UNIVERSIDADE FEDERAL DE SÃO CARLOS
DEPARTAMENTO DE COMPUTAÇÃO
ENGENHARIA DE COMPUTAÇÃO

Bruno Fonseca Mengaldo

Análise comparativa de algoritmos de construção de grafos e técnicas de incorporação de palavras na análise de sentimentos

São Carlos - SP

2024

Bruno Fonseca Mengaldo

Análise comparativa de algoritmos de construção de grafos e técnicas de incorporação de palavras na análise de sentimentos

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia de Computação da Universidade Federal de São Carlos, como requisito parcial para a obtenção do título de Bacharel em Engenharia de Computação.

Orientação Prof. Dr. Alan Demétrius Baria Valejo

São Carlos - SP

2024

À minha família e amigos.

Agradecimentos

Agradeço ao meu pai, João, e à minha mãe, Helenice, por sempre me incentivarem e apoiarem nos diversos aspectos da minha vida, mas sobretudo nos estudos. Vocês foram minha base e sem vocês não estaria onde estou. Também agradeço ao meu irmão, pelo suporte e amizade.

Agradeço a todos os meus amigos que foram fonte de inspiração e felicidade.

Também sou grato ao professor Dr. Alan Demétrius Baria Valejo por ter me apresentado ao tema, me orientado e acompanhado ao longo de todo trabalho.

"A educação é a arma mais poderosa que você pode usar para mudar o mundo"
(Nelson Mandela)

Resumo

A análise de sentimentos se tornou uma ferramenta crucial para compreender a percepção do público em diversas áreas, como marketing, política e mídias sociais. Ela permite extrair percepções valiosas de grandes volumes de texto, como avaliações de consumidores ou opiniões expressas nas redes sociais. Compreender o sentimento por trás das palavras pode guiar estratégias de negócios, campanhas políticas e até mesmo aprimorar a interação com o usuário. Uma abordagem comum na análise de sentimentos envolve a aplicação de técnicas de aprendizado de máquina, que podem variar de métodos simples baseados em regras a modelos complexos de processamento de linguagem natural. Recentemente, com o avanço da inteligência artificial, surgiram métodos mais sofisticados que aproveitam não apenas o conteúdo textual, mas também as relações estruturais dos dados. Nesse contexto, o objetivo deste trabalho é realizar uma análise comparativa de algoritmos de classificação semi-supervisionados em grafos. Esses algoritmos são particularmente úteis quando se dispõe de uma quantidade limitada de dados rotulados, uma situação comum em análises de sentimentos devido ao custo e esforço necessários para a anotação manual de grandes conjuntos de dados. A análise experimental explora a qualidade dos grafos gerados a partir de diferentes algoritmos de construção de grafos em relação a diferentes representações de *word embeddings*.

Palavras-chave: Aprendizado de Máquina; Processamento de Linguagem Natural; Análise de Sentimento; Aprendizado Semi-Supervisionado; Grafos; Representações Textuais.

Abstract

Sentiment analysis has become a crucial tool for understanding public perception in various areas, such as marketing, politics, and social media. It allows the extraction of valuable insights from large volumes of text, like consumer reviews or opinions expressed on social networks. Understanding the sentiment behind the words can guide business strategies, political campaigns, and even enhance user interaction. A common approach in sentiment analysis involves the application of machine learning techniques, which can range from simple rule-based methods to complex natural language processing models. Recently, with the advancement of artificial intelligence, more sophisticated methods have emerged that leverage not just textual content, but also the structural relationships of the data. With that said, the goal of this work is to conduct a comparative analysis of semi-supervised classification algorithms in graphs. These algorithms are particularly useful when there is a limited amount of labeled data available, a common situation in sentiment analysis due to the cost and effort required for manual annotation of large datasets. The experimental analysis explores the quality of graphs generated from different graph construction algorithms in relation to different word embeddings representations.

Keywords: Machine Learning; Natural Language Processing; Sentiment Analysis; Semi-Supervised Learning; Graphs; Textual Representations.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

No cenário atual, marcado pelo surgimento e crescimento exponencial de redes sociais, como Twitter e Reddit, e de gigantes do varejo online, como Mercado Livre e Amazon, a geração e o armazenamento de dados em larga escala se tornaram uma realidade incontornável. Essas plataformas são inundadas diariamente com uma quantidade imensa de texto, sejam eles postagens, comentários ou avaliações de produtos. A necessidade de armazenar, processar e, mais crucialmente, extrair significado e percepções desses vastos repositórios de texto apresenta desafios e oportunidades únicas.

No contexto dos e-commerces, as avaliações dos produtos são um recurso valioso, que influencia decisões de compra e molda a percepção da marca. Da mesma forma, nas redes sociais, um grande número de postagens reflete uma ampla gama de opiniões e sentimentos públicos, oferecendo uma janela para as tendências da sociedade e o pulso da opinião pública.

Dentro deste contexto, a análise textual de sentimentos surge como uma das abordagens mais eficazes para interpretar esses dados. Ela é uma técnica no campo do Processamento de Linguagem Natural (PLN), que permite não apenas classificar se um texto expressa um sentimento positivo, negativo ou neutro, mas também capturar a intensidade e as nuances dessas emoções. A análise de sentimentos, portanto, transformou-se em uma ferramenta chave para empresas e organizações que buscam compreender melhor a voz de seus clientes e usuários.

Uma forma de usar a análise de sentimento para classificar textos automaticamente é usando algoritmos e modelos de Aprendizado de Máquina (AM), um dos subcampos da Inteligência Artificial (IA). Eles são capazes de processar e rotular grandes quantidades de dados rapidamente, devido à identificação de padrões na base de dados.

No contexto do aprendizado de máquina, é possível dividi-lo em 3 tipos: o supervisionado, não-supervisionado e semi-supervisionado. O aprendizado supervisionado utiliza apenas dados rotulados, que são conjuntos de dados que contêm tanto as entradas quanto os rótulos correspondentes, servindo como exemplos diretos para o modelo aprender o mapeamento entre entradas e saídas. O não-supervisionado utiliza apenas dados não rotulados, esses incluem apenas as características de entrada sem rótulos associados. Por fim, o semi-supervisionado faz uso de ambos os tipos de dado.

O aprendizado semi-supervisionado é uma abordagem especialmente útil quando há escassez de dados rotulados, uma situação comum devido ao custo e esforço necessários para a rotulação manual. Para aplicar esse tipo de aprendizado no contexto da análise de sentimento, pode-se usar uma abordagem baseada em grafos, uma técnica que tem

apresentado resultados competitivos se comparada a algoritmos tradicionais da literatura (BERTON, 2016).

Usar essa estratégia exige algumas etapas prévias de PLN e construção de grafos. A primeira delas é o pré-processamento dos dados, em que ruídos como pontuações e caracteres especiais são removidos. Depois o texto é dividido em unidades menores (tokens) que podem ser frases ou palavras. E por último ocorrem os processos de remoção das preposições e outras *stop words* - palavras comuns que não contribuem significativamente para o significado do texto - e de lematização - redução das palavras às suas formas base ou lemas.

Em seguida, são aplicadas técnicas de incorporação de palavras - também chamadas de *word embeddings* - que criam uma representação numérica dos dados textuais. Existem inúmeras técnicas responsáveis por fazer essa manipulação e cada terá resultados diferentes, podendo capturar seus significados e relações semânticas.

Por fim é necessário criar o grafo a partir das representações vetoriais das palavras, que servirá de entrada ao algoritmo semi-supervisionado. Aqui, cada nó pode representar uma palavra e as arestas podem refletir as relações ou semelhanças entre esses nós, entretanto, outras modelagens em grafos podem ser consideradas. Assim como o passo anterior, também existem diferentes abordagens para a construção do grafo, que terão diferentes comportamentos.

1.1 Objetivos

1.1.1 Objetivo Geral

O objetivo desse trabalho é explorar a influência dos algoritmos de construção de grafo e das diferentes técnicas de representação textual na análise de sentimento. Para isso, foram pesquisadas diferentes formas de incorporação de palavras, que depois foram relacionadas com métodos de construção de grafos e usados para treinar e predizer resultados por um modelo de aprendizado semi-supervisionado.

1.1.2 Objetivos específicos

Os objetivos específicos do trabalho são:

- Contribuir com a literatura sobre o desempenho dos algoritmos de construção de grafos;
- Investigar e aplicar diferentes técnicas de representação textual na análise de sentimentos;

- Comparar o desempenho de métodos de construção de grafos na classificação semi-supervisionada;
- Identificar as melhores combinações de representações textuais e estruturas de grafos para análise de sentimentos eficaz.

1.2 Organização do trabalho

Este documento é composto por cinco capítulos. O Capítulo 1 inicia a discussão, introduzindo a pesquisa, seus objetivos fundamentais e a motivação por trás do estudo. Em seguida, o Capítulo 2 mergulha na teoria subjacente necessária para atingir os objetivos propostos, cobrindo tópicos como Aprendizado de Máquina (AM) e dimensionalidade, e detalhando os algoritmos e métricas adotadas para a avaliação. O Capítulo 3 revela a abordagem metodológica empregada nos experimentos. Posteriormente, o Capítulo 4 dedica-se à exploração e análise dos dados obtidos, comparando os vários algoritmos de construção de grafo e métodos de representação textual. Por último, o Capítulo 5 sintetiza as descobertas principais do estudo e esboça direções para futuras pesquisas.

2 Fundamentação Teórica

Este capítulo descreve os conceitos fundamentais que formam a base deste estudo. Estes incluem o aprendizado de máquina, com ênfase em algoritmos semi-supervisionados e propagação de rótulo com base em grafos. Também abordam análise de sentimento e representações atributo-valor.

2.1 Inteligência Artificial

A Inteligência Artificial (IA) é um campo da ciência da computação que se dedica a criar sistemas capazes de realizar tarefas que, até recentemente, exigiam a inteligência humana. Essas tarefas incluem aprendizado, raciocínio, resolução de problemas, percepção e linguagem (RUSSELL; NORVIG, 2016). A IA não se limita a um esforço para replicar a inteligência humana em máquinas, mas também busca entender e modelar inteligências possíveis, seja biológica ou artificial (ERTEL, 2018).

2.2 Aprendizado de Máquina

O Aprendizado de Máquina (AM), conforme definido por (MITCHELL, 1997), é um ramo da Inteligência Artificial que capacita sistemas computacionais a aprimorar seu desempenho em tarefas específicas por meio da experiência. Esta área da computação foca em desenvolver algoritmos que podem aprender a partir de dados e fazer previsões ou tomar decisões baseadas nesses dados, sem serem explicitamente programados para isso. O AM é fundamental no processamento de grandes conjuntos de dados, onde a programação manual de regras seria impraticável ou impossível.

Um aspecto central do AM, destacado por (GUIDO S.; MULLER, 2016), é a capacidade de identificar padrões em dados complexos. Isso inclui tarefas como classificação (categorizando dados em classes predefinidas), regressão (prevendo valores contínuos) e clusterização (agrupando dados semelhantes). Essas tarefas são fundamentais em diversas aplicações, como no reconhecimento de fala, diagnósticos médicos e sistemas de recomendação.

O campo do AM está em constante evolução, com novas técnicas e algoritmos sendo desenvolvidos continuamente. Isso inclui avanços em redes neurais profundas, que têm impulsionado progressos significativos em áreas como visão computacional e processamento de linguagem natural. Estes avanços abrem novas possibilidades e desafios, expandindo as fronteiras do que máquinas podem aprender e realizar.

2.2.1 Aprendizado semi-supervisionado

É possível dividir o AM em 3 categorias: supervisionado, não-supervisionado e semi-supervisionado. O aprendizado supervisionado utiliza apenas dados rotulados, que contêm tanto as entradas quanto os rótulos correspondentes e servem como exemplos diretos para o modelo aprender o mapeamento entre entradas e saídas. O aprendizado não-supervisionado utiliza apenas dados não rotulados, que incluem apenas as características de entrada sem rótulos associados. Por fim, o aprendizado semi-supervisionado faz uso de ambos os tipos de dado.

Essa última categoria de aprendizado tem ganhado destaque por sua eficácia em situações em que os dados rotulados são escassos. Conforme descrito por (FACELI et al., 2011), esta abordagem combina um pequeno conjunto de dados rotulados com um grande volume de dados não rotulados para treinar modelos. O treinamento é o processo pelo qual um modelo de aprendizado de máquina ajusta seus parâmetros para aprender a relação entre as entradas e as saídas desejadas, com base nos dados, visando generalizar bem para novos dados não vistos.

Em cenários reais, a obtenção de grandes quantidades de dados rotulados pode ser cara ou impraticável, tornando a essa forma de aprendizado uma alternativa valiosa. Esta metodologia é particularmente útil em áreas como reconhecimento de fala, processamento de linguagem natural e classificação de imagens, onde a rotulação manual de grandes conjuntos de dados é um processo demorado e caro.

O aprendizado semi-supervisionado utiliza diversas técnicas para aproveitar os dados não rotulados. Uma abordagem comum é a auto-rotulação, onde o modelo, inicialmente treinado com os dados rotulados, é usado para rotular os dados não rotulados. Outra técnica envolve a aprendizagem baseada em grafos, onde os dados são representados em um grafo, e a informação dos rótulos é propagada através das conexões (VEGA-OLIVEROS et al., 2014).

2.3 Grafos

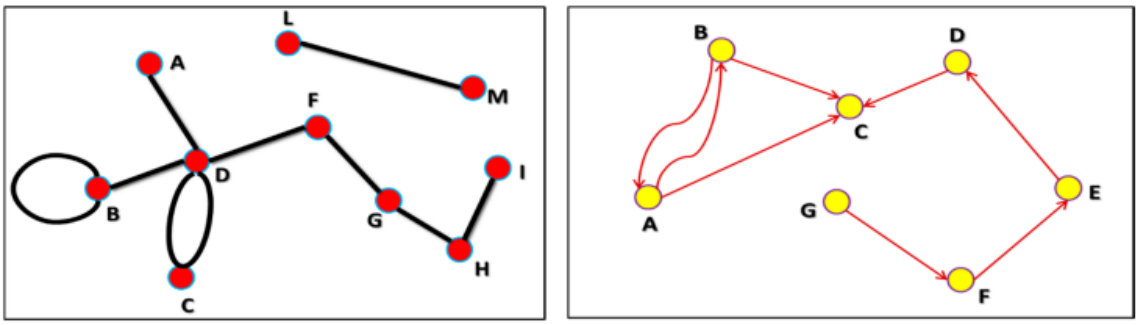
Os grafos, no contexto do Aprendizado de Máquina e da análise de dados, desempenham um papel fundamental na modelagem de relações e padrões complexos. Eles são estruturas compostas de nós e arestas, que representam as conexões entre esses nós. Esta representação é usada para analisar dados relacionais, como redes sociais, sistemas de recomendação e interações biológicas (BERTON, 2016).

Matematicamente, um grafo $G(V, E)$ é composto por um conjunto de n vértices $V = \{v_1, v_2, \ldots, v_n\}$ e um conjunto de m arestas $E = \{e_1, e_2, \ldots, e_m\}$. Assim, a aresta que conecta os vértices v_i e $v_j \in V$ é $e_{i,j}$.

Um grafo pode ser direcionado ou não direcionado, ponderado ou não ponderado, dependendo da natureza dos dados e do problema em questão. Em um grafo direcionado, as arestas têm uma direção, indicando uma relação assimétrica entre os nós, enquanto em um grafo não direcionado, as arestas são bidirecionais. Grafos ponderados atribuem pesos às arestas, que podem representar a força ou a capacidade da conexão entre os nós. A Figura 1 ilustra um grafo não direcionado no exemplo (a) e um direcionado no (b).

Um grafo ponderado $G(V, E, W)$ tem, além de vértices e arestas, uma matriz de peso W, com w sendo a função que atribui um peso $w(e_{i,j})$ para cada aresta $e_{i,j} \in E$.

Figura 1 – Exemplo de grafos direcionados e não direcionados.



(a) Example of an undirected graph. (b) Example of the directed graph.

Fonte: (MAJEED; RAUF, 2020).

A análise de grafos envolve a exploração de propriedades como a conectividade, os caminhos mais curtos, a centralidade dos nós e a detecção de comunidades ou agrupamento. Essas propriedades podem ser exploradas para entender melhor a estrutura e a dinâmica dos dados. Por exemplo, a análise de centralidade pode revelar nós influentes em uma rede social, enquanto a detecção de comunidades pode identificar grupos de usuários com interesses ou comportamentos semelhantes.

2.3.1 Algoritmos de Construção de Grafos

Nem sempre um conjunto de dados é naturalmente relacional, como no caso de dados textuais. Eles podem estar definidos como vetores de características ou até mesmo na forma de atributo-valor. Por isso, é importante adotar algoritmos de construção de grafos para manipular os dados, e conseguir utilizar a abordagem de grafos. A escolha da abordagem tem impacto direto nos resultados, assim também é importante considerar as possibilidades (ZHU, 2005).

2.3.1.1 *k-Nearest-Neighbours*

O *k-Nearest-Neighbours* (k-NN) é um método de construção de grafos que cria conexões com base no conceito de k vizinhos mais próximos (BRITO et al., 1997). Tais

vizinhos são encontrados usando algum cálculo de distância. Existem diversos cálculos, como Manhattan ou Cosseno, e um dos mais utilizados, o Euclidiano.

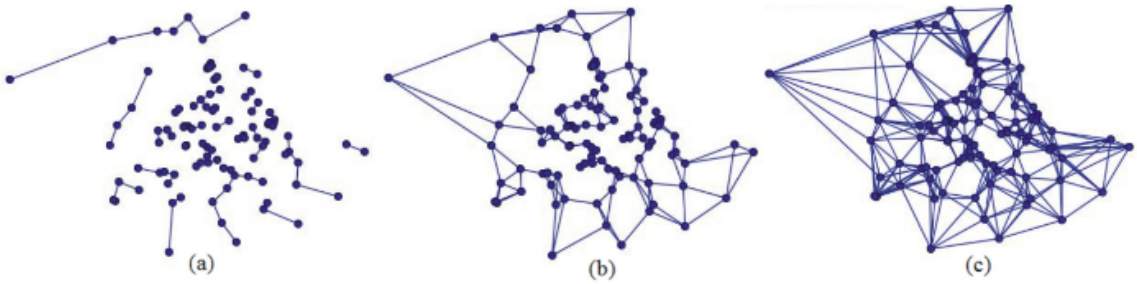
A distância euclidiana entre dois pontos, p e q, cujas coordenadas são, respectivamente, (i, j) e (x, y), é dada por (TORELLI, 2005):

$$\text{dist}(p, q) = \text{dist}((i, j), (x, y)) = \sqrt{(i - x)^2 + (j - y)^2} \tag{2.1}$$

O algoritmo se baseia nas distâncias para construir o grafo, cujos k vizinhos mais próximos de cada nó são conectados por uma aresta. Esse grafo representa a proximidade e similaridade dos nós.

A Figura 2 tem diferentes grafos, gerados com diferentes valores de k. O exemplo (a), que tem o menor k, gerou um grafo desconexo. Em (b) e (c) tem-se maior densidade entre os nós, dado o aumento de k.

Figura 2 – Grafos gerados usando k-NN, em um conjunto de dados com 100 elementos e distribuição gaussiana (a) $k = 1$,(b) $k = 3$ e (c) $k = 7$.



Fonte: (BERTON, 2016).

2.3.1.2 *Mutual k-NN*

Esta variação do k-NN considera uma conexão entre dois pontos apenas se ambos forem vizinhos mútuos, ou seja, cada um está entre os k vizinhos mais próximos do outro. Isso tende a criar um grafo mais robusto, reduzindo as conexões espúrias e melhorando a qualidade da propagação de rótulos (VEGA-OLIVEROS et al., 2014).

Como exige que o grafo siga o princípio dos vizinhos mútuos, o *Mutual k-NN* (Mk-NN) é visto como mais restritivo. Recomenda-se utilizá-lo para identificar subconjuntos específicos ou regiões de alta densidade. A Figura 3 se baseia no Mk-NN, com $k = 11$, para gerar o grafo a partir de um determinado conjunto de dados.

2.3.1.3 *E-Vizinhança*

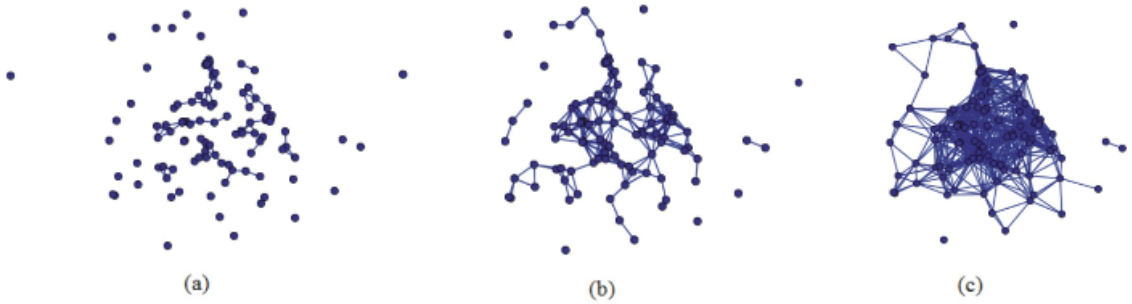
O algoritmo E-Vizinhança, também conhecido como *Epsilon*, cria conexões no grafo com base em um limite previamente definido. Se dois pontos estão dentro de uma distância ε um do outro, eles são conectados. É possível enxergar como uma circunferência de raio ε

Figura 3 – Grafo gerado com Mk-NN e $k = 11$.

Fonte: (BRITO et al., 1997).

centrada em um nó. Assim, todos os outros nós presentes dentro dela serão conectados ao vértice central por meio de arestas. A Figura 4 apresenta exemplos da variação desse limite. Um valor muito baixo, pode fazer com que muitos nós fiquem fora do grafo. Um ε muito alto, por sua vez, pode gerar um grafo muito denso.

Figura 4 – Grafos construídos com E-Vizinhança, em um conjunto de dados com 100 elementos e distribuição gaussiana (a) $E = 0.3$, (b) $E = 0.5$ e (c) $E = 0.9$



Fonte: (BERTON, 2016).

2.3.2 Aprendizado semi-supervisionado em grafos

Seja $X = \{x_1, x_2, \ldots, x_n\}$ os vértices de um grafo com pesos nas arestas $w_{ij} \geq 0$ entre x_i e x_j. Assumimos que o grafo é simétrico, logo $w_{ij} = w_{ji}$. Definimos o grau $d_i = \sum_{j=1}^{n} w_{ij}$. Para um problema de classificação multiclasse com k classes, deixamos que o vetor de base padrão $e_i \in \mathbb{R}^k$ represente a i-ésima classe. Assumimos que os primeiros m vértices $x_1, x_2, \ldots, x_m$ são dados rótulos $y_1, y_2, \ldots, y_m \in \{e_1, e_2, \ldots, e_k\}$, onde $m < n$. A

tarefa da aprendizagem semi-supervisionada baseada em grafos é estender os rótulos para o restante dos vértices $x_{m+1}, x_{m+2}, \ldots, x_n$ (CALDER et al., 2020).

O algoritmo de aprendizado Laplaciano (ZHU, 2005) estende os rótulos resolvendo o problema:

$$\begin{cases} \mathcal{L}u(x_i) = 0, & \text{se } m+1 \leq i \leq n, \\ u(x_i) = y_i, & \text{se } 1 \leq i \leq m, \end{cases} \tag{2.2}$$

onde $\mathcal{L}$ é o Laplaciano do grafo não normalizado dado por:

$$\mathcal{L}u(x_i) = \sum_{j=1}^{n} w_{ij}(u(x_i) - u(x_j)). \tag{2.3}$$

Aqui, $u : X \to \mathbb{R}^k$, com u como $u(x_i) = (u_1(x_i), u_2(x_i), \ldots, u_k(x_i))$. A decisão de rótulo para o vértice x_i é determinada pelo maior componente de $u(x_i)$.

O aprendizado Laplaciano também é chamado de propagação de rótulos (ZHU, 2005), já que a equação de Laplace 2.2 pode ser resolvida substituindo repetidamente $u(x_i)$ pela média ponderada de seus vizinhos, visto como rótulos de propagação dinâmica.

Em taxas de rótulos muito baixas, é proposto substituir o problema 2.2 pelo Aprendizado Poisson (*Poisson learning*): Seja $\bar{y} = \frac{1}{m}\sum_{i=1}^{m} y_i$ o vetor médio de rótulos e deixe $\delta_{ij} = 1$ se $i = j$ e $\delta_{ij} = 0$ se $i \neq j$. Calcula-se a solução da equação de Poisson (CALDER et al., 2020):

$$\mathcal{L}u(x_i) = \sum_{j=1}^{m}(y_j - \bar{y})\delta_{ij} \quad \text{para } i = 1, \ldots, n \tag{2.4}$$

satisfazendo $\sum_{i=1}^{n} d_i u(x_i) = 0$.

Como destacado anteriormente, o aprendizado semi-supervisionado utiliza-se tanto de dados rotulados como não rotulados. No âmbito desse aprendizado em grafos, o aprendizado de Poisson é especialmente concebido para operar sob condições de escassez de rótulos e destaca-se por sua eficiência e robustez (WAN et al., 2021).

Este método de aprendizado tira proveito da estrutura gráfica dos dados para propagar os rótulos conhecidos através do grafo, utilizando uma abordagem para inferir os rótulos desconhecidos. Esse processo de interpolação garante que os nós próximos no grafo, e portanto provavelmente pertencentes à mesma classe ou categoria, recebam rótulos similares.

2.4 Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN) é uma área de estudo interdisciplinar que se situa na interseção da ciência da computação, inteligência artificial e linguística. Seu objetivo é capacitar as máquinas a compreender e interpretar a linguagem humana de maneira que possam realizar tarefas úteis como tradução automática, reconhecimento de fala, e análise de sentimentos. O PLN combina regras linguísticas com algoritmos computacionais para decifrar a complexidade inerente ao idioma, lidando com aspectos como semântica, sintaxe e contexto.

Uma das aplicações do PLN é a análise de sentimentos, que visa identificar e categorizar opiniões expressas em textos. Esta aplicação se torna particularmente desafiadora devido à sutileza e variedade de expressões humanas. Sentimentos e opiniões são frequentemente expressos de maneira implícita, através de sarcasmo, metáforas e linguagem figurativa, exigindo que sistemas de PLN não apenas processem o texto em um nível superficial, mas também compreendam nuances e subtextos.

Para tratar essas complexidades, o PLN faz uso de técnicas de aprendizado de máquina, tanto supervisionadas quanto semi-supervisionadas. Os modelos de aprendizado supervisionado requerem grandes conjuntos de dados anotados manualmente, enquanto as abordagens semi-supervisionadas podem aproveitar conjuntos de dados maiores com anotações esparsas, tornando o processo mais eficiente e escalável.

2.4.1 Análise de Sentimento

A análise de sentimento, um subcampo do PLN, envolve a interpretação e classificação das emoções expressas em textos. Essa técnica é amplamente aplicada para avaliar as opiniões e sentimentos expressos em avaliações de produtos, postagens em mídias sociais, e outras formas de comunicação digital. A capacidade de analisar automaticamente o sentimento de grandes volumes de texto fornece *insights* valiosos para empresas e organizações em diversos setores.

A análise de sentimento geralmente categoriza o texto em sentimentos positivos, negativos ou neutros. Além disso, pode-se avaliar a intensidade desses sentimentos. Esta análise envolve várias etapas, começando pela coleta e pré-processamento de dados textuais, seguido pela aplicação de algoritmos de aprendizado de máquina para classificar os sentimentos. Modelos de aprendizado supervisionado, onde os exemplos de texto são previamente rotulados com sentimentos, são comumente usados. Contudo, com o avanço das técnicas de aprendizado semi-supervisionado e não supervisionado, novos métodos estão sendo explorados para lidar com a complexidade e nuances da linguagem humana.

2.4.2 Representação Atributo-Valor

A representação atributo-valor é uma parte importante do processamento de dados textuais no aprendizado de máquina. Essa abordagem converte texto em um formato numérico que os algoritmos de aprendizado de máquina podem processar, criando um vetor onde cada atributo (ou característica) representa uma dimensão no espaço vetorial.

Assim como nos algoritmo de construção de grafos, as técnicas de representação atributo-valor influenciam no resultado do processamento. Cada uma tem suas vantagens e limitações, e deve ser escolhida com base nas necessidades específicas da tarefa de aprendizado de máquina a ser realizada.

2.4.2.1 *Bag of Words*

A técnica *Bag of Words* (BoW) transforma texto em um vetor de frequência de palavras (MITCHELL, 1997). Neste modelo, cada documento é representado por um vetor onde cada dimensão corresponde a uma palavra no vocabulário do conjunto de dados, e o valor em cada dimensão representa a frequência dessa palavra no documento. Embora o BoW seja simples e intuitivo, ele ignora a ordem das palavras e o contexto, o que pode ser uma limitação para algumas tarefas.

2.4.2.2 *Word2Vec*

O *Word2Vec* é uma técnica mais sofisticada que gera representações vetoriais de palavras, capturando relações semânticas e contextuais. Usando redes neurais, o algoritmo aprende representações vetoriais para cada palavra de forma que palavras com contextos semelhantes tenham representações vetoriais semelhantes. Essa abordagem é útil para capturar nuances, sinônimos, antônimos e outras similaridades de contexto em textos (GUIDO S.; MULLER, 2016).

2.4.2.3 *Term Frequency-Inverse Document Frequency*

O *Term Frequency-Inverse Document Frequency* (TF-IDF) pondera a frequência das palavras em um documento em relação à sua frequência em todo o conjunto de dados (FACELI et al., 2011). Isso ajuda a destacar palavras que são importantes para um documento específico, e não são comuns em todos os documentos. Um documento refere-se a qualquer unidade de texto que pode ser analisada e processada como um único item, como uma publicação em blog, um artigo ou uma postagem em mídia social. Esse algoritmo é particularmente útil para extrair palavras-chave e para tarefas que exigem a distinção entre documentos baseados em seu conteúdo único.

2.5 Medidas de avaliação

Avaliar o desempenho de modelos de aprendizado semi-supervisionado é importante para entender a eficácia de tais sistemas. Essa avaliação é feita por meio de uma série de métricas estatísticas que fornecem percepções sobre a capacidade do modelo de fazer previsões corretas, sua robustez em face de dados não rotulados e a qualidade geral do aprendizado. Cada métrica oferece uma perspectiva diferente, destacando aspectos específicos do desempenho do modelo, como a precisão das previsões, a capacidade de recuperar informações relevantes, e o equilíbrio entre as taxas de acerto e erro.

Dado que os modelos semi-supervisionados são treinados com uma mistura de dados rotulados e não rotulados, é importante que as métricas de avaliação escolhidas sejam capazes de refletir a eficiência na utilização desses dois conjuntos de dados. Os diferentes algoritmos são corretos para diferentes propósitos, então não se pode afirmar que um é o melhor para tudo (HARTIGAN, 1985). As métricas mais comuns incluem acurácia, precisão, recall, F1-Score, além da matriz de confusão.

2.5.1 Matriz de confusão

A Matriz de Confusão é uma ferramenta para a avaliação de desempenho de modelos de classificação, incluindo aqueles utilizados no aprendizado semi-supervisionado. Ela fornece uma representação visual e quantitativa das previsões do modelo em comparação com os valores reais. Ela é útil para entender não apenas o desempenho geral do modelo, mas também os tipos específicos de erros que o modelo está cometendo.

A matriz é tipicamente uma tabela 2x2, representada pela Tabela 1, que classifica as previsões em quatro categorias:

1. Verdadeiros Positivos (VP): Casos em que o modelo previu corretamente a classe positiva.
2. Falsos Positivos (FP): Casos em que o modelo previu incorretamente a classe negativa como positiva.
3. Verdadeiros Negativos (VN): Casos em que o modelo previu corretamente a classe negativa.
4. Falsos Negativos (FN): Casos em que o modelo previu incorretamente a classe positiva como negativa.

É importante destacar que, no contexto das métricas de avaliação, os termos "positivo" e "negativo" não se referem diretamente às classes dos dados, como 'positivo', 'neutro' ou 'negativo' em uma tarefa de análise de sentimentos. Ao invés disso, eles se

referem ao aspecto binário do desempenho de classificação: “positivo” indica um acerto (verdadeiro positivo ou verdadeiro negativo), enquanto “negativo” indica um erro (falso positivo ou falso negativo) na classificação. Independentemente da classe a que o dado pertence, esses termos são empregados para descrever a precisão da classificação. As definições dessas quatro categorias serão utilizadas para o cálculo das outras métricas de avaliação.

Tabela 1 – Matriz de Confusão

		Classificação real	
		P	N
Classificação prevista	P	VP	FP
	N	FN	VN

2.5.2 Acurácia

A acurácia é uma das métricas mais diretas e frequentemente citadas quando se trata de avaliar modelos de classificação. Ela reflete a proporção de previsões corretas (positivas e negativas) em relação ao total de casos examinados. Matematicamente, é expressa pela fórmula:

$$\text{Acurácia} = \frac{VP + VN}{VP + FP + VN + FN} \tag{2.5}$$

Ela pode ser uma medida útil quando as classes estão balanceadas e o custo de um falso positivo é equivalente ao de um falso negativo. Entretanto, é importante ser cauteloso ao usar a acurácia em bases de dados desbalanceados, onde a prevalência de uma classe pode inflar a métrica, dando a ilusão de um modelo altamente preciso quando, na realidade, ele pode estar apenas prevendo a classe majoritária.

A classe majoritária em um conjunto de dados é a categoria que possui o maior número de instâncias ou exemplos. Np contexto de classificação, em que os dados são divididos em diferentes classes ou rótulos, a classe majoritária é aquela que é mais frequentemente observada ou representada nos dados.

2.5.3 Precisão

A precisão é uma métrica que se concentra especificamente na proporção de identificações positivas corretas. É particularmente importante em situações onde os falsos positivos têm implicações significativas. Ela é calculada pela fórmula:

$$\text{Precisão} = \frac{VP}{VP + FP} \tag{2.6}$$

A precisão visa responder à pergunta: “Dentre todas as instâncias que o modelo previu como positivas, quantas realmente eram positivas?”.

No contexto do aprendizado semi-supervisionado, a precisão é serve para entender como o modelo performa na identificação correta de casos positivos, especialmente quando uma quantidade limitada de dados rotulados é utilizada para treinamento. Ela é útil em cenários como diagnósticos médicos ou detecção de fraudes, em que falso positivos podem ser custosos ou perigosos. Por outro lado, ela não leva em conta os verdadeiros negativos e, portanto, deve ser usada em conjunto com outras métricas, como o recall, para obter uma visão mais completa do desempenho do modelo.

2.5.4 Recall

O recall, também conhecido como sensibilidade, mede a capacidade do modelo de identificar corretamente todas as instâncias relevantes, ou seja, a proporção de positivos reais que foram corretamente identificados pelo modelo. Ele é essencial em situações em que é crítico capturar todas as instâncias positivas. A fórmula para calculá-lo é:

$$\text{Recall} = \frac{VP}{VP + FN} \tag{2.7}$$

No contexto do aprendizado semi-supervisionado, em que dados rotulados podem ser limitados, o recall é importante para entender como o modelo se comporta na identificação de casos positivos dentro de um conjunto maior, potencialmente não rotulado. Um alto recall indica que o modelo é eficiente em identificar a maioria dos casos positivos, mas é importante balancear o recall com a precisão, por exemplo, para garantir que o modelo não esteja simplesmente prevendo a maioria dos casos como positivos, o que inflaria artificialmente a performance nessa métrica.

2.5.5 F1-Score

O F1-Score harmoniza a precisão e o recall em uma única medida. É particularmente útil em situações em que é importante manter um equilíbrio entre a captura de instâncias positivas e a precisão na classificação dessas instâncias. Ele proporciona uma visão compreensiva do desempenho do modelo quando ambos os aspectos são igualmente importantes. A fórmula para calculá-lo é:

$$\text{F1-Score} = 2 \times \frac{\text{Precisão} \times \text{Recall}}{\text{Precisão} + \text{Recall}} \tag{2.8}$$

Nesta fórmula, a precisão e o recall são multiplicados e o resultado é dobrado, sendo então dividido pela soma da precisão e do recall. O F1-Score alcança seu melhor valor em 1 - precisão e recall perfeitos - e o pior em 0. Idealmente, um modelo de aprendizado

semi-supervisionado deve maximizar tanto a precisão quanto o recall, mas na prática, muitas vezes há uma troca (*trade-off*) entre estas duas métricas. O F1-Score oferece um meio de balancear essa troca, sendo particularmente útil em conjuntos de dados com distribuições de classe desequilibradas, onde a acurácia por si só pode não ser uma métrica suficientemente informativa.

3 Metodologia

Este capítulo detalha sobre a implementação dos algoritmos e técnicas empregados para realizar os experimentos e obter os resultados, além do conjunto de dados. A linguagem de programação escolhida para a prática foi Python[1], com o auxílio das seguintes bibliotecas: spaCy[2], gensim[3], scikit-learn[4], igraph[5] e graphlearning[6].

3.1 Conjuntos de dados

Foi utilizado um conjunto de dados retirados do trabalho de Stack Tecnologias[7]. Trata-se de uma coleção de tweets, coletados com o objetivo de representar uma variedade de opiniões e sentimentos expressos pelos usuários em relação a temas políticos no estado de Minas Gerais. Cada tweet foi categorizado em um dos três rótulos: positivo, negativo ou neutro. Essa classificação foi baseada no conteúdo textual e no contexto emocional expresso em cada mensagem.

Embora os detalhes específicos sobre o grupo de anotadores responsáveis pela classificação não estejam disponíveis, observa-se que os rótulos foram aplicados seguindo os padrões abaixo:

- Positivo: Tweets classificados como positivos geralmente contêm palavras ou expressões que indicam aprovação ou suporte a uma pessoa, política ou evento. Por exemplo, um tweet elogiando uma decisão política ou expressando otimismo em relação a um candidato seria rotulado como positivo.

- Negativo: Os tweets negativos são aqueles que expressam desaprovação, crítica ou descontentamento. Isso pode incluir comentários sobre desapontamento com resultados eleitorais, discordância com políticas adotadas ou insatisfação com a atuação de figuras políticas.

- Neutro: Esta categoria é atribuída a tweets que não expressam sentimentos claros ou são meramente informativos. Por exemplo, um tweet que relata um evento político sem adicionar uma opinião pessoal seria considerado neutro.

[1] <https://www.python.org/>
[2] <https://spacy.io/>
[3] <https://radimrehurek.com/gensim/index.html>
[4] <https://scikit-learn.org/stable/>
[5] <https://python.igraph.org/en/stable/>
[6] <https://github.com/jwcalder/GraphLearning>
[7] <https://github.com/stacktecnologias/stack-repo/blob/master/Tweets_Mg.csv>

A base tem um total de 8199 tweets. Destes, 3300 foram classificados como positivo; 2453 como neutro; e 2446 como negativos.

Com os dados rotulados, a intenção é utilizar uma parte - cerca de 80% - para realizar os treinos. Com isso, é possível utilizar os 20% restantes para a etapa de testes e obtenção de métricas de avaliação.

3.2 Algoritmos Utilizados

Os algoritmos escolhidos para realizar os experimentos foram especificados no Capítulo 2, como parte da Fundamentação Teórica. A seguir estão explicitados alguns dos parâmetros fixados para a implementação dos algoritmos:

***Lematização*:** Implementação do spaCy - class spacy.load()

- *default*: pt_core_news_sm

***Bag of Words*:** Implementação do scikit-learn - CountVectorizer

- *analyzer*: word

***TD-IDF*:** Implementação do scikit-learn - TfidfVectorizer

- *analyzer*: word

***Word2Vec*:** Implementação do gensim - Word2Vec

- *vector_size*: 100
- *window*: 5
- *min_count*: 1
- *workers*: 4

k-NN Implementação do scikit-learn - kneighbors_graph

- *mode*: distance
- *metric*: euclidean
- *include_self*: falso

Mk-NN Implementação do scikit-learn - kneighbors_graph

- *mode*: connectivity

- *metric*: euclidean
- *include_self*: falso

E-Vizinhança Implementação do scikit-learn - radius_kneighbors_graph

- *mode*: distance
- *metric*: euclidean
- *include_self*: falso

Algoritmo Poisson Implementação do graphlearning - class gl.ssl.poisson()

- *labels*: 3

3.3 Experimentos

A realização dos testes computacionais se deu através do uso efetivo do Python, aproveitando as funcionalidades das bibliotecas destacadas. A estruturação do código foi planejada em diferentes níveis de interação, assegurando uma execução lógica e sistemática das tarefas.

A realização dos experimentos é dividida em quatro etapas. Cada etapa é projetada para preparar uma nova abordagem que será usada na análise comparativa. O objetivo é identificar o efeito que as técnicas de incorporação de palavras e os algoritmos de construção de grafos tem no desempenho final gerado pelo algoritmo de Poisson.

A fase inicial consiste no pré-processamento dos dados, que prepara o conjunto de dados brutos para análise posterior. Nela, usou-se a biblioteca spaCy para realizar as operações listadas. As principais ações nesta etapa incluem:

- Limpeza: Remoção de caracteres não desejados, links e marcações de usuários;
- Tokenização: Segmentação de texto em unidades fundamentais para análise, como palavras ou termos;
- Remoção de *Stop Words*: Eliminação de palavras comuns que não contribuem significativamente para o sentido do texto, como artigos, pronomes e preposições;
- Lematização: Redução das palavras para suas formas raízes ou lemas, uniformizando variações da mesma palavra.

Usando como exemplo um tweet da base de dados, são aplicadas as etapas de pré-processamento nele, com objetivo de entender e exemplificar o que acontece em cada uma delas.

> @diarioaco: Governo de Minas entrega 401 veículos para o transporte de alunos da rede pública de ensino - https://t.co/vLGPRqteud https:...

- Limpeza: Governo de Minas entrega 401 veículos para o transporte de alunos da rede pública de ensino -;
- Tokenização: Governo, de, Minas, entrega, 401, veículos, para, o, transporte, de, alunos, da, rede, pública, de, ensino, -;
- Remoção de *Stop Words*: Governo, Minas, entrega, 401, veículos, transporte, alunos, rede, pública, ensino;
- Lematização: 'Governo', 'Minas', 'entregar', '401', 'veículo', 'transporte', 'aluno', 'rede', 'público', 'ensino'.

Após a limpeza e estruturação dos dados, a seguinte etapa é transformá-los em um formato que possa ser interpretado por algoritmos de construção de grafo. Aqui fez-se uso das bibliotecas citadas no capitulo 3 para definir os métodos Bag of Words, TF-IDF e Word2Vec. Assim, foram criadas 3 representações, uma para cada método.

Depois de realizar a transformação textual para um formato numérico, segue-se para etapa de construção de grafos. Aqui também foram usadas as implementações do capítulo 3 para k-NN, Mk-NN e E-Vizinhança. Os 3 algoritmos foram aplicados para cada um dos métodos de representação anteriores.

Com os 9 grafos criados, é utilizada a biblioteca de graphlearning para aplicar o Algoritmo Poisson em cada grafo. Nessa etapa, cada modelo é treinado com o conjunto de treino e as predições são feitas para o conjunto de teste. Com os resultados do aprendizado, calcula-se as métricas de Acurácia, Precisão, Recall e F1-Score.

4 Análise e discussão dos resultados

O objetivo desse Capítulo é mostrar os resultados experimentais provenientes da execução dos experimentos apresentados no Capítulo 3. Como mencionado no Capítulo 2, no contexto das métricas de avaliação, os termos “positivo” e “negativo” não se referem diretamente às classes dos dados, como na tarefa de análise de sentimentos, e sim ao aspecto binário do desempenho de classificação: “positivo” indica um acerto e “negativo” indica um erro.

A Tabela 2 apresenta a acurácia para cada grafo. A análise dela nos diferentes modelos revela um panorama variado em termos de eficácia geral na classificação. O modelo com Word2Vec + k-NN apresentou a acurácia mais alta de 69.70%, indicando uma capacidade geral de fazer previsões corretas em comparação com as outras combinações de representações e algoritmos de agrupamento. Por outro lado, a configuração com TF-IDF + E-Vizinhança registrou a menor acurácia, apenas 7.26%, sugerindo uma incompatibilidade significativa entre essa representação de atributo-valor e o método de construção do grafo para este conjunto de dados.

Tabela 2 – Acurácia obtida para cada variação de grafo.

Combinação dos algoritmos	**Acurácia**
Bag of Words + k-NN	0.5195121951219512
Bag of Words + Mk-NN	0.5213414634146342
Bag of Words + E-Vizinhança	0.5542682926829269
TF-IDF + k-NN	0.4926829268292683
TF-IDF + Mk-NN	0.5054878048780488
TF-IDF + E-Vizinhança	0.07256097560975609
Word2Vec + k-NN	**0.6969512195121951**
Word2Vec + Mk-NN	0.6158536585365854
Word2Vec + E-Vizinhança	0.6786585365853659

A precisão dos modelos, apresentada na Tabela 3, indica a proporção de previsões positivas corretas em relação a todas as previsões positivas feitas pelo modelo. A análise revela que o modelo Word2Vec + E-Vizinhança alcançou a maior precisão média, de 69.97%, indicando uma alta eficiência em identificar corretamente casos positivos quando comparado com as outras configurações. Por outro lado, o modelo TD-IDF + E-Vizinhaça registrou a menor precisão, de 4.22%, refletindo uma baixa taxa de acertos em suas previsões positivas.

O recall é uma métrica fundamental que avalia a capacidade do modelo de identificar corretamente todos os casos positivos. Essa métrica é importante, especialmente em situações quando é importante capturar todos os exemplos relevantes de uma classe

Tabela 3 – Precisão obtida para cada variação de grafo.

Combinação dos algoritmos	**Precisão**
Bag of Words + k-NN	0.557179051282563
Bag of Words + Mk-NN	0.6288553009044097
Bag of Words + E-Vizinhança	0.6580062649714391
TF-IDF + k-NN	0.5164530905973638
TF-IDF + Mk-NN	0.616598367436537
TF-IDF + E-Vizinhança	0.04218406234851027
Word2Vec + k-NN	0.6915814928449278
Word2Vec + Mk-NN	0.612214989801474
Word2Vec + E-Vizinhança	**0.6997559721873805**

específica.

Os resultados de recall, presentes na Tabela 4, mostram variações notáveis entre os diferentes modelos. O Word2Vec + E-Vizinhança apresentou o recall mais alto com 69.32%, indicando uma forte capacidade de capturar a maioria dos casos positivos. Em contraste, o TF-IDF + E-Vizinhança registrou o recall mais baixo, apenas 8.32%, sugerindo que muitos casos positivos foram perdidos por este modelo.

Tabela 4 – Recall obtido para cada variação de grafo.

Combinação dos algoritmos	**Recall**
BoW + k-NN	0.5390268137042806
BoW + Mk-NN	0.5073362841886965
BoW + E-Vizinhança	0.53611143539775
TF-IDF + k-NN	0.5166422169003548
TF-IDF + Mk-NN	0.48869678524706434
TF-IDF + E-Vizinhança	0.08322056456777542
Word2Vec + k-NN	0.6817923089026999
Word2Vec + Mk-NN	0.5974694913109898
Word2Vec + E-Vizinhança	**0.6931551737221148**

O F1-Score é uma métrica que combina precisão e recall, oferecendo um equilíbrio entre as duas. É particularmente útil em situações em que é importante manter um balanço entre a identificação correta dos casos positivos (precisão) e a captura de todos os casos positivos relevantes (recall).

Os resultados do F1-Score, mostrados na Tabela 5, para os diferentes modelos são indicativos de como cada modelo equilibra a precisão e o recall. O modelo Word2Vec + E-Vizinhança exibiu o F1-Score mais elevado com 67.86%, indicando um excelente equilíbrio entre precisão e recall. Esse resultado sugere que o modelo foi eficaz tanto em identificar corretamente os casos positivos quanto em capturar a maioria dos casos positivos. Por outro lado, o modelo TF-IDF + E-Vizinhança registrou o F1-Score mais baixo, apenas 4.75%, refletindo um desempenho insatisfatório em ambos os aspectos de

precisão e recall.

Tabela 5 – F1-Score obtido para cada variação de grafo.

Combinação dos algoritmos	**F1-Score**
BoW + k-NN	0.521824892861927
BoW + Mk-NN	0.5205584491828462
BoW + E-Vizinhança	0.5488842641763193
TF-IDF + k-NN	0.4903074131899126
TF-IDF + Mk-NN	0.49758638464565097
TF-IDF + E-Vizinhança	0.047522005757632
Word2Vec + k-NN	0.6783962578087044
Word2Vec + Mk-NN	0.5987476376385437
Word2Vec + E-Vizinhança	**0.6785948163549614**

Os resultados mostram que a escolha da representação vetorial e do algoritmo de construção de grafo tem um impacto significativo no desempenho do modelo, afetando as métricas de acurácia, precisão, recall e F1-score.

Notavelmente, o modelo que utilizou Word2Vec em conjunto com o E-Vizinhança demonstrou ser o mais eficiente, alcançando os melhores resultados em termos de precisão, recall e F1-score. Isso sugere que a riqueza semântica capturada pelo Word2Vec, aliada à eficácia do E-Vizinhança na formação de grafos, é particularmente adequada para a tarefa de análise de sentimentos no contexto estudado. Por outro lado, a combinação TF-IDF com E-Vizinhança mostrou-se menos eficaz, indicando que esta abordagem pode não ser a mais apropriada para o conjunto de dados em análise.

Analisando com mais detalhes o modelo que teve melhor desempenho, a Tabela 6 mostra a distribuição das métricas por classe. Observa-se que o modelo Word2Vec + E-Vizinhança tem um desempenho relativamente balanceado em todas as classes, com destaque para a classe Positivo que apresenta uma precisão de 89%, mas um recall mais baixo de 57%, sugerindo que, enquanto o modelo é bastante preciso em identificar tweets positivos corretamente, ele deixa de capturar um número significativo de tweets positivos reais. A classe Negativo tem um recall de 86%, indicando uma boa capacidade de capturar a maioria dos tweets negativos, enquanto a precisão é moderada com 65%. A classe Neutro tem desempenho o mais equilibrado entre precisão e recall, indicando um nível consistente de previsão para essa categoria.

Tabela 6 – Métricas do Word2Vec + E-Vizinhança por classe.

Classe	**Precisão**	**Recall**	**F1-Score**
Negativo	0.65	0.86	0.74
Neutro	0.56	0.65	0.60
Positivo	0.89	0.57	0.70

A Tabela 7 mostra os resultados por classe para o modelo com menor desempenho

geral. O modelo TF-IDF + E-Vizinhança mostra métricas significativamente mais baixas, com valores de precisão e recall que indicam um desempenho insatisfatório. Notavelmente, a classe Positivo tem precisão e recall de 0%, o que significa que o modelo não foi capaz de identificar corretamente nenhum tweet positivo. As classes Negativo e Neutro também têm métricas muito baixas, mas ainda mostram alguma capacidade do modelo de prever corretamente, ainda que limitada.

Tabela 7 – Métricas do TF-IDF + E-Vizinhança por classe.

Classe	**Precisão**	**Recall**	**F1-Score**
Negativo	0.09	0.24	0.13
Neutro	0.04	0.01	0.01
Positivo	0.00	0.00	0.00

Essas descobertas ressaltam a necessidade de uma avaliação criteriosa das técnicas de pré-processamento e modelagem, e a importância de ajustar os métodos às especificidades dos dados e às exigências da tarefa de classificação em mãos.

5 Conclusão

Este trabalho teve como objetivo principal avaliar a eficácia de um algoritmo de classificação semi-supervisionado baseado em grafos para a análise de sentimentos. Focando em um conjunto de dados específico de tweets relacionados à política no estado de Minas Gerais, o estudo explorou diferentes combinações de técnicas de incorporação de palavra e métodos de construção de grafos para compreender como cada abordagem afeta o desempenho final do modelo.

Os experimentos realizados neste estudo permitiram observar várias nuances importantes sobre a aplicação dessas combinações. Ficou evidente que a escolha da representação de atributo-valor em combinação com métodos específicos de construção de grafos tem um papel importante no desempenho global dos modelos. Cada combinação influenciou a capacidade do modelo de interpretar e classificar corretamente os sentimentos expressos nos tweets.

Dentre as combinações testadas, o modelo que empregou Word2Vec em conjunto com E-Vizinhança destacou-se, obtendo os melhores resultados em termos de precisão, recall e F1-score. Essa eficiência pode ser atribuída à habilidade do Word2Vec em capturar nuances contextuais do texto e à eficácia do E-Vizinhança em formar grafos que refletem essas nuances de forma eficiente. Por outro lado, a combinação de TF-IDF com E-Vizinhança apresentou um desempenho notavelmente inferior, sugerindo que essa abordagem pode não ser ideal para capturar a complexidade e a variedade dos sentimentos expressos nos tweets. Isso pode ser devido à natureza do TF-IDF, que, embora eficiente em destacar palavras importantes, pode não capturar as relações contextuais e semânticas tão eficazmente quanto o Word2Vec.

Foi observado também que os modelos tiveram desempenhos variados na identificação de diferentes tipos de sentimentos. Essa variação destaca a complexidade da análise de sentimentos e a necessidade de ajustar modelos e técnicas conforme o tipo de sentimento e o contexto do texto.

Os resultados obtidos neste estudo reforçam a relevância de um planejamento cuidadoso na escolha de métodos de pré-processamento e estratégias de modelagem em tarefas de aprendizado semi-supervisionado. A eficiência demonstrada pelo Word2Vec em combinação com E-Vizinhança abre caminhos para sua aplicação em outras áreas de análise de sentimentos, sugerindo uma abordagem promissora para a extração de *insights* valiosos de dados textuais.

5.1 Trabalhos Futuros

Para expandir a pesquisa realizada neste trabalho, propõe-se a seguinte agenda para estudos futuros:

- Testar as abordagens em diferentes conjuntos de dados para validar a generalização dos resultados;
- Realizar uma análise mais aprofundada sobre a influência dos hiperparâmetros na eficácia dos modelos, como os k vizinhos e a configuração dos vetores do Word2Vec;
- Avaliar o impacto de novas representações vetoriais na análise de sentimentos, como o FastText, extensão do Word2Vec que leva em consideração a estrutura interna das palavras; e o BERT, modelo de *deep learning* que calcula as ponderações entre entrada e saída dinamicamente;
- Explorar a aplicabilidade de diferentes algoritmos de aprendizado semi-supervisionado na análise de sentimentos.

Referências

BERTON, L. *Construção de redes baseadas em vizinhança para o aprendizado semissupervisionado*. Tese (Doutorado) — Universidade de São Paulo, 2016. Citado 4 vezes nas páginas 20, 24, 26 e 27.

BRITO, M. et al. Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection. *Statistics Probability Letters*, v. 35, n. 1, p. 33–42, 1997. ISSN 0167-7152. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167715296002131>. Citado 2 vezes nas páginas 25 e 27.

CALDER, J. et al. Poisson learning: Graph based semi-supervised learning at very low label rates. In: III, H. D.; SINGH, A. (Ed.). *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 2020. (Proceedings of Machine Learning Research, v. 119), p. 1306–1316. Disponível em: <https://proceedings.mlr.press/v119/calder20a.html>. Citado na página 28.

ERTEL, W. *Introduction to Artificial Intelligence*. [S.l.]: Springer, 2018. Citado na página 23.

FACELI, K. et al. *Inteligência Artificial: Uma abordagem de Aprendizagem de Máquina*. [S.l.]: LTC, 2011. Citado 2 vezes nas páginas 24 e 30.

GUIDO S.; MULLER, A. C. *Introduction to Machine Learning with Python: a guide for data scientists*. [S.l.]: O'reilly, 2016. Citado 2 vezes nas páginas 23 e 30.

HARTIGAN, J. A. Statistical theory in clustering. *Journal of Classification*, v. 2, n. 1, p. 63–76, 12 1985. ISSN 1432-1343. Disponível em: <https://doi.org/10.1007/BF01908064>. Citado na página 31.

MAJEED, A.; RAUF, I. Graph theory: A comprehensive survey about graph theory applications in computer science and social networks. *Inventions*, v. 5, n. 1, 2020. ISSN 2411-5134. Disponível em: <https://www.mdpi.com/2411-5134/5/1/10>. Citado na página 25.

MITCHELL, T. M. *Machine Learning*. [S.l.]: McGraw-Hill Education(ISE Editions), 1997. Citado 2 vezes nas páginas 23 e 30.

RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. [S.l.]: Pearson, 2016. Citado na página 23.

TORELLI, J. C. *Implementação paralela da transformada de distância euclidiana exata*. 2005. Citado na página 26.

VEGA-OLIVEROS, D. A. et al. Regular graph construction for semi-supervised learning. In: IOP PUBLISHING. *Journal of physics: Conference series*. [S.l.], 2014. v. 490, n. 1, p. 012022. Citado 2 vezes nas páginas 24 e 26.

WAN, S. et al. *Contrastive Graph Poisson Networks: Semi-Supervised Learning with Extremely Limited Labels*. [s.n.], 2021. Disponível em: <https://proceedings.neurips.cc/paper/2021/file/31c0b36aef265d9221af80872ceb62f9-Paper.pdf>. Citado na página 28.

ZHU, X. *Semi-supervised learning literature survey*. [S.l.], 2005. Citado 2 vezes nas páginas 25 e 28.