, Taylor & Francis, v. 56, n. 4, p. 323–331, 2016. Citado na página 14.

CAJUEIRO, D. O. et al. A comprehensive review of automatic text summarization techniques: method, data, evaluation and coding. *arXiv preprint arXiv:2301.03403*, 2023. Disponível em: <https://arxiv.org/abs/2301.03403>. Citado 2 vezes nas páginas 19 e 20.

CAO, M. A survey on neural abstractive summarization methods and factual consistency of summarization. *arXiv preprint arXiv:2204.09519*, 2022. Disponível em: <https://arxiv.org/abs/2204.09519>. Citado 2 vezes nas páginas 19 e 20.

CHEN, M.; MAO, S.; LIU, Y. Big data: A survey. *Mobile Networks and Applications*, Springer, v. 19, n. 2, p. 171–209, 2014. Citado na página 14.

DALE, E.; CHALL, J. S. A formula for predicting readability. *Educational Research Bulletin*, v. 27, p. 11–20, 1948. Citado na página 34.

ERKAN, G.; RADEV, D. R. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, v. 22, p. 457–479, 2004. Citado 3 vezes nas páginas 21, 34 e 39.

ERKAN, G.; RADEV, D. R. Lexrank: graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, AI Access Foundation, El Segundo, CA, USA, v. 22, n. 1, p. 457–479, dec 2004. ISSN 1076-9757. Citado 3 vezes nas páginas 29, 30 e 31.

FLESCH, R. A new readability yardstick. *Journal of Applied Psychology*, v. 32, n. 3, p. 221, 1948. Citado na página 34.

GOOGLE. *Google Play Store*. 2024. <https://play.google.com/store/apps>. Acessado em: 27 de agosto de 2024. Citado na página 37.

GOV.BR - Aplicativo Oficial. 2024. <https://play.google.com/store/apps/details?id=br.gov.meugovbr>. Acessado em: 12 de agosto de 2024. Citado na página 15.

GOV.BR: Aplicativo de Serviços do Governo Brasileiro. 2020. <https://apps.apple.com/br/app/gov-br/id1521353931>. Acessado em: 27 de agosto de 2024. Citado na página 37.

GRAPHAWARE. *Efficient Unsupervised Topic Extraction with NLP and Neo4j*. 2017. <https://graphaware.com/neo4j/2017/10/03/efficient-unsupervised-topic-extraction-nlp-neo4j.html>. Acessado em: 26 ago. 2024. Citado na página 26.

HUNTER, J. D. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, IEEE COMPUTER SOC, v. 9, n. 3, p. 90–95, 2007. Citado na página 37.

JOACHIMS, T. Text categorization with support vector machines: Learning with many relevant features. In: *Proceedings of the European Conference on Machine Learning (ECML)*. [S.l.]: Springer, 1998. p. 137–142. Citado na página 21.

JOACHIMS, T. Text categorization with support vector machines: Learning with many relevant features. In: *Proceedings of the 10th European Conference on Machine Learning (ECML)*. [S.l.: s.n.], 1998. p. 137–142. Citado na página 35.

JoMingyu. *Google-Play-Scraper: A Python package to scrape Google Play Store data*. 2019. <https://pypi.org/project/google-play-scraper/>. Acessado em: 27 de agosto de 2024. Citado na página 37.

KLEINBERG, J. Authoritative sources in a hyperlinked environment. *Journal of The ACM - JACM*, v. 46, 01 1999. Citado na página 25.

LANGVILLE, A. N.; MEYER, C. D. *Google's PageRank and Beyond: The Science of Search Engine Rankings*. [S.l.]: Princeton University Press, 2006. ISBN 978-0691122021. Citado na página 25.

LIU, Y.; LAPATA, M. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*, 2019. Citado na página 32.

MANNING, C. D.; RAGHAVAN, P.; SCHüTZE, H. *Introduction to Information Retrieval*. [S.l.]: Cambridge University Press, 2008. Citado na página 34.

MANYIKA, J. et al. *Big Data: The Next Frontier for Innovation, Competition, and Productivity*. 2011. Citado na página 14.

MAYER-SCHöNBERGER, V.; CUKIER, K. *Big Data: A Revolution That Will Transf orm How We Live, Work, and Think*. [S.l.]: Eamon Dolan/Mariner Books, 2013. Citado na página 14.

MIHALCEA, R.; TARAU, P. Textrank: Bringing order into text. 07 2004. Citado 3 vezes nas páginas 25, 27 e 28.

MIHALCEA, R.; TARAU, P. Textrank: Bringing order into texts. In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. [S.l.: s.n.], 2004. p. 404–411. Citado na página 21.

NASCIMENTO, D. N. C. R. *Sumarização de Textos em Língua Portuguesa: Abordagens Abstrativa e Extrativa com Modelos de Linguagem Pré-Treinados*. Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio de Janeiro, 2023. Disponível em: <https://www.maxwell.vrac.puc-rio.br/64511/64511.PDF>. Citado 2 vezes nas páginas 19 e 20.

NENKOVA, A.; MCKEOWN, K. Automatic summarization. *Foundations and Trends in Information Retrieval*, Now Publishers Inc., v. 5, n. 2-3, p. 103–233, 2011. Citado na página 15.

NumPy. *NumPy: The fundamental package for scientific computing with Python*. 2006. <https://numpy.org>. Acessado em: 27 de agosto de 2024. Citado na página 37.

OLIVEIRA, A. S.; COSTA, A. H. R. Plsum: Generating pt-br wikipedia by summarizing multiple websites. *arXiv preprint arXiv:2112.01591*, 2021. Disponível em: <https://arxiv.org/abs/2112.01591>. Citado na página 19.

PAGE, L. et al. The pagerank citation ranking: Bringing order to the web. 11 1998. Citado na página 25.

PAGE, L. et al. *The PageRank Citation Ranking: Bringing Order to the Web*. [S.l.], 1999. Citado 3 vezes nas páginas 22, 23 e 24.

Pandas. *Pandas: Python Data Analysis Library*. 2009. <https://pandas.pydata.org>. Acessado em: 27 de agosto de 2024. Citado na página 37.

PEDREGOSA, F. et al. *Scikit-learn: Machine Learning in Python*. 2011. 2825–2830 p. Citado na página 35.

PEDROSA, G. et al. A user-centered approach to analyze public service apps based on reviews. In: *Proceedings of the 25th International Conference on Enterprise Information Systems*. SCITEPRESS - Science and Technology Publications, 2023. p. 453–459. Disponível em: <http://dx.doi.org/10.5220/0011774100003467>. Citado na página 15.

Python. *Python Brasil*. 2024. <https://python.org.br>. Acessado em: 27 ago. 2024. Citado na página 36.

RADEV, D. et al. Centroid-based summarization of multiple documents. *Information Processing & Management*, Elsevier, v. 40, n. 6, p. 919–938, 2004. Citado na página 15.

RAFFEL, C. et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, v. 21, n. 140, p. 1–67, 2020. Citado na página 32.

SOUZA, M. S.; OLIVEIRA, R. G.; PEREIRA, L. F. Sumarização automática de comentários de usuários: Um estudo de caso em sistemas de opinião. *Revista Brasileira de Computação Aplicada*, Associação Brasileira de Computação, v. 12, n. 3, p. 45–60, 2020. Citado na página 14.

THE Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, v. 2, n. 2, p. 159–165, 1958. Citado na página 18.

Wikipedia. *Bag-of-words model – Wikipedia, The Free Encyclopedia*. 2024. Acessado em: 27 ago. 2024. Disponível em: <https://en.wikipedia.org/wiki/Bag-of-words_model>. Citado na página 29.

WIKIPEDIA. *PageRank*. 2024. <https://en.wikipedia.org/wiki/PageRank>. Acessado em: 26 ago. 2024. Citado na página 23.

ZHANG, J. et al. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In: *International Conference on Machine Learning*. [S.l.: s.n.], 2020. p. 11328–11339. Citado na página 32.