UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

MARIANA ZAGATTI SABINO

AVALIAÇÃO DE MÉTODOS DE CONSTRUÇÃO DE REDES NA CLASSIFICAÇÃO SEMI-SUPERVISIONADA DE TEXTOS

TRABALHO DE CONCLUSÃO DE CURSO

SÃO CARLOS
2021

MARIANA ZAGATTI SABINO

AVALIAÇÃO DE MÉTODOS DE CONSTRUÇÃO DE REDES NA CLASSIFICAÇÃO SEMI-SUPERVISIONADA DE TEXTOS

Trabalho de conclusão de curso apresentado como requisito parcial para a obtenção do título de Bacharel em Engenharia de Computação pela Universidade Federal de São Carlos. Orientador: Prof. Dr. Alan Demétrius Baria Valejo (Universidade Federal de São Carlos).

SÃO CARLOS

2021

AGRADECIMENTOS

Agradeço a meus pais, Marcia e Pedro, e a meu irmão, Gabriel, meus principais apoiadores na busca pela formação acadêmica, e a eles dedico as minhas conquistas.

Agradeço a meu orientador, professor Alan, tendo sido ele essencial na realização do trabalho mais importante da minha graduação, por toda ajuda, disponibilidade e paciência prestadas a mim.

Agradeço também aos demais professores que passaram por meu caminho, por terem guiado meu aprendizado e contribuído para a formação do meu conhecimento. Muitas vezes, inclusive, me ofereceram ainda mais do que isso, tendo me dado o privilégio de seus conselhos e de suas amizades.

Agradeço a meus amigos e colegas de curso, companheiros inestimáveis nessa jornada; sem eles tudo teria sido muito mais difícil. A troca de experiências e o aprendizado conjunto foram fundamentais para meu crescimento.

Agradeço também a meu namorado Guillermo, pelo apoio e carinho incondicionais.

RESUMO

Devido à grande quantidade de dados produzidos diariamente no formato de texto, seja publicamente em redes sociais ou de forma privada dentro de empresas, há a necessidade de analisá-los e extrair deles informação. O objetivo é transformá-los em ferramentas úteis, como sistemas de tradução e assistentes virtuais. A área de processamento de linguagem natural, em conjunto com o aprendizado de máquina, fornece as tecnologias necessárias para tal objetivo. Uma das tarefas mais exploradas nesse contexto é a classificação de textos em categorias de interesse, sendo que dentre as diversas abordagens existentes nessa área, destacam-se os algoritmos de aprendizado semi-supervisionado, em específico os transdutivos que recebem como entrada dados em formato de redes e retornam os dados rotulados. Essa estratégia necessita, inicialmente, da construção de uma rede a partir dos dados analisados, sendo que diversos algoritmos podem ser utilizados para esse propósito, os quais produzem redes com características topológicas distintas, interferindo diretamente na acurácia da classificação. Nesse contexto, o objetivo deste estudo é analisar a influência dos algoritmos de construção de redes na classificação semi-supervisionada de texto. Foi feita uma avaliação empírica em coleções reais de documentos. Os resultados apontaram que algoritmos que não geram redes regulares e utilizam a distância cosseno, que é mais adequada para dados textuais, performam melhor. São eles: *k*-NN, Epsilon, GBLP e M*k*-NN.

Palavras–chave: Aprendizado de Máquina. Processamento de Linguagem Natural. Aprendizado Semi-Supervisionado. Inteligência Artificial. Construção de Redes.

ABSTRACT

Due to the shear amount of data produced daily in text format, being it publicly on social media or privately inside enterprises, there is a growing need to analyze and extract information from them. The objective is to transform this data into useful tools such as translation systems and virtual assistants. The area of Natural Language Processing, together with Machine Learning, provides the necessary technologies for such objective. One of the most explored tasks in this context is text classification. Between the diverse approaches existing in this area, semi-supervised learning algorithms stand out. Specifically, transductive algorithms, that receive as input data in the form of networks and return labeled data. This strategy needs the initial construction of a network based on the analyzed data, a task for which many algorithms can be used, producing networks with different topological characteristics, interfering directly in the classification accuracy. In this context, the objective of this study is to analyze the influence of network-building algorithms on semi-supervised text classification. An empirical evaluation on real document collections was carried out. The results point that algorithms that generate non-regular networks have a better overall performance, furthermore algorithms that allow the use of the cosine metric, more suitable for text-based data, performed better than those that don't. These methods are: *k*-NN, Epsilon, GBLP and M*k*-NN.
Keywords: Machine Learning. Natural Language Processing. Semi-Supervised Learning. Artificial Intelligence. Network Construction.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Enquanto a internet está mais do que nunca presente em todas as áreas da sociedade, desde a vida pessoal até as tecnologias utilizadas por empresas e governos, a quantidade de informação disponibilizada na internet, especialmente em formato de textos, também aumenta a cada dia; esteja ela disponível publicamente em redes sociais ou apenas em bancos de dados de sistemas privados.

Para lidar com tamanha quantidade de dados em formato de texto, extrair deles significado e transformá-los em informações úteis para diversas aplicações, a área de processamento de linguagem natural (PLN) apresenta as ferramentas e tecnologias necessárias. Algumas aplicações de PLN envolvem análise de sentimentos e mineração de opinião, trabalhando, por exemplo, com postagens em redes sociais. *Chatbots* e outros tipos de ferramentas de interação automática entre uma empresa e seus clientes são outro exemplo de aplicações, com o objetivo de diminuir a necessidade da presença de seres humanos, por exemplo, em horários fora do comercial, mantendo ainda assim as respostas às interações assertivas. Outra área envolvida diretamente com o PLN é a tradução e interpretação, uma vez que são muito populares ferramentas que mudam o idioma de um texto para outra língua. Além disso, o PLN faz possível a interação de seres humanos com dispositivos por meio de voz, como por exemplo, as assistentes virtuais Siri, da Apple, Cortana, da Microsoft, e Alexa, da Amazon, que reagem a comandos e perguntas de usuários. Outra aplicação importante é a classificação automática de textos em categorias (temas ou tópicos) de interesse, permitindo que os mesmos sejam encontrados com facilidade e relacionados com textos que abordam temas em comum.

A classificação automática de textos é uma das tarefas amplamente explorada na área de PLN, entretanto, é notório que a quantidade de dados textuais cresce exponencialmente com o passar o dos anos, sendo produzida e armazenada diariamente uma quantidade massiva desses dados, tais como e-mails, postagens em redes sociais, artigos em blogs e sites de notícias, artigos científicos, dentro outros. Portanto, essa tarefa é realizada por meio de algoritmos de aprendizado de máquina, subárea da inteligência artificial, que requerem pouca intervenção humana e que trabalham com algoritmos que modificam seu comportamento automaticamente com base em suas experiências, na busca pela melhor solução de um problema. Os algoritmos de aprendizado de máquina para classificação de documentos estão divididos em três categorias: supervisionados, não supervisionados e semi-supervisionados. No primeiro caso, considera-se apenas documentos

rotulados para induzir um modelo e classificar novos documentos. Na segunda categoria, os algoritmos inferem uma divisão natural para os documentos, com base na estrutura topológica e informações adicionais relacionadas aos dados não rotulados. Na última categoria, considera-se documentos rotulados e não rotulados para classificar novos documentos.

Na classificação semi-supervisionada de textos, os métodos baseados em redes vêm ganhando notoriedade e produzindo resultados expressivos em comparação com algoritmos tradicionais da literatura (ROSSI; LOPES; REZENDE, 2016). Entretanto, a maioria desses métodos necessitam da construção de uma rede a partir dos dados analisados. Para o objetivo de construção de redes, existem diversos algoritmos que podem ser utilizados, os quais possuem estratégias diferentes e produzem redes com características topológicas distintas.

Por isso, faz-se necessária a existência de pesquisas no sentido de analisar os métodos de construção de redes e avaliar se existe uma diferença significativa na precisão dos algoritmos de classificação semi-supervisionada ao serem usados cada um desses métodos. Além disso, é importante verificar quais estratégias de construção de redes são mais eficientes ao lidarem com dados textuais.

1.1 Objetivo Geral

O objetivo do presente trabalho é analisar algoritmos de construção de redes e avaliar seu desempenho ao serem utilizados em conjunto com algoritmos de aprendizado semi-supervisionado na classificação de conjuntos de dados em formato de texto.

1.2 Objetivos Específicos

Tratam-se de objetivos específicos os seguintes tópicos:

- Pesquisar o estado da arte em relação a algoritmos de construção de redes, bem como de algoritmos de classificação semi-supervisionada, considerando o escopo de classificação de dados textuais.
- Obter conjuntos de dados em formato de texto sobre temas diversos, divididos em classes, processados com técnicas de pré-processamento de textos.

- Construir redes a partir dos dados textuais, utilizando diversos algoritmos com essa finalidade, e variando seus parâmetros em busca de obter o melhor resultado possível.
- Avaliar os algoritmos de classificação semi-supervisionada nas redes geradas pelos algoritmos de construção de redes.
- Analisar os resultados obtidos, buscando encontrar os algoritmos de construção de redes que levam a uma maior acurácia na classificação de textos.

1.3 Organização do Trabalho

O trabalho está organizado da forma a seguir:

- No Capítulo 1, é apresentada a definição do problema, bem como objetivos geral e específicos.
- No Capítulo 2, é feita a fundamentação teórica do estado da arte sobre inteligência artificial, aprendizado de máquina, algoritmos de aprendizagem semi-supervisionada, processamento de linguagem natural, mineração de texto, pré-processamento de dados textuais, redes e algoritmos de construção de redes; também é feita uma breve revisão de literatura com trabalhos relacionados.
- No Capítulo 3, disserta-se sobre as metodologias utilizadas nesta pesquisa, são dadas informações sobre os conjuntos de dados e os algoritmos escolhidos, além de explicações sobre a modelagem e as ferramentas de análise.
- No Capítulo 4, são discutidos os experimentos e os resultados obtidos por meio deles, além de ser feita uma avaliação comparativa entre os algoritmos selecionados.
- Por fim, o Capítulo 5 apresenta as conclusões e as contribuições do trabalho para o estado da arte, bem como sugestões para futuras pesquisas.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Inteligência artificial

A primeira definição do termo “inteligência artificial” (IA), cunhada em 1955 por John McCarthy, foi: “O objetivo da IA é desenvolver máquinas que se comportam como se fossem inteligentes”.

Conforme escrito por Ertel (2011), algumas das principais qualidades dos seres humanos, que os definem como inteligentes, são a capacidade de adaptação e a habilidade de ajustar-se a diversos ambientes, mudando seu comportamento de acordo com a situação ao aprender com o meio.

De acordo com Russell et al. (1995), a meta de parte dos estudos realizados na área de IA é aprender mais sobre os seres humanos e tentar entender como seu cérebro funciona, para modelar no computador a automatização de ações que os definem como inteligentes. Com isso, é possível construir entidades que mostram comportamento inteligente, como o de uma pessoa.

Tais entidades, ao serem capazes de realizarem atividades associadas com o pensamento humano, como a tomada de decisões, a resolução de problemas e a aprendizagem, são úteis ao atuar na resolução de problemas específicos.

Ainda de acordo com Ertel (2011), muitos princípios no campo de redes neurais têm como origem a área de neurociência, que estuda o cérebro. No entanto, existe uma outra abordagem que não se prende à forma como humanos resolvem problemas. Nesse cenário, o foco está em buscar uma solução ótima (ou a melhor solução possível) para o problema a ser resolvido, sem fixar um método. Nota-se que na IA não existe um método universal capaz de resolver qualquer problema, mas sim uma grande variedade de soluções efetivas, que dependem da aplicação pretendida.

A IA engloba áreas mais gerais, como percepção e raciocínio lógico, e também tarefas mais específicas, como diagnosticar doenças ou provar teoremas matemáticos.

2.2 Aprendizado de máquina

Justamente porque a capacidade de aprender dos seres humanos, que é uma das características que os definem como inteligentes, ser tão superior, o aprendizado de máquina é um dos principais subcampos da inteligência artificial, segundo Burkov (2019).

O aprendizado de máquina é a disciplina que foca em solucionar um determinado problema prático construindo modelos que se baseiam em uma coleção de exemplos. A partir de uma determinada coleção de exemplos, é criado um modelo estatístico. Os exemplos utilizados, que são relativos ao problema em questão, podem ter sido coletados da natureza, criados por seres humanos ou até mesmo gerados por outros algoritmos, dependendo do caso.

Os algoritmos no ramo de aprendizado de máquina são classificados em supervisionados, não supervisionados, ou semi-supervisionados. No escopo deste trabalho, são utilizados algoritmos semi-supervisionados. No entanto, todos os tipos de classificação de dados são brevemente apresentados a seguir.

2.2.1 Aprendizagem supervisionada

Ainda de acordo com Burkov (2019), na aprendizagem supervisionada, os exemplos contidos no conjunto de dados, identificados por $\{(\mathbf{x}_i, y_i)\}$, com i = 0, 1, ..., N, são rotulados, ou seja, suas classes são conhecidas.

Cada elemento x_i do conjunto de dados é um vetor de características. O que descreve esses elementos é que em cada dimensão do vetor há um valor, chamado de característica, que descreve o exemplo. Para todos os exemplos contidos na coleção, na mesma posição do vetor há sempre a mesma informação. Por exemplo, para o caso de o vetor representar dados de uma pessoa, as características contidas nos vetores podem ser altura, peso ou gênero.

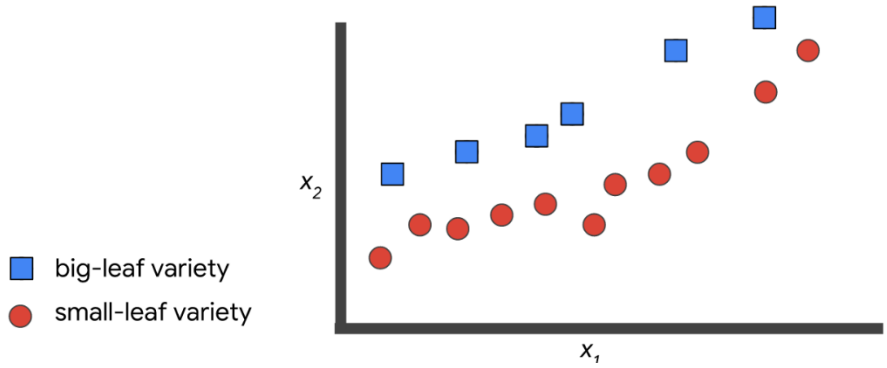
O objetivo da aprendizagem supervisionada é produzir, a partir do conjunto de dados, um modelo que recebe um vetor de características como entrada, e produz como saída informação para deduzir um rótulo.

Um exemplo contido em Burkov (2019) seria receber um vetor descrevendo uma pessoa e retornar como saída a probabilidade de ela ter uma determinada doença. Outro exemplo seria deduzir a partir da temperatura ambiente a receita da venda de sorvete em um supermercado.

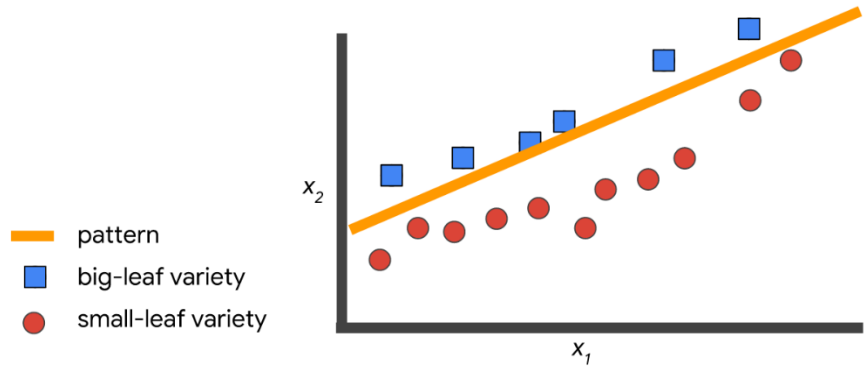
De acordo com Honda et al. (2017), algumas das principais técnicas de aprendizado supervisionado em aprendizado de máquina são regressão linear, redes neurais artificiais, árvores de decisão, *k*-vizinhos mais próximos e Naïve Bayes.

A imagem a seguir exemplifica o funcionamento de um algoritmo de aprendizagem supervisionada capaz de identificar variedades de folhas. Inicialmente, é treinado a partir de dados rotulados, como visto na Figura 1 (a), na qual os pontos no gráfico representam os dados divididos em duas classes, sendo que cada classe tem uma cor e uma forma geométrica. Em seguida, é criado um modelo para identificação dos dados, como visto na Figura 1 (b). Posteriormente, ele é capaz de classificar novos dados não rotulados, como ilustrado na Figura 1 (c).

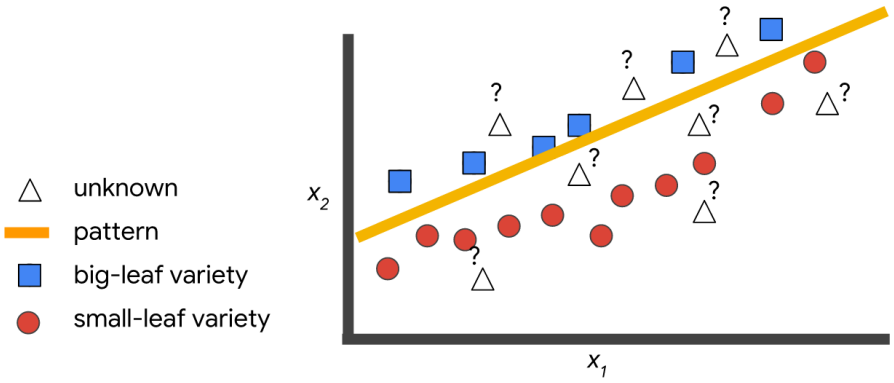
Figura 1 – Funcionamento de um algoritmo de aprendizagem supervisionada.



(a) Dados rotulados.



(b) Modelo induzido pelo algoritmo.



(c) Novos dados a serem classificados.

Fonte: GOOGLE DEVELOPERS, 2021.

2.2.2 Aprendizagem não supervisionada

Burkov (2019) explica que, na aprendizagem não supervisionada, os dados contidos no conjunto de dados, identificados por $\{\mathbf{x_i}\}$, com i = 0, 1, ..., N, não são rotulados.

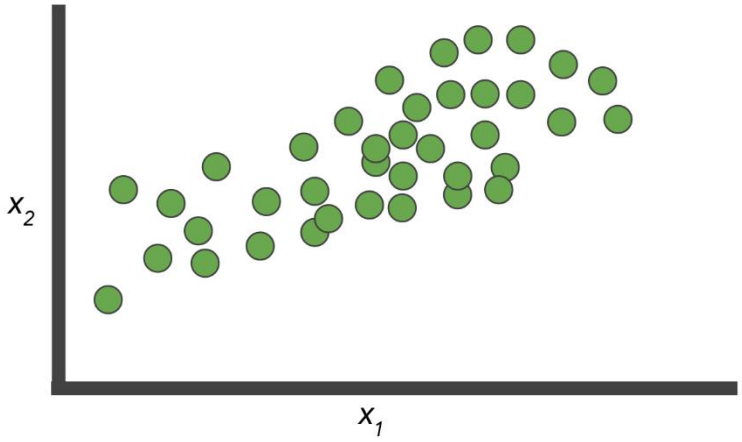
O objetivo com essa abordagem é tomar os vetores de características x_i como entrada e transformá-los em outro vetor ou em um valor, para resolver um problema prático, representando os dados de maneira mais condensada e informativa (HONDA; FACURE; YAOHAO, 2017).

A técnica de redução de dimensionalidade, por exemplo, retorna o próprio vetor de características em uma versão reduzida. Já em modelos de agrupamento, em que os dados são separados em grupos, é retornado, para cada vetor de características do conjunto de dados, um identificador, geralmente o *id*, referente ao grupo em que o dado foi alocado. Por outro lado, na detecção de *outliers*, o algoritmo calcula e retorna um número real referente a quanto um determinado exemplo é diferente de um exemplo "típico" no conjunto de dados.

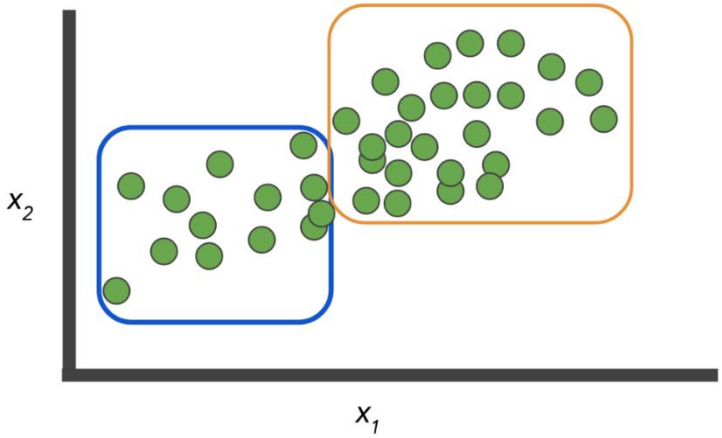
Um exemplo de aplicação para resolução de um problema real seria obter registros de compras de consumidores de um certo estabelecimento e usar a aprendizagem não supervisionada para dividi-los em grupos com perfis diferentes de consumo.

A Figura 2 ilustra a separação de dados em grupos com o uso de um algoritmo de classificação não supervisionada. Inicialmente, como na Figura 2 (a), o algoritmo recebe como entrada dados não rotulados. Em seguida, divide os dados em dois grupos, como na Figura 2 (b). Posteriormente, quando recebe novos dados como entrada, é capaz de rotular os dados com base nos grupos previamente definidos, como na Figura 2 (c).

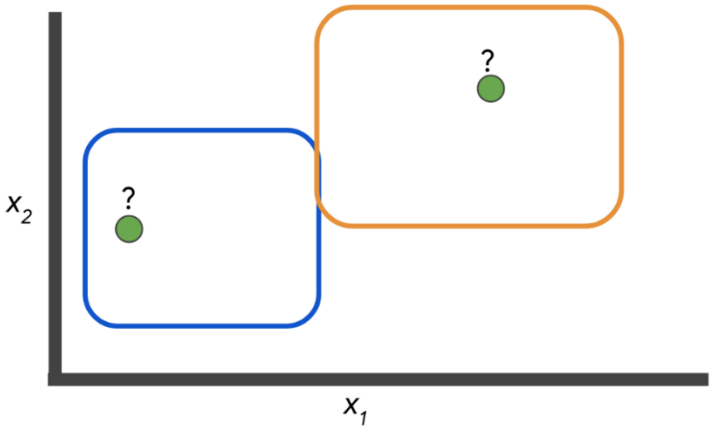
Figura 2 – Exemplo de funcionamento de algoritmo de aprendizagem não supervisionada.



(a) Dados iniciais.



(b) Agrupamento dos dados.



(c) Classificação de novos dados.

Fonte: GOOGLE DEVELOPERS, 2021.

2.2.3 Aprendizagem semi-supervisionada

Na aprendizagem semi-supervisionada, que se trata da abordagem mais relevante para o presente trabalho, o conjunto de dados contém exemplos com rótulos e sem rótulos, usualmente sendo o último em maior quantidade (BURKOV, 2019).

A motivação, de acordo com Sanches (2003), é que exemplos rotulados são muito mais escassos em comparação com dados não rotulados, que existem em abundância.

Grande parte dos algoritmos de aprendizado semi-supervisionado utilizados atualmente são variações dos algoritmos que já eram existentes na literatura nas outras categorias (supervisionados e não supervisionados), modificados de forma a lidar com os dois tipos de dados.

Esse tipo de aprendizagem de máquina pode ser usado tanto em tarefas de classificação como de agrupamento. No caso da classificação, utiliza-se alguns exemplos de dados não rotulados na fase de treinamento do classificador; para isso, rotula-se esses dados, tendo uma certa margem de segurança. Esse processo resulta em um modelo com maior precisão computado pelo algoritmo. Já ao utilizar dados rotulados no agrupamento, que faz parte do aprendizado de máquina não supervisionado, esses exemplos servem para prover um conhecimento prévio que guia o processo, por exemplo, na forma de restrições para as divisões dos grupos. Espera-se que o uso desse processo resulte em melhores grupos.

2.3 Processamento de linguagem natural

De acordo com Pereira (2011), o processamento de linguagem natural (PLN) é uma subárea da inteligência artificial que trata do desenvolvimento de modelos computacionais com o objetivo de completar tarefas que se relacionam com informação expressa em linguagem escrita ou falada, como analisar, traduzir, interpretar e sintetizar textos.

Para isso, as pesquisas em PLN têm o foco nos seguintes aspectos da comunicação em linguagem natural: a fonologia, ou seja, o som que compõe as palavras faladas em um idioma; a morfologia, que estuda a estrutura das palavras em termos das unidades primitivas pelas quais são formadas; a sintaxe, que é a maneira como os termos em uma frase se relacionam, a semântica, que é o significado de uma palavra; e a pragmática, que verifica se o significado atribuído é apropriado no contexto.

O termo “natural” se refere à linguagem humana em qualquer idioma, em detrimento de, por exemplo, linguagem matemática, notação lógica ou científica, ou linguagens de programação computacionais, conforme escrito por Jackson et al. (2002).

Essa área desperta um grande interesse comercial, uma vez que o mercado de informação é o que mais cresce no mundo. Mais do que nunca, a quantidade de informação na internet vem crescendo. A maior parte da informação criada ou disponibilizada na internet ainda é expressa em linguagem e está disponível no formato de textos, em comparação com, por exemplo, imagens, sons, vídeos ou equações.

Muitas das informações que posteriormente vão pertencer a bancos de dados relacionais formalmente estruturados tem essa origem, sendo necessária, portanto, assistência para isso. Logo, conclui-se que o problema está não numa indisponibilidade de informação, mas sim na necessidade de existência e disponibilidade de ferramentas para organizá-la e oferecê-la no momento adequado. Assim, o PLN exerce um papel muito importante.

Ainda de acordo com Jackson et al. (2002), são alguns usos para a PLN: a obtenção de documentos eletrônicos, a extração de informações, a categorização de textos, e a mineração de textos.

2.4 Mineração de texto

Segundo Vijayarani et al. (2015), a mineração de texto é a atividade por meio da qual são processados dados no formato de texto para distinguir padrões e realizar a extração de informações relevantes.

Ela se diferencia, por exemplo, da categorização de documentos, porque essa não gera informações novas nem detecta padrões, se não que apenas faz uma organização dentro de conceitos já existentes. Autoridades no assunto concordam que idealmente a mineração deve trazer algo interessante sobre o texto, que o relacione com o mundo real (JACKSON; MOULINIER, 2002).

Observa-se que a mineração de texto tenta resolver limitações em atividades de processamento de linguagem natural. Por outro lado, é necessário utilizar algumas ferramentas de PLN para realizar a mineração de texto.

As ferramentas de mineração de texto são capazes de trabalhar com dados como documentos HTML, e-mails ou documentos de textos. Esses podem ser dados não estruturados, ou seja, que não estão em um banco de dados estruturado tradicional, e também podem ser dados semiestruturados, que também não estão contidos em bancos de dados estruturados, mas que por sua vez não são dados completamente não tratados (VIJAYARANI; ILAMATHI; NITHYA, 2015).

De acordo com Alvares (2014), o processo de mineração de textos pode ser dividido em três fases: a preparação dos dados, a extração de conhecimento e o pós-processamento.

A etapa em que os dados são preparados é realizada para tratar o conjunto de dados textuais do qual será extraído conhecimento, melhorando sua qualidade. O objetivo é fazer com que esses dados representem a maior quantidade possível de características relevantes dos documentos.

A seguir, serão apresentadas especificamente as etapas que constituem a preparação (ou pré-processamento) dos dados.

2.4.1 Tokenização

A tokenização divide um fluxo de caracteres entre tokens, que podem ser palavras, pontuação, números e outros itens, conforme visto em Kibble (2013). Ela é uma das operações mais básicas possíveis de serem realizadas em um texto.

Um exemplo seria a cadeia de caracteres: “Nuvem gigante de poeira encobre cidades no interior de SP; veja vídeo.”. A tokenização dessa cadeia de caracteres seria a seguinte, colocando cada token entre aspas duplas: “Nuvem”, “gigante”, “de”, “poeira”, “encobre”, “cidades”, “no”, “interior”, “de”, “SP”, “;”, “veja”, “vídeo” e “.”.

2.4.2 Remoção de *stopwords*

Como são analisadas quantidades expressivas de documentos e os textos em questão podem conter uma quantidade muito grande de palavras, é necessário remover termos que não são úteis para a análise, com o objetivo de torná-la mais rápida e precisa. Alguns exemplos são as palavras que constam em todos os documentos, as chamadas *stopwords*, por exemplo, palavras como “era”, “seu”, “eu”, “aquela” e “que”. Algumas das palavras mais comuns em documentos tratam-se de artigos, preposições e pronomes (VIJAYARANI; ILAMATHI; NITHYA, 2015).

Outros termos que devem ser removidos são aqueles que não são úteis para distinguir o conteúdo e separá-lo entre classes. Por exemplo, “esporte” não auxiliaria a distinguir documentos entre os temas “futebol” e “basquete”.

2.4.3 *Stemming*

Os algoritmos de *stemming* removem prefixos e sufixos das palavras, assim identificando similaridades no texto em função da morfologia das palavras e, portanto, diminuindo o número de atributos dos documentos.

O objeto do *stemming* é transformar os termos individuais em uma representação genérica o suficiente para englobar diversas variantes de uma palavra, tomando cuidado para que sua essência ainda seja compreendida. Essa representação é chamada de *stem*. Um exemplo seria “cuid”, um *stem* para palavras como “cuidei”, “cuidado” e “cuidar”.

Existem dois tipos de erros que devem ser prevenidos nos algoritmos de *stemming*, o *overstemming* e o *understemming*. No primeiro, *overstemming*, são removidas letras demais, o que faz com que palavras com sentidos diferentes apontam para o mesmo *stem*. Um exemplo seria o *stem* “comp” para as palavras “computador” e “comparar”. No segundo, *understemming*, não são removidas letras o suficiente, e assim mais de um *stem* surgem para palavras com o mesmo sentido (ALVARES, 2014).

2.4.4 Conversão de valores simbólicos para numéricos

Segundo das Neves (2003), todas as técnicas de mineração de dados têm a capacidade de lidar com valores numéricos, mas nem todas, como por exemplo as redes neurais, lidam com valores simbólicos. Portanto, é necessário transformar os valores simbólicos dos documentos em suas representações numéricas. A seguir, nas próximas subseções, são apresentados métodos que fazem dita conversão.

2.4.4.1 *Bag of Words*

De acordo com Chiang (2018), o método *Bag of Words* conta quantas vezes cada palavra aparece em um documento. Trata-se de um método extremamente simples do ponto de vista computacional, porém geralmente eficaz, de capturar o conteúdo de um texto. Com ele, desconsidera-se a estrutura das frases, ou seja, a ordem em que as palavras aparecem no documento.

A seguir, a Tabela 1 tem o objetivo de exemplificar a contagem de palavras em textos. São consideradas três avaliações em formato de texto. São elas: “A sopa não estava muito quente.”, “A sobremesa estava muito fria e saborosa.” e “A sopa estava fria.”. Nelas, existe a aparição de dez palavras distintas. Na tabela, são expressas a quantidade total de palavras e a quantidade de aparições de cada palavra nas avaliações.

Tabela 1 – Exemplo de contagem de palavras em textos.

	A	sopa	sobremesa	não	estava	muito	quente	fria	e	saborosa	Tamanho da avaliação (em quantidade de palavras)
Avaliação 1	1	1	0	1	1	1	1	0	0	0	6
Avaliação 2	1	0	1	0	1	1	0	1	1	1	7
Avaliação 3	1	1	0	0	1	0	0	1	0	0	4

Fonte: Autora do presente trabalho.

2.4.4.2 TF-IDF

TF-IDF é a sigla para “*term frequency - inverse document frequency*”, em uma tradução livre, “frequência do termo - frequência inversa nos documentos”.

Esse método calcula a frequência relativa de uma palavra em um documento, e a compara com a proporção dessa palavra em todo o conjunto de documentos. Termos que aparecem com frequência em muitos documentos são considerados menos relevantes em comparação com termos que aparecem apenas em um grupo pequeno de documentos (RAMOS, 2003).

A seguir, é apresentado na Figura 3 um exemplo de cálculo na abordagem TF-IDF, retirado de Tripathi (2018). São consideradas duas frases, A e B, respectivamente, sendo elas “*The car is driven on the road*”, e “*The truck is driven on the highway*”. As colunas TF A e B mostram a proporção em que cada palavra aparece nas duas frases. Já nas outras colunas é calculada a

relevância de cada termo em A e B. É possível notar que as palavras que aparecem nas duas frases não são consideradas nada relevantes.

Figura 3 - Cálculo de TF-IDF em duas frases.

Word	TF		IDF	TF*IDF	
	A	B		A	B
The	1/7	1/7	log(2/2) = 0	0	0
Car	1/7	0	log(2/1) = 0.3	0.043	0
Truck	0	1/7	log(2/1) = 0.3	0	0.043
Is	1/7	1/7	log(2/2) = 0	0	0
Driven	1/7	1/7	log(2/2) = 0	0	0
On	1/7	1/7	log(2/2) = 0	0	0
The	1/7	1/7	log(2/2) = 0	0	0
Road	1/7	0	log(2/1) = 0.3	0.043	0
Highway	0	1/7	log(2/1) = 0.3	0	0.043

Fonte: TRIPATHI, 2018.

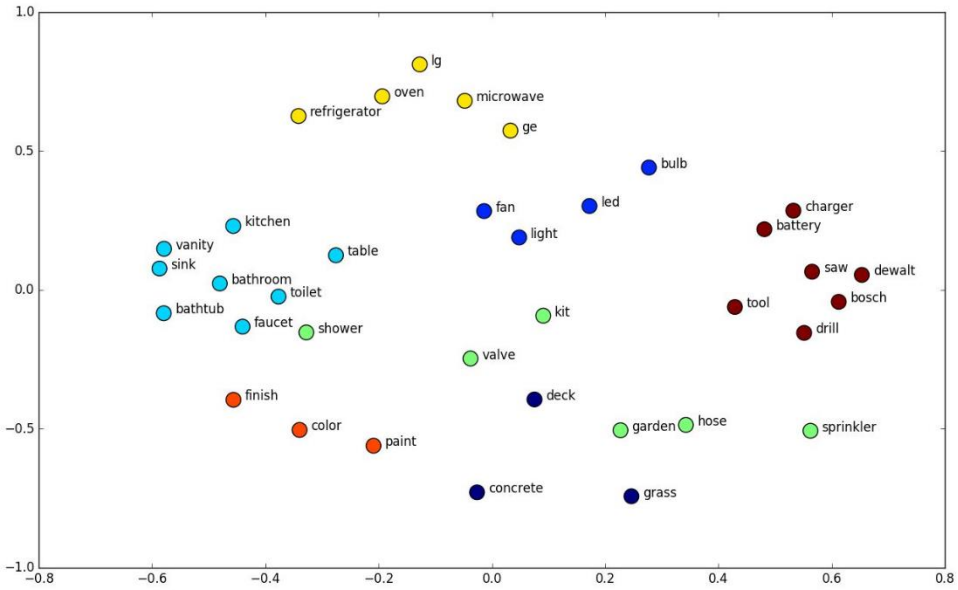
2.4.4.3 *Word Embedding*

Enquanto no método *Bag Of Word*s cada palavra é representada por um número inteiro, o método de *Word Embedding* é um conjunto de técnicas em que cada palavra obtida de um vocabulário não rotulado é representada por um vetor de números reais. Este contempla diversos aspectos da palavra, dentre elas os significados semântico e sintático. O vetor de características de cada termo faz com que ele seja mapeado em um determinado espaço real (WANG; WANG; CHEN; WANG; KUO, 2019).

De acordo com Bengio et al. (2003), ao usar o *Word Embedding*, palavras similares como “cachorro” e “gato”, “está” e “estava”, “Berlim” e “Alemanha” terão representações parecidas. Esses termos têm papéis parecidos em uma frase, semântica e sintaticamente, o que justifica terem vetores de características similares. Com esse método, é possível predizer a probabilidade de um termo aparecer na frase, dada uma determinada palavra que o precede (DE CARVALHO, 2018).

A seguir, na Figura 4 é possível ver a representação em $\mathbb{R}^2$ dos vetores de algumas palavras em um espaço real. Nota-se que palavras relacionadas, com características similares, encontram-se próximas na representação.

Figura 4 - Representação em $\mathbb{R}^2$ de um espaço *embedding*.



Fonte: LYNN, 2018.

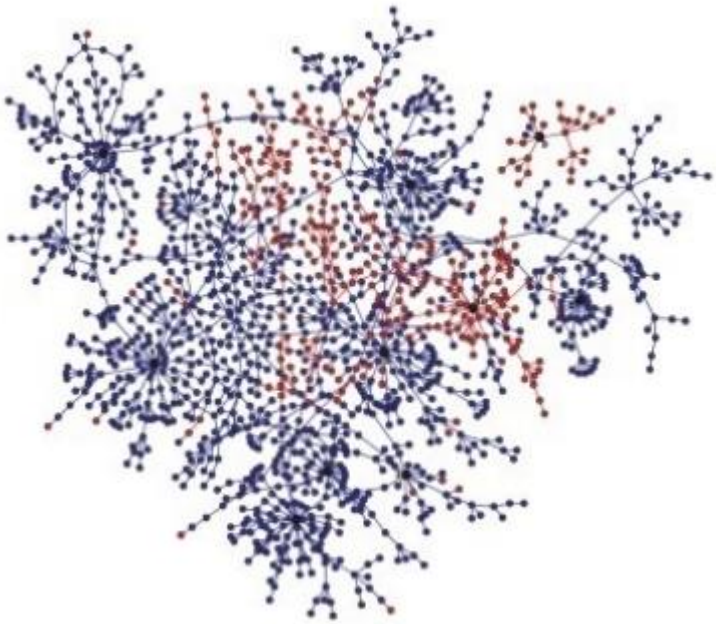
2.5 Redes

Nos métodos baseados em redes, o conjunto de dados é representado por vértices e suas relações por arestas.

Diversos dados da vida real com aspecto relacional podem ser representados em redes, nas quais os vértices são as partes de um sistema e as arestas são as relações estabelecidas entre elas. Observa-se que os dados nesse contexto apresentam correlação, têm semelhanças entre si e possuem ligações.

A Figura 5 ilustra um exemplo de representação de rede, no qual os pontos configuram vértices e as linhas que os conectam são as arestas.

Figura 5 – Exemplo de rede.



Fonte: BERTON, 2016.

De acordo com Berton (2016), algumas das vantagens dos métodos baseados em redes são: o fato de ser possível fazer uma representação da estrutura topológica dos dados; a possibilidade de particionar uma rede em sub-redes em representações hierárquicas; a detecção de agrupamentos (ou comunidades) ou classes arbitrárias; e a combinação de estruturas locais e estatísticas globais. Além disso, mencionam-se: o uso da inferência coletiva com vértices afetando uns aos outros; a

propagação de rótulos, consequência da autocorrelação entre vizinhos; a utilização das características da vizinhança de um vértice; e o fato dessa estratégia poder ser relacionada com pesquisas sobre teoria dos grafos e redes complexas, áreas com muitos estudos formais.

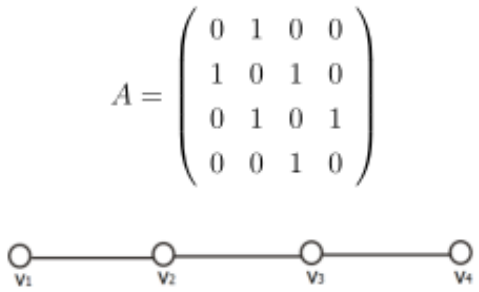
As redes denominadas complexas seguem as seguintes características: apresentam propriedades topológicas como padrão de conexão não trivial; possuem um número muito grande de vértices e arestas; e verifica-se a presença de *hubs*, vértices da rede que possuem muitas conexões em comparação com o resto dos vértices, que possuem poucas.

2.5.1 Definições básicas

Segundo Berton (2016), uma rede $G(V, E)$ representa um conjunto finito de n vértices $V = \{v_1, v_2, ..., v_n\}$ e um conjunto finito de m arestas $E = \{e_1, e_2, ..., e_m\}$, sendo cada aresta um par não ordenado de V. Uma aresta e_k liga os vértices v_i e $v_j \in V$ e é representada por $v_{i,j}$. Se dois vértices estão conectados por uma aresta, são ditos adjacentes ou vizinhos.

A chamada matriz de adjacência de uma rede é uma matriz quadrada de ordem N. Em cada entrada a_{ij}, é representada uma aresta que se inicia no vértice v_i e termina no vértice v_j, relacionando-os, conforme o exemplo da Figura 6.

Figura 6 – Um grafo e sua respectiva matriz de adjacência.

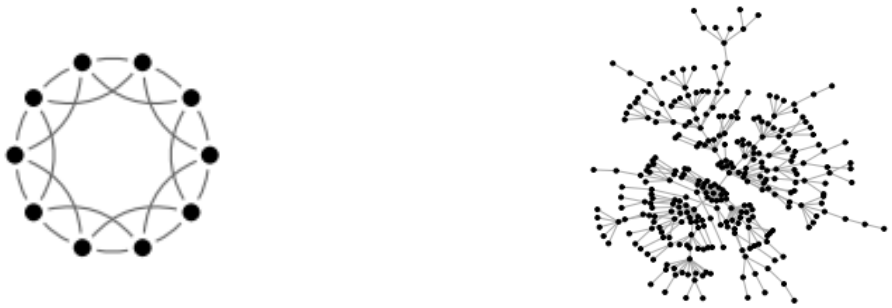


Fonte: BERTON, 2016.

As redes possuem características de conectividade que podem variar, dependendo do conjunto de dados. Redes nas quais todos os vértices possuem o mesmo grau, ou possuem graus próximos uns dos outros, são chamadas de redes regulares, como ilustra a Figura 7 (a). Por outro lado, existem redes nas quais poucos vértices possuem grau muito superior à média, ou seja,

vértices com muitas conexões (também chamados de *hubs*), e muitos vértices com poucas conexões. Essas redes são chamadas redes livres de escala (*scale-free*), como ilustrado na Figura 7 (b).

Figura 7 – Rede regular e rede livre de escala.



(a) (b)

Fonte: VALEJO, 2014.

2.5.2 Construção de redes

Embora alguns sistemas possam ser naturalmente modelados usando redes, devido à sua característica relacional natural, outros dados no formato de atributo-valor ou vetor de características necessitam de algoritmos específicos para a construção das redes. A seguir, são apresentados os algoritmos de construção de redes considerados no presente trabalho.

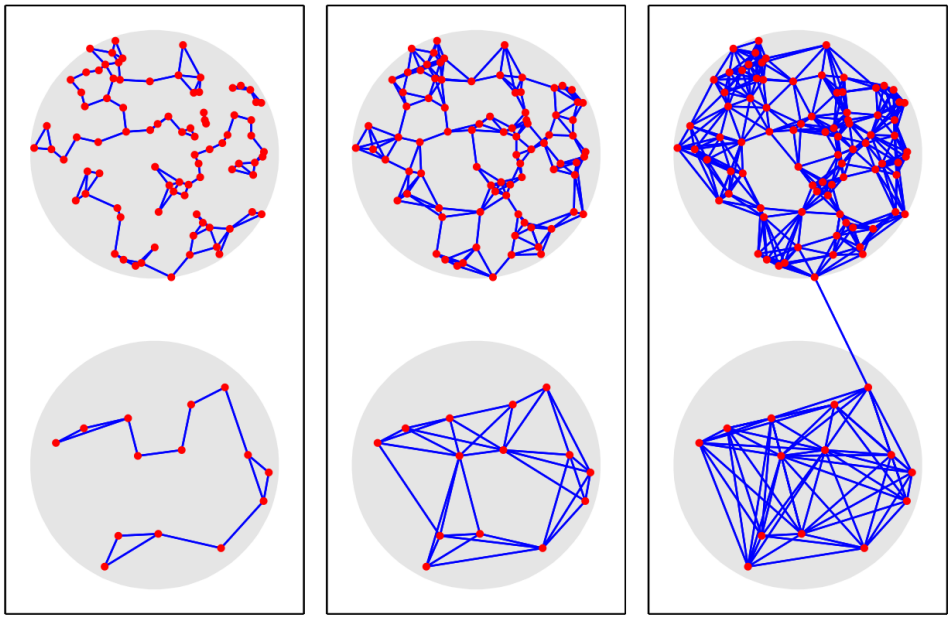
2.5.2.1 *k*-NN

Segundo Maier et al. (2007), a sigla *k*-NN significa *k-nearest neighbors*, em português, *k*-vizinhos mais próximos. O objetivo dessa abordagem é dividir os exemplos considerando suas similaridades.

Considera-se um conjunto de exemplos $X = \{x_1, x_2, ..., x_n\}$, o vizinho mais próximo de um exemplo x_i é um exemplo x_j, sendo $j \neq i$, com a distância mínima de x_i, sendo que várias medidas de distância podem ser utilizadas, por exemplo, a distância euclidiana ou o cosseno. Quanto mais próximos dois vizinhos são, mais similaridades eles compartilham entre si. Considerando tais similaridades, o algoritmo *k*-NN constrói uma rede conectando com uma aresta os *k* vizinhos mais próximos.

Na rede retornada, cada exemplo é representado por um vértice, e os vizinhos mais próximos têm suas conexões representadas pelas arestas. Quanto maior o valor *k*, mais densa em quantidade de arestas será a rede resultante. No exemplo da Figura 8, os pontos em vermelho que representam exemplos estão divididos entre duas classes, localizadas pelos círculos em cinza. À esquerda, $k = 2$, no centro, $k = 4$, e, à direita, $k = 8$. É possível observar que para $k = 2$ a rede superior está desconectada, ou seja, foram identificados mais grupos do que o esperado. Para k = 4 ambos os grupos foram identificados. Por outro lado, com $k = 8$, há uma conexão entre os dois grupos. Isso ilustra a importância da escolha de um *k* adequado. Embora o grau dos vértices seja influenciado pelo valor de *k*, a rede construída pelo algoritmo *k*-NN não é necessariamente regular, pois alguns vértices podem ser o vizinho mais próximo de muitos outros vértices, implicando em um elevado grau, enquanto que outros vértices podem ter grau mais próximo ou igual a *k*.

Figura 8 - Demonstração do uso do algoritmo *k*-NN, com $k = 2$, $k = 4$ e $k = 8$, respectivamente.



Fonte: MAIER; HEIN; LUXBURG, 2007.

2.5.2.2 Epsilon

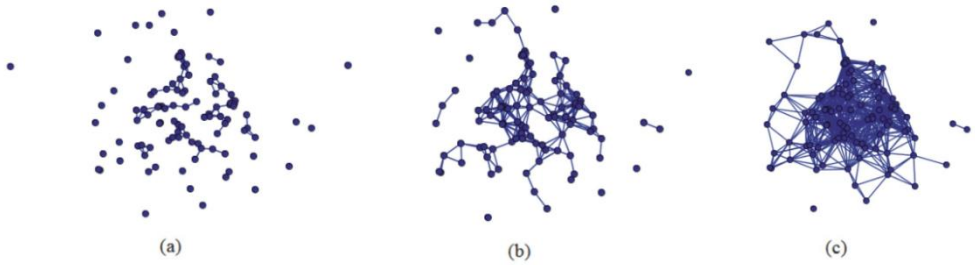
Assim como no método *k*-NN, no Epsilon é considerada a distância, ou similaridade, entre quaisquer dois exemplos. Porém, neste é definida uma constante *epsilon* (ε), sendo que $\varepsilon > 0$, e é adicionada uma aresta entre dois exemplos se a distância entre eles for menor que ε. Em outras

palavras, dado um exemplo, cria-se uma circunferência de raio ε ao redor dele e são adicionadas arestas conectando esse exemplo com todos os outros dentro da circunferência (TALUKDAR, 2009).

Nota-se que é necessária uma escolha adequada de ε para que a rede resultante não seja desconexa. Um problema é que ao aumentar o valor de ε para que a rede seja conexa, alguns vértices podem ter um grau muito grande, ou seja, ter muitas arestas conectadas a ele, o que resulta em uma rede muito densa.

Na Figura 9, é possível ver um exemplo com redes Epsilon, cada rede com uma variação no valor de ε. É possível observar que, quanto mais se aumenta o valor de ε, mais conexões aparecem nos vértices da rede, o que a torna mais densa.

Figura 9 – Redes Epsilon para um conjunto de dados com 100 elementos, com ε = 0.3, ε = 0.5, ε = 0.9, respectivamente.



Fonte: BERTON, 2016.

2.5.2.3 M*k*-NN

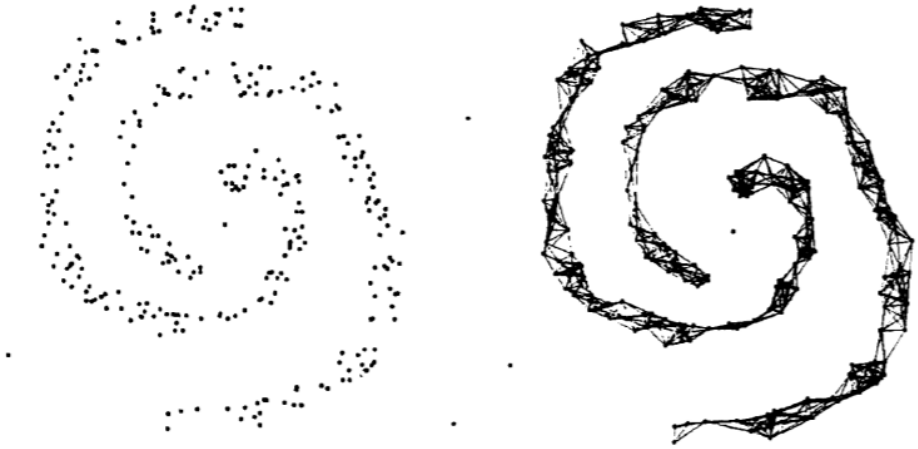
Conforme visto em Maier et al. (2007), *mutual k-nearest neighbors* (M*k*-NN) é uma variação do *k*-NN. Nela, os vértices v_i e v_j estão conectados somente se $v_i \in k\text{-}NN(v_j)$ e $v_j \in k\text{-}NN(v_i)$, ou seja, somente se ambos são mutuamente vizinhos mais próximos.

Note que, embora o M*k*-NN gere uma rede mais esparsa que o *k*-NN, a quantidade de arestas de cada vértice é limitada por *k*, o que previne a rede da existência de vértices com uma quantidade extremamente alta de arestas, ainda que a rede resultante não esteja distante de uma regular.

Esse método pode eliminar a influência de ruídos e de *outliers*, o que traz uma vantagem do M*k*-NN em relação ao *k*-NN.

É possível ver, na Figura 10, a representação de um conjunto de dados e a rede gerada a partir de um conjunto de dados *X* ao ser utilizado o algoritmo M*k*-NN com $k = 11$, ou seja, com cada vértice da rede tendo um limite de no máximo onze vizinhos mutuamente mais próximos.

Figura 10 – Rede gerada a partir de um conjunto de dados utilizando o algoritmo M*k*-NN.



Fonte: BRITO; CHÁVEZ; QUIROZ; YUKICH, 1997.

2.5.2.4 C*k*-NN

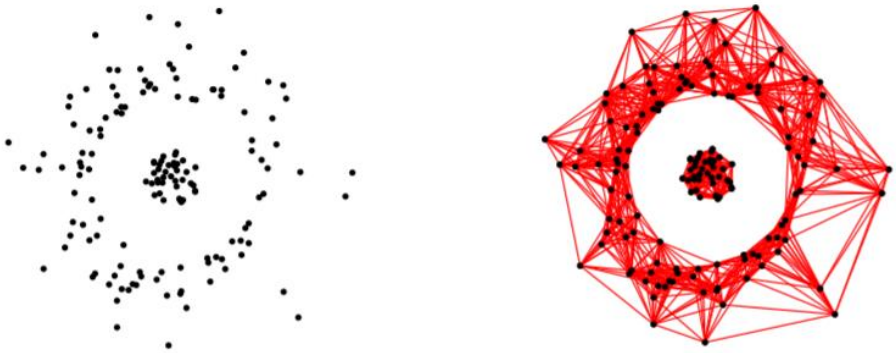
De acordo com Berry (2019), o método de construção de redes *continuous k-nearest neighbors* (C*k*-NN) gera uma rede não direcionada cujas arestas não têm pesos. O fato da rede gerada não possuir peso pode ser uma desvantagem em relação a outros algoritmos que incorporam peso às arestas, como é o caso do *k*-NN e o M*k*-NN.

Os exemplos *x*, *y* são conectados se $d(x, y) < \delta * \sqrt{(d(x, x_k) * d(y, y_k))}$, sendo δ um valor que pode variar continuamente. A variação desse valor permite que o parâmetro *k* seja fixado para cada conjunto de dados, de maneira similar ao *k*-NN e suas variações. Esse algoritmo tende a formar uma rede mais próxima de uma rede regular, ou seja, pode gerar vértices com um grau próximo uns dos outros em alguns casos (embora não seja uma rede regular), porém consegue capturar estruturas topológicas mais complexas, como ilustrado na Figura 11, na qual a rede resultante separa os vértices ao centro dos vértices nas bordas.

Outra desvantagem é que esse método não é flexível o suficiente para permitir que outras medidas de distâncias sejam incorporadas ao algoritmo. Por exemplo, não é possível utilizar, sem

alterações no algoritmo (ou seja, de forma direta), a distância cosseno, o que pode prejudicar seu uso em dados textuais, visto que o cosseno tem obtido bons resultados para esse tipo de dado, conforme escrito por Park et al. (2020), Liu et al. (2017) e Nguyen et al. (2011).

Figura 11 – Conjunto de dados e a rede gerada a partir dele pelo C*k*-NN.



Fonte: BERRY; SAUER, 2019.

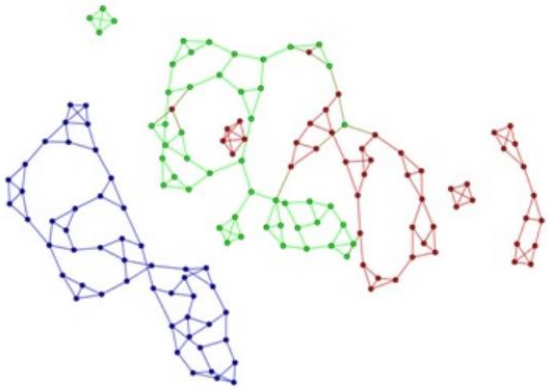
2.5.2.5 B-matching

O b-matching é uma abordagem útil para criar redes a partir de dados com alta dimensionalidade. Durante sua execução, é calculado o grau de similaridade entre os nós, usando uma função de similaridade (TALUKDAR, 2009).

A principal diferença das redes geradas por b-matching em relação às geradas pelo algoritmo *k*-NN é a otimização no sentido de garantir que cada nó tenha exatamente *b* vizinhos, resultando em uma rede exatamente regular, na qual todos os vértices têm a mesma quantidade de arestas, o que é considerado uma vantagem na propagação de rótulos em alguns cenários. Entretanto, essa característica pode ser uma desvantagem em outros cenários, por exemplo, alguns trabalhos mostram que muitos sistemas reais são representados por redes que não são regulares (NEWMAN, 2003).

A Figura 12 tem a finalidade de mostrar uma rede gerada por b-matching. No exemplo, *b* = 3, e é possível observar que cada vértice se conecta com apenas outros *b* vértices.

Figura 12 – Exemplo de uma rede *b-matching* com *b* = 3.



Fonte: BERTON, 2016.

2.5.2.6 S*k*-NN

Conforme proposto por Vega-Oliveros et al. (2014), o algoritmo Sequential *k*-NN (S*k*-NN) cria conexões entre vértices incrementalmente. Um nó é escolhido considerando um critério de relevância, como por exemplo a proximidade. Este vértice é então conectado a seu vizinho mais próximo, desde que este tenha um grau menor que *k*.

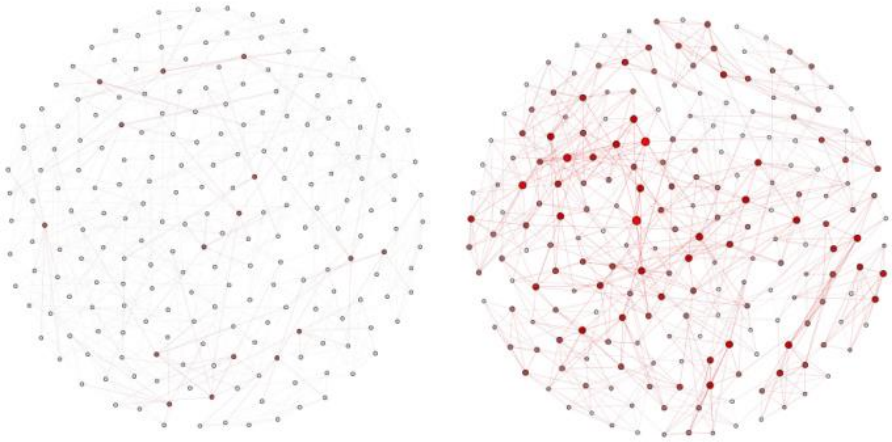
Como dito por Berton et al. (2016), o b-matching, por exemplo, tem a vantagem de assegurar que os vértices tenham exatamente *b* vizinhos e a rede seja perfeitamente regular, o que é adequado para o aprendizado semi-supervisionado em alguns contextos. No entanto, a quantidade de processamento computacional necessária para a criação de redes com o *b-matching* é bastante expressiva e o torna impraticável para grandes conjuntos de dados.

No S*k*-NN, a condição de regularidade dos vértices é relaxada. Embora o S*k*-NN não gere redes perfeitamente regulares, um de seus objetivos é eliminar a possibilidade de existência de *hubs*, que podem deteriorar a eficácia da classificação em alguns cenários. *Hubs* podem ocorrer, por exemplo, nas redes geradas pelo *k*-NN.

A Figura 13 compara as redes geradas para o mesmo conjunto de dados pelo S*k*-NN e pelo *k*-NN. Além disso, a Figura 14 mostra a distribuição de graus para os vértices das referidas redes. Nota-se que o S*k*-NN gera muitos vértices com grau similares uns aos outros; por outro lado, a rede *k*-NN possui uma distribuição próxima a uma rede livre de escala.

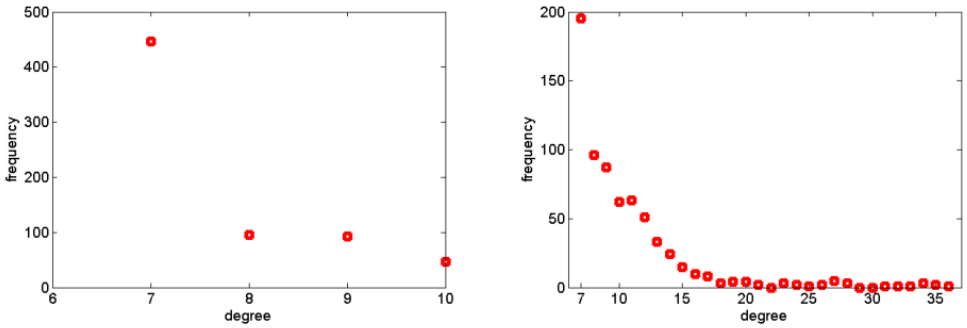
Em alguns cenários, como dito anteriormente, gerar redes altamente regulares pode ser uma desvantagem, por exemplo, em redes que conectam palavras, uma vez que documentos textuais possuem poucas palavras muito frequentes e muitas palavras pouco frequentes.

Figura 13 – Redes geradas a partir de um conjunto de dados, respectivamente pelos algoritmos S*k*-NN e *k*-NN.



Fonte: VEGA-OLIVEROS; BERTON; EBERLE; LOPES; ZHAO, 2014.

Figura 14 – Distribuição dos graus dos vértices das redes geradas a partir de um conjunto de dados, respectivamente pelos algoritmos S*k*-NN e *k*-NN.



Fonte: VEGA-OLIVEROS; BERTON; EBERLE; LOPES; ZHAO, 2014.

2.5.2.7 NNK

Para a construção de redes com o método NNK, *non negative kernel regression*, é necessário escolher os *k* maiores produtos escalares internos entre o exemplo considerado e os demais exemplos, sendo que os valores calculados representam a similaridade entre os dados.

A vantagem do NNK é que pode ser mais robusto que *k*-NN e Epsilon em alguns cenários, uma vez que o número de vizinhos conectados ao exemplo não é pré-determinado, e sim dependente da posição relativa dos vizinhos (SHEKKIZHAR; ORTEGA, 2020). A Figura 15 ilustra uma comparação entre as redes obtidas por *k*-NN e NNK para o mesmo conjunto de dados. Observa-se como o NNK, em comparação com *k*-NN, é capaz de eliminar conexões desnecessárias entre nós. Essa característica pode ser vantajosa em redes densas, porém, em redes esparsas pode ser uma desvantagem, pois a rede resultante pode ser desconexa e possuir *hubs* em excesso.

Figura 15 – Redes geradas por *k*-NN e NNK, respectivamente, para o mesmo conjunto de dados.

Fonte: HEKKIZHAR; ORTEGA, 2020.

2.5.2.8 GBLP

De acordo com Berton (2016), *graph based on link prediction* (GBLP) é um algoritmo de construção de redes que se baseia em medidas de predição de *links* e cria redes pequeno mundo, nas quais o agrupamento local é alto e a distância entre vértices é pequena. Não costuma ser muito usado como um algoritmo de construção de redes para fins de classificação, sendo mais comum em aplicações como predição de amizades futuras em redes sociais.

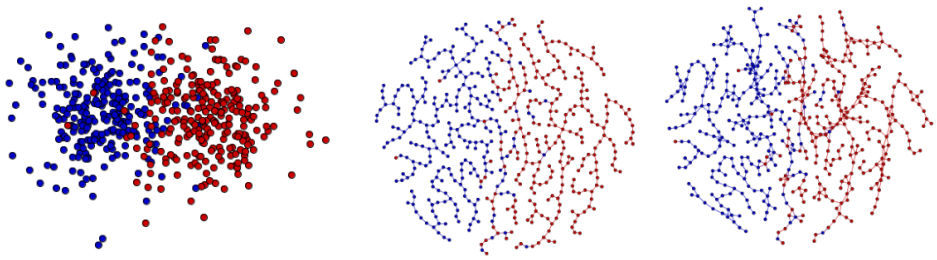
É necessário que seja utilizada uma rede inicial criada por um método mais tradicional, como o *k*-NN, que deve ser esparsa, ou seja, o *k* ser pequeno, a fim de minimizar os custos dos métodos de predição de *links*. No presente trabalho, são usadas inicialmente redes geradas por *k*-NN com valores baixos de *k*, e por MST, conforme sugerido pela autora. Para estimar novos

vértices na rede é utilizado o índice *weighted common neighbors* (WCN), que considera o número de vizinhos em comum entre um par de vértices, ponderado pelo peso das arestas, embora qualquer índice para predição de *links* possa ser utilizado nessa etapa (VALVERDE-REBAZA; VALEJO; BERTON; FALEIROS; LOPES, 2015).

Para predição dos *links* é associado a cada par de vértices desconectados uma pontuação relativa à similaridade entre eles. As pontuações são ordenadas e passa a existir uma conexão entre uma porcentagem dos vértices mais similares.

A Figura 16, retirada de Berton (2016), ilustra os passos para construção de uma rede GBLP. Na Figura 16 (a) é mostrada uma base de dados. Na Figura 16 (b) está a rede gerada por MST a partir da base de dados. Já na Figura 16 (c), foi utilizado GBLP na rede gerada por MST, com o índice para predição de *links* Katz, considerando 30% dos *links* preditos.

Figura 16 – Exemplo de construção de uma rede GBLP.



(a) (b) (c)

Fonte: BERTON, 2016.

2.5.2.9 MST

O problema da *minimum spanning tree* (MST), na tradução para o português, árvore de extensão mínima, é definido como a seguir.

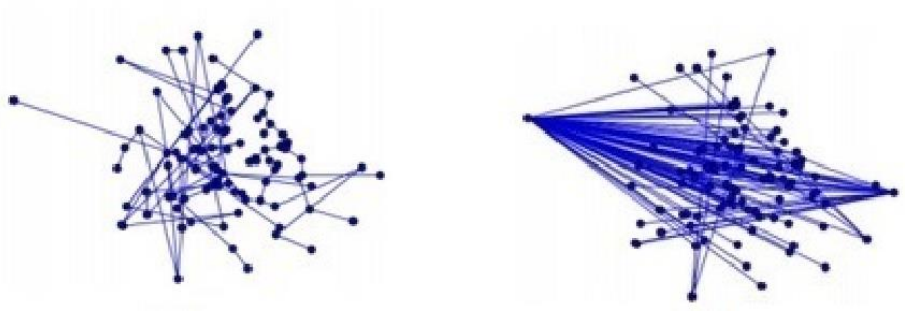
Considera-se uma rede cujos vértices representam cidades, as arestas representam possíveis caminhos de comunicação entre elas e o peso das arestas representam o custo de construção ou o tamanho do caminho. O problema trata-se de definir um conjunto de caminhos que conecte todas as cidades e tenha custos de construção mínimos, ou tamanho mínimo entre os caminhos, ou seja, encontrar uma árvore de extensão mínima (GRAHAM; HELL, 1985).

De maneira abstrata, o objetivo é definir uma rede acíclica conexa, contendo todos os vértices da rede. Uma rede acíclica trata-se de uma rede sem ciclos, sendo um ciclo um caminho cujo primeiro vértice coincide com o último. As redes geradas pelos algoritmos de MST são esparsas e podem ser usadas em conjunto com outros métodos de construção de redes. Essa técnica é comumente utilizada para deixar uma rede conexa.

Existem diversos algoritmos com o objetivo de descobrir uma rede que seja a árvore de extensão mínima para algum problema. Pode-se utilizar os algoritmos de Kruskal e de Prim para se encontrar essa rede (BERTON, 2016).

A Figura 17 ilustra um exemplo de rede *minimum spanning tree*, e também de uma rede *maximum spanning tree*, tratando-se esta do oposto da anterior, ou seja, o caminho com maior custo total.

Figura 17 – Exemplo de uma rede *minimum spanning tree* e de uma rede *maximum spanning tree*, respectivamente, para um conjunto de dados com 100 elementos.



Fonte: BERTON, 2016.

2.5.3 Aprendizado semi-supervisionado transdutivo

As duas abordagens utilizadas no aprendizado semi-supervisionado são a transdutiva e a indutiva (GUTIÉRREZ, 2010).

Na aprendizagem pela abordagem indutiva, o objetivo é que se gere um modelo que posteriormente seja capaz de rotular novos dados. Ou seja, há um foco não nos dados que foram considerados durante o treino, mas principalmente em exemplos que que estão fora do conjunto inicialmente considerado.

Já na abordagem transdutiva para o aprendizado de máquina, o algoritmo recebe de maneira simultânea exemplos de ambos os tipos de dados, rotulados e não rotulados. Nesse momento, que é a fase de aprendizagem, o algoritmo já aprende a classificar os dados sem rótulos, e não são apresentados novos dados. Não há a necessidade de generalizar o modelo para novos exemplos, pois o aprendizado transdutivo conhece previamente todos os dados a serem rotulados (BERTON, 2016).

Conforme escrito por Gutiérrez (2010), em 1996 o cientista Vladimir Vapnik definiu a teoria da inferência transdutiva, tendo como princípio o seguinte, em suas palavras: "para resolver um problema de interesse, não é necessário resolver um problema mais geral como passo intermediário". Ou seja, diferente dos modelos indutivos, que tentam calcular uma função geral que contemple todo o intervalo do universo de valores, os métodos transdutivos calculam uma função mais simples apenas para um certo intervalo de valores.

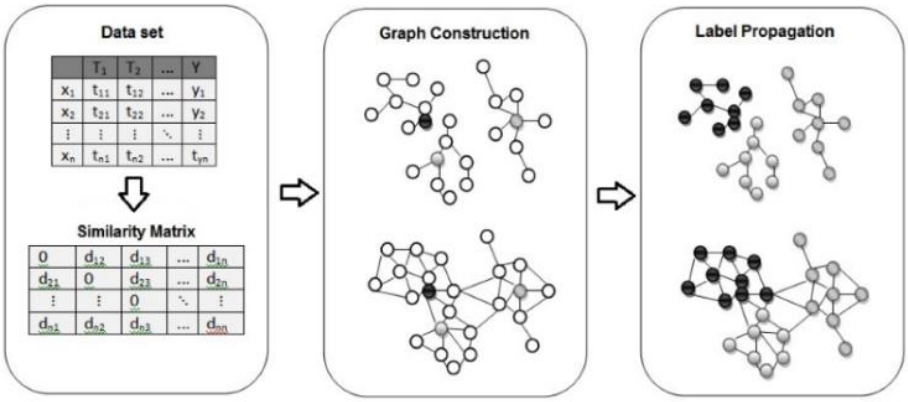
Ainda segundo Gutiérrez (2010), para definir uma estratégia, é necessário que se considere o problema a ser resolvido. Se os dados são inteiramente conhecidos previamente, pode ser que seja um caso propício para que seja aplicado um método transdutivo.

Em geral, os métodos baseados em redes, como os utilizados no presente trabalho, utilizam a abordagem de aprendizado transdutivo.

Conforme visto em Talukdar (2009), nos métodos de classificação semi-supervisionada baseados em redes, inicialmente é construída uma rede a partir do conjunto de dados considerado, e em um segundo momento são atribuídas classificações aos dados não rotulados usando algum algoritmo de classificação semi-supervisionado transdutivo de interesse. Os rótulos são inferidos por meio da estrutura topológica da rede construída e das informações dos dados rotulados, o que explica a necessidade de se usar um algoritmo de construção de redes adequado, uma vez que isso interfere diretamente na acurácia dos modelos de classificação utilizados.

Na Figura 18, retirada de Berton et al. (2018), é ilustrado o funcionamento de algoritmos de aprendizagem semi-supervisionada baseados em redes. A Figura 18 (a) ilustra a matriz de similaridade construída a partir dos dados originais, sendo essa matriz calculada considerando alguma medida de distância, por exemplo a distância euclidiana ou a distância cosseno. Na Figura 18 (b), um algoritmo é utilizado para construir a rede a partir da matriz. Por fim, na Figura 18 (c), algum algoritmo de classificação semi-supervisionada é utilizado, para propagar os rótulos para os vértices não rotulados.

Figura 18 – Funcionamento de um algoritmo de aprendizagem semi-supervisionada.



(a) (b) (c)

Fonte: BERTON; LOPES; VEGA-OLIVEROS, 2018.

A seguir, são apresentados alguns algoritmos clássicos de classificação semi-supervisionada baseados em redes, considerados no presente trabalho.

2.5.3.1 Campos gaussianos e funções harmônicas

Essa abordagem, *Gaussian Fields and Harmonic Functions* (GFHF), proposta por Zhu et al. (2003), é baseada no modelo de campos gaussianos e funções harmônicas. Nesse algoritmo é utilizado uma rede cujos vértices representam os dados, e cujos pesos das arestas contém valores que simbolizam a similaridade entre dados.

O problema de aprendizagem em questão, por sua vez, é representado em termos de um campo aleatório gaussiano sobre essa rede, ou seja, os campos gaussianos são utilizados para ponderar a similaridade entre vértices e, consequentemente, o peso das arestas.

A função harmônica determina que a informação de classe de um objeto é dada pela média das informações de classe dos vértices vizinhos, ponderada pelos pesos das arestas. A função harmônica é dada apenas a vértices não rotulados.

2.5.3.2 Caminhada Aleatória

O algoritmo Caminhada Aleatória ou *Random Walk (*RW), usa uma caminhada aleatória e se caracteriza por ser um processo estocástico utilizado como um mecanismo de transporte e pesquisa em redes. A caminha aleatória é performada conforme calcula-se uma probabilidade para que se siga do vértice atual em direção a outro vértice ou se mantenha no vértice atual, sendo que tal probabilidade é proporcional ao peso da aresta em questão. Dessa forma, vértices tendem a caminhar com maior frequência em regiões densamente conectadas, nas quais vértices podem compartilhar propriedades e desempenhar papéis semelhantes.

De forma mais geral, vértices rotulados irão propagar seus rótulos de forma probabilística aos vértices não rotulados por meio das caminhadas (ZHOU; SCHÖLKOPF, 2004).

2.5.3.3 Propagação de Rótulos

Propagação de Rótulos ou *Label Propagation* (LP) é um algoritmo popular, simples e econômico em termos de tempo de processamento, comumente usado na tarefa de detecção de comunidade e classificação de vértices.

Considerando a classificação de vértices, inicialmente vértices rotulados propagam seus rótulos para vértices vizinhos. A cada iteração, cada rótulo de vértice não rotulado é atualizado com o rótulo mais frequente em sua vizinhança. Ao final, cada vértice é atribuído ao rótulo ao qual a maioria de seus vizinhos pertence.

Intuitivamente, grupos de vértices densamente conectados convergirão para rótulos semelhantes. Finalmente, os vértices atribuídos ao mesmo rótulo serão considerados em um único grupo ou comunidade na convergência e serão classificados com o mesmo rótulo (JUNG; HERO III; MARA; JAHROMI, 2017).

2.6 Trabalhos relacionados

Nesta seção, são apresentadas pesquisas relacionadas com o presente trabalho, ou seja, trabalhos que tenham estudado algoritmos de construção de redes para o uso em algoritmos de

classificação semi-supervisionada e que tenham utilizado como entrada dados textuais, e que tenham contribuído para o avanço do estado da arte.

Berton et al. (2018), em seu artigo sobre métodos de construção de redes para o aprendizado semi-supervionado, investigam o desempenho de alguns algoritmos de construção de redes. É feita uma análise comparativa dos algoritmos b-matching, *k*-NN, S*k*-NN e M*k*-NN. No entanto, não é feita comparação a partir de classificação de dados textuais, o que é o foco do presente trabalho, mas sim em um contexto geral, ou seja, em redes de vários domínios.

Outro artigo relevante é o de Castillo et al. (2017), que revisa diferentes representações de redes para a classificação de dados textuais, em detrimento de outras representações, como vetores, porém não trata diretamente de algoritmos de construção de redes.

Já Chong et al. (2020) exploram a importância dos métodos de construção de redes. Porém, seu artigo trata de algoritmos de classificação semi-supervisionada em geral, sem um foco especificamente no aprendizado transdutivo.

Por fim, como trabalho mais próximo desta monografia, Ozaki et al. (2011) trabalharam com classificação semi-supervisionada de dados de linguagem natural. No entanto, utilizaram o algoritmo M*k*-NN em uma comparação com os tradicionais *k*-NN e o b-matching. Além disso, não consideraram uma avaliação da medida de distância cosseno, que é bem conhecida por fornecer bons resultados nesse tipo de dados. Por fim, outra lacuna é que o trabalho não faz referência como os dados textuais foram modelados, por exemplo, se foi utilizado *Bag of Words*, TF-IDF, *word embedding*, dentre outros métodos. A principal constatação desse trabalho foi que algoritmos de construção de redes que geram redes livres de escala tendem a produzir uma melhor modelagem para a classificação semi-supervisionada de textos.

Desse modo, considerando a literatura atual, fizemos uma análise comparativa entre oito algoritmos de construção de redes na classificação semi-supervisionada de textos, usando três algoritmos de classificação semi-supervisionados transdutivos. A hipótese é que algoritmos de construção de redes que utilizam a distância cosseno e constroem redes menos regulares é mais próxima das redes livres de escala são melhores para modelar dados textuais e utilizadas no problema de classificação de textos.

3. METODOLOGIA

Este capítulo descreve o processo de desenvolvimento da pesquisa do presente trabalho, apresentando os métodos, técnicas, ferramentas e conjuntos de dados utilizados para a análise dos métodos de construção de rede na classificação semi-supervisionada de textos.

3.1 Aquisição dos conjuntos de dados

Foram utilizados conjuntos de dados retirados do trabalho de Rossi et al. (2013). A seguir, há uma breve descrição de cada um dos conjuntos de dados textuais utilizados, de acordo com a documentação de Rossi et al. (2013).

3.1.1 CSTR

A coleção Computer Science Technical Reports (CSTR) é composta por resumos e relatórios técnicos publicados entre 1991 e 2007 no Departamento de Ciência da Computação da Universidade de Rochester. Os 299 documentos são divididos entre quatro classes: PLN, visão e robótica, sistemas e teoria.

3.1.2 SyskillWebert

Essa coleção é formada por 334 páginas web divididas entre quatro classes: bandas, ovelha, cabra e biomedicina.

3.1.3 Irish-Sentiment

Essa coleção é composta por 1660 artigos retirados das fontes irlandesas RTE News, The Irish Times e The Irish Independent. Voluntários classificaram os artigos como positivos, negativos ou irrelevantes, sendo essas as três classes entre as quais os documentos se dividem.

3.1.4 Oh0

Oh0 é um conjunto que contém 1003 documentos, extraídos da coleção OHSUMED, contendo publicações médicas de 1987 a 1991. Os documentos estão divididos entre dez classes cujos temas são México, ácido úrico, 6-Ketoprostaglandin F1 alpha, laringe, química cerebral, creatina quinase, ética, *fundus oculi*, Inglaterra, e prótese valvular cardíaca.

3.1.5 Tr23

Essa coleção, derivada das coleções Trec-5, Trec-6 e Trec-7, da *Text Retrieval Conference*, contém 204 documentos, divididos entre seis classes, com rótulos 280, 272, 277, 271, 274 e 273.

3.2 Escolha dos algoritmos

Para a etapa de construção das redes a partir dos dados, foram escolhidos os algoritmos tradicionais da literatura, bem como alguns mais recentes que vêm demonstrando bons resultados. São eles: *k*-NN, M*k*-NN, Epsilon, NNK, b-matching, C*k*-NN, GBLP e S*k*-NN. Eles foram implementados em Python, com exceção do b-matching, que possui uma versão implementada em C, disponível em Khan et al. (2018).

Os algoritmos *k*-NN, M*k*-NN, S*k*-NN e Epsilon foram configurados para utilizarem a distância cosseno como métrica; já os outros algoritmos utilizam suas próprias medidas de similaridade, sendo que o C*k*-NN e o b-matching utilizam variações da distância euclidiana. Os algoritmos S*k*-NN, C*k*-NN e, principalmente, o b-matching, tendem a gerar redes mais regulares, ou seja, com o grau dos vértices próximos entre si.

Já para a classificação semi-supervisionada dos dados, foram selecionadas três versões de algoritmos clássicos da literatura: GFHF (ZHU; GHAHRAMANI; LAFFERTY, 2003), RW (ZHOU; SCHÖLKOPF, 2004) e LP (JUNG; HERO III; MARA; JAHROMI, 2017), apresentados na Seção 2.5.3. Foi utilizada a biblioteca GraphLearning, disponibilizada por Flores et al. (2020). Os parâmetros dos algoritmos escolhidos foram os padrões oferecidos pela biblioteca GraphLearning.

3.3 Modelagem

Para o pré-processamento dos dados, foram utilizados os conceitos apresentados na seção 2.4 sobre mineração de textos. Ou seja, foi necessário realizar a tokenização nas frases dos textos para a obtenção dos termos, seguida pela remoção de *stopwords* para que fossem consideradas apenas palavras relevantes, o *stemming* dos termos para unificar palavras com a mesma raiz, e, por fim, a representação de valores simbólicos em valores numéricos para o uso nos algoritmos, para a qual foi utilizada a técnica *Bag Of Words*.

Em seguida, os dados obtidos foram enviados para os métodos de construção de redes. Quando necessário, para as redes desconexas, foi utilizada uma MST para tornar a rede conexa.

Posteriormente, foram utilizados os algoritmos de classificação semi-supervisionada transdutiva para classificar os textos não rotulados. Para isso, foram utilizados 30% dos dados rotulados, estando os rótulos distribuídos de maneira balanceada entre as classes. Além disso, foi feita uma randomização dos dados rotulados cinco vezes, de forma a variá-los, e foram obtidos a média e o desvio padrão dos resultados atingidos a partir deles. Após isso, foram feitos os testes com o algoritmo em uso, para a obtenção da acurácia da classificação.

Observa-se também que, para algoritmos com quantidade determinada, com limite de vizinhos ou com raio de vizinhança, o teste foi repetido 5 vezes com diferentes valores do parâmetro em questão, com o objetivo de se obter o melhor resultado possível para cada algoritmo. Portanto, foram avaliados os resultados considerando os melhores valores dos parâmetros para cada algoritmo e os resultados considerando a influência da variação dos parâmetros dos algoritmos.

3.4 Avaliação

A medida de comparação dos algoritmos foi a acurácia obtida pelos algoritmos de classificação semi-supervisionada ao rotular os dados textuais.

Inicialmente, foi implementado um algoritmo para obter histogramas com o objetivo de mostrar a frequência de aparições das palavras em cada um dos conjuntos de dados, bem como a proporção da aparição de palavras do dicionário nos documentos dos conjuntos de dados, e analisar como isso afeta as redes construídas a partir deles.

Conforme realizado por Ozaki et al. (2011), também foi obtida a quantidade proporcional de arestas conectando vértices de classes distintas, ou seja, a proporção de φ-arestas, com o objetivo de realizar uma análise do desempenho dos algoritmos no momento da criação de redes.

Foi realizada, ademais, uma análise a partir dos resultados de acurácia obtida pelas combinações de algoritmos de construção de redes e de classificação semi-supervisionada, para cada conjunto de dados textuais. Esses dados foram representados em tabelas.

Além disso, foi realizada uma análise estatística, conforme o recomendado por Demšar (2006), para a comparação de mais de dois algoritmos de construção de redes atuando no mesmo conjunto de dados. No geral, não há um procedimento estabelecido para comparar o desempenho de classificadores em vários conjuntos de dados. Existem diferentes técnicas estatísticas que podem ser adotadas e que dependem da experiência para decidir se as diferenças entre os algoritmos são reais ou aleatórias.

Demšar define algumas diretrizes para realização desses testes estatísticos, por exemplo, o teste t-pareado, o *Wincoxons's rank test*, o teste ANOVA, o *Friedman's test*, entre outros. Para usuários não experientes em testes estatísticos pode ser um grande desafio escolher a diretriz correta, uma vez que esta não é uma decisão intuitiva ou trivial.

Para realizar uma comparação estatística das medidas de desempenho obtidas pelos algoritmos nas diferentes bases de dados utilizadas deste trabalho, foi utilizada a biblioteca Autorank (HERBOLD, 2020) do Python. O objetivo do Autorank é simplificar a análise estatística para não especialistas, pois uma única chamada de função lida com a escolha das diretrizes citadas acima. Funções adicionais permitem a geração de gráficos apropriados e tabelas de resultados, bem como um documento completo em Latex. Tudo o que é necessário são os dados sobre as populações em um *dataframe* do Pandas (PANDAS, 2021). O Autorank utiliza a seguinte abordagem para realizar a comparação estatística: primeiro, usa o teste de Shapiro-Wilk para verificar se a população dos resultados segue uma distribuição normal; é utilizada a Correção de Bonferoni para esses testes. Caso os dados sigam uma distribuição normal, é usado em seguida o Teste de Bartlett para homogeneidade, caso contrário é utilizado o Teste de Levene. Por fim, baseando-se nos testes de normalidade e homogeneidade, o Autorank seleciona os testes e métodos mais apropriados para determinar os intervalos de confiança para comparação estatística.

Destaca-se que para os testes estatísticos foi utilizada a média dos resultados obtidos pelos algoritmos de classificação com cada algoritmo de construção de redes.

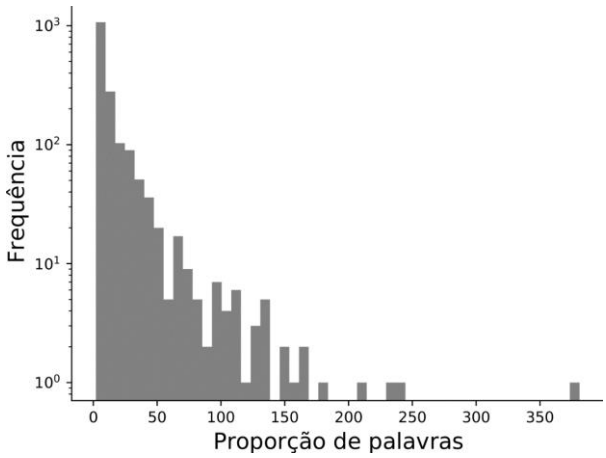
Por fim, foram plotados gráficos com o objetivo de demonstrar o desempenho dos algoritmos de construção de redes de acordo com a quantidade de rótulos utilizados na etapa de treinamento dos algoritmos de classificação.

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

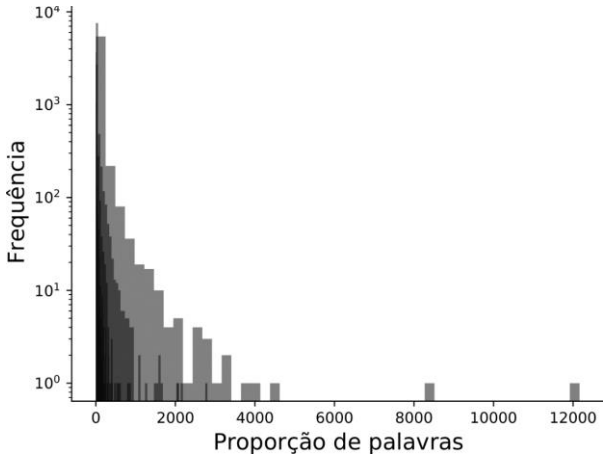
São ilustrados a seguir, na Figura 19, os histogramas demonstrando a frequência de aparição de palavras, para cada um dos conjuntos de dados, sendo (a) CTRS, (b) Irish-Sentiment,(c) SyskillWebert, (d) Oh0 e (e) Tr23, conforme é possível ver nas figuras.

Nota-se pelos gráficos que todos os conjuntos de dados refletem um padrão, no qual muitas palavras aparecem poucas vezes, e algumas poucas palavras aparecem muitas vezes.

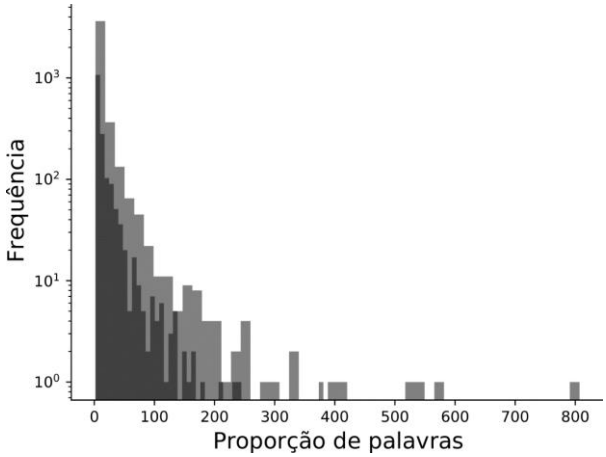
Figura 19 - Frequência de aparição de palavras nos conjuntos de dados.



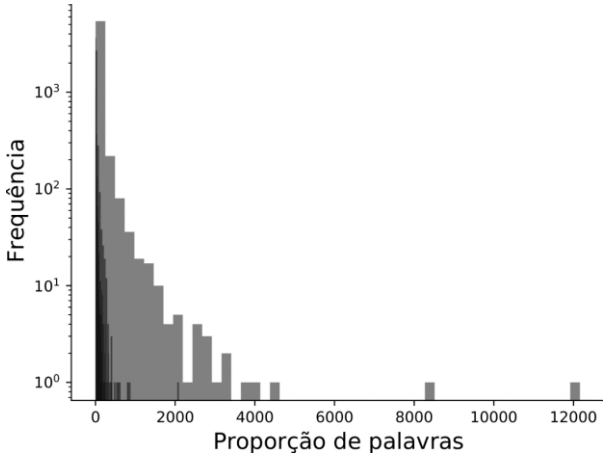
(a) CSTR



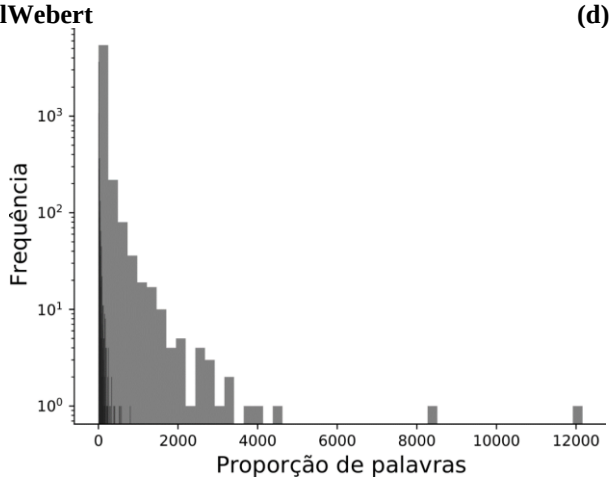
(b) Irish-Sentiment



(c) SyskillWebert



(d) Oh0



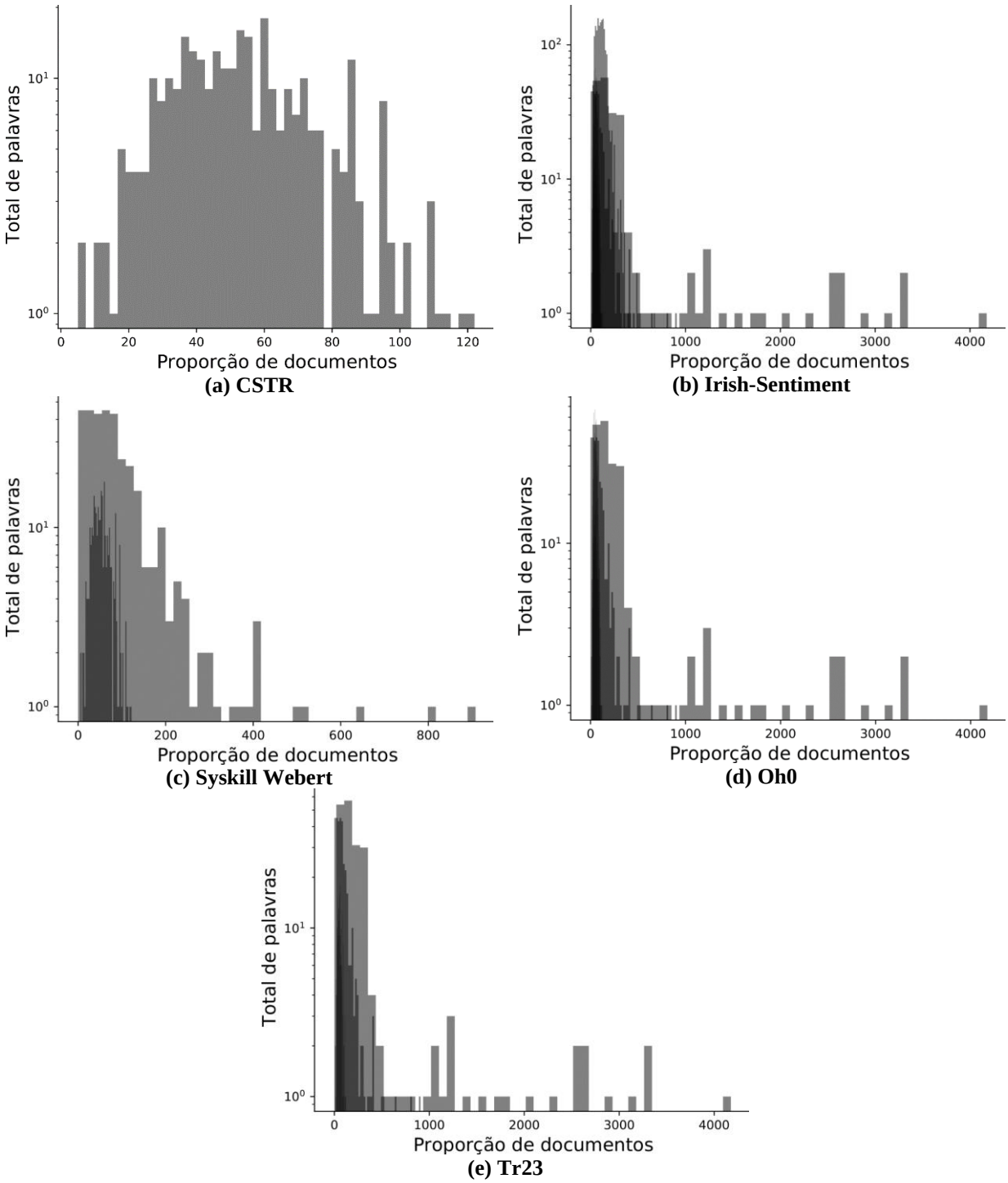
(e) Tr23

Fonte: Autora do presente trabalho.

Na Figura 20, é possível ver histogramas para cada um dos conjuntos de dados, com uma comparação do total de palavras usadas nos documentos em relação à quantidade de documentos.

Novamente, os gráficos refletem um padrão, no qual poucos documentos têm muitas palavras, e muitos documentos têm poucas palavras. Assim como nos histogramas anteriores, esse é um indício de que a tendência é que as redes construídas a partir desses dados não sejam regulares, isto é, alguns vértices devem ter muitas arestas, enquanto outros devem ter poucas, refletindo a maneira como as palavras aparecem nos documentos.

Figura 20 – Quantidade total de palavras em relação a documentos no conjunto de dados.



Fonte: Autora do presente trabalho.

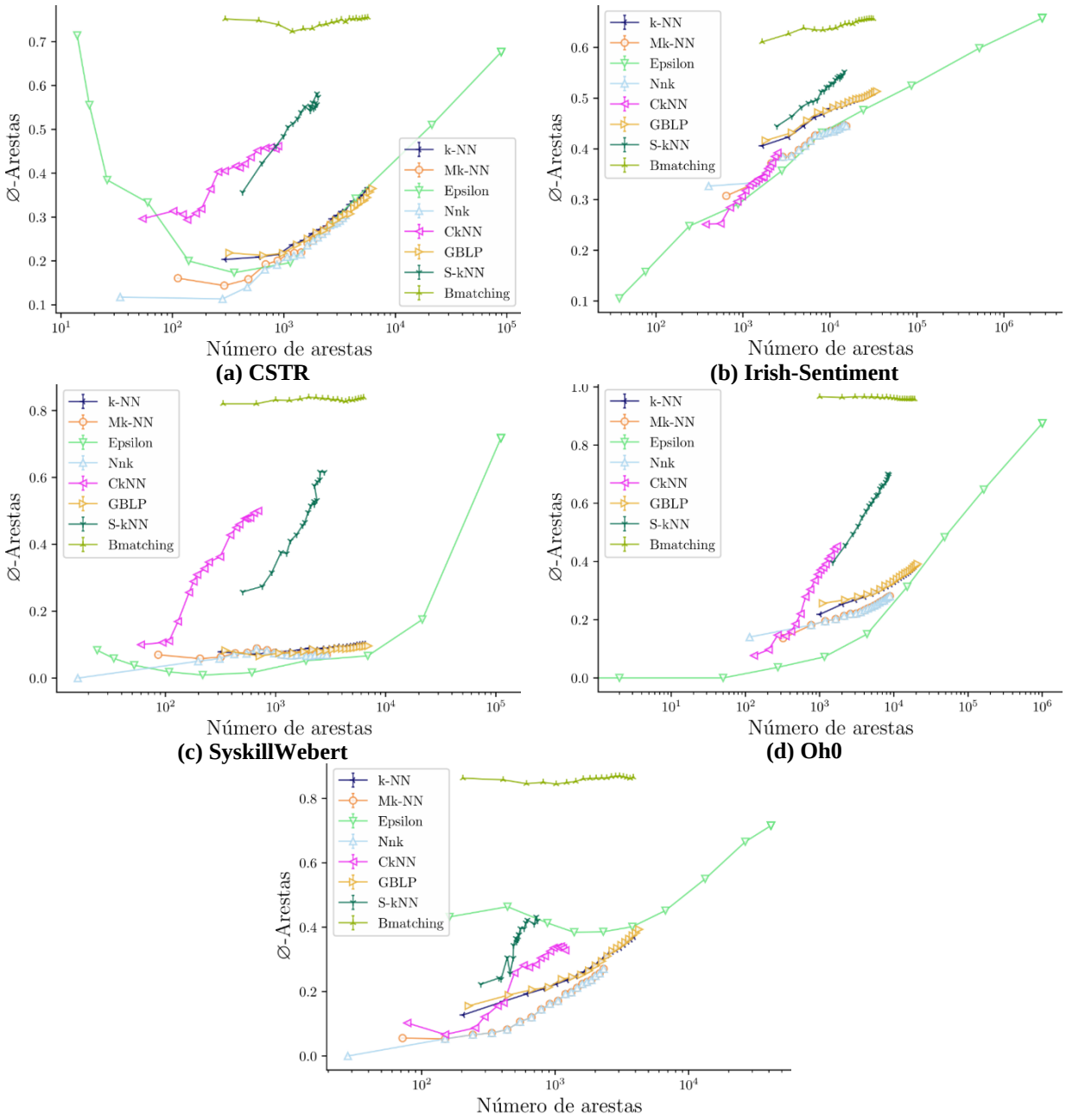
A seguir, na Figura 21, é ilustrada a proporção de φ-arestas obtida por cada um dos algoritmos de construção de redes, para cada conjunto de dados. Cada vez que os algoritmos foram

utilizados com uma variação em seu parâmetro trata-se de um ponto sobre a reta que representa o respectivo algoritmo de construção de redes. Todos os algoritmos de construção de redes avaliados possuem um parâmetro, seja ele o número de vizinhos, o tamanho da vizinhança ou o raio da vizinhança. Ao ser aumentado o parâmetro, também aumenta a quantidade de arestas na rede, ou seja, a esparsidade da rede diminui, conforme a legenda da Figura. Portanto, o eixo x dos gráficos indicam a esparsidade da rede, sendo uma forma de relacionar todos os diferentes parâmetros utilizados pelos algoritmos.

Nota-se que os algoritmos que tendem a gerar redes mais próximas de uma rede regular, como o b-matching, o C*k*-NN e o S*k*-NN, em geral têm uma proporção muito grande de φ-arestas, o que pode comprometer a acurácia da classificação dos dados. O b-matching, particularmente, obteve uma proporção superior a 0.6 em todos os casos analisados, o que se trata de um resultado ruim em comparação com os demais. Isso pode ser atribuído ao fato de que, como mencionado anteriormente, algumas palavras aparecem muitas vezes e deveriam ter muito mais conexões do que outras, e redes regulares não são capazes de expressar essa característica.

Os demais algoritmos conseguem obter resultados satisfatórios para a proporção de φ-arestas, desde que sejam definidos bons parâmetros. O M*k*-NN e o *k*-NN obtêm bons resultados para essa métrica, em geral, mesmo considerando a variação do parâmetro *k*.

Figura 21 - Proporção de φ-arestas obtida por cada algoritmo de construção de redes para cada conjunto de dados, considerando a esparsidade da rede.



(a) CSTR

(b) Irish-Sentiment

(c) SyskillWebert

(d) Oh0

(e) Tr23

Fonte: Autora do presente trabalho.

A seguir, as Tabelas 2, 3, 4, 5 e 6 apresentam, para cada conjunto de dados, os valores da acurácia obtida por cada combinação de algoritmo de construção de redes em conjunto com cada um dos algoritmos de classificação semi-supervisionada. Nessas tabelas, o algoritmo de construção de redes com melhor resultado com cada algoritmo de classificação tem seu resultado em destaque.

A partir da análise da média da acurácia obtida com o uso de cada algoritmo de construção de redes, é possível notar que o *k*-NN, que é o mais tradicional, em geral tem resultados muito bons, bem como o M*k*-NN, que é parecido em sua implementação, e o Epsilon, que também se trata de um algoritmo clássico. O GBLP e o NNK também mostraram bom desempenho, em geral. Especificamente, o *k*-NN obteve melhor resultado nas redes SyskillWebert e Oh0; o M*k*-NN obteve melhor resultado nas redes Irish-Sentiment; o NNK obteve melhor resultado nas redes Irish-Sentiment e Tr23; e o GBLP obteve melhor resultado na rede CSTR.

Por outro lado, o b-matching apresentou um desempenho muito fraco em todos os conjuntos de dados, sendo o pior em todos os cinco casos, o que pode ser explicado pelo fato de gerar redes regulares e utilizar a distância euclidiana em vez do cosseno, como dito anteriormente. Outros dois algoritmos com o desempenho inferior ao demais, em geral, foram o C*k*-NN e o S*k*-NN, que têm comportamento parecido com o b-matching no sentido de gerar redes regulares. O C*k*-NN utiliza uma variação da distância euclidiana e o S*k*-NN utiliza a distância cosseno.

Tabela 2: Acurácia obtida para o conjunto de dados CSTR.

	GFHF	RW	LP	Média
k-NN	**0.86 ± 0.01**	0.82 ± 0.02	0.70 ± 0.05	0.79
Mk-NN	0.84 ± 0.02	0.81 ± 0.01	0.70 ± 0.03	0.78
Epsilon	0.83 ± 0.02	0.79 ± 0.04	0.70 ± 0.03	0.77
NNK	0.84 ± 0.02	0.82 ± 0.01	0.72 ± 0.02	0.79
b-matching	0.56 ± 0.04	0.51 ± 0.02	0.51 ± 0.03	0.53
Ck-NN	0.69 ± 0.09	0.72 ± 0.05	0.64 ± 0.04	0.68
GBLP	0.84 ± 0.02	**0.83 ± 0.02**	**0.74 ± 0.03**	**0.80**
Sk-NN	0.70 ± 0.06	0.72 ± 0.07	0.61 ± 0.07	0.68

Fonte: Autora do presente trabalho.

Tabela 3: Acurácia obtida para o conjunto de dados Irish-Sentiment.

	GFHF	RW	LP	Média
k-NN	0.70 ± 0.01	0.69 ± 0.01	0.63 ± 0.01	0.67
Mk-NN	**0.72 ± 0.01**	**0.72 ± 0.01**	0.65 ± 0.01	**0.70**
Epsilon	0.69 ± 0.01	0.70 ± 0.02	0.64 ± 0.01	0.68
NNK	0.71 ± 0.01	**0.72 ± 0.01**	**0.66 ± 0.01**	**0.70**
b-matching	0.62 ± 0.01	0.66 ± 0.01	0.57 ± 0.00	0.62
Ck-NN	0.67 ± 0.01	0.68 ± 0.01	0.64 ± 0.01	0.66
GBLP	0.69 ± 0.01	0.69 ± 0.01	0.62 ± 0.01	0.67
Sk-NN	0.69 ± 0.01	0.70 ± 0.01	0.64 ± 0.02	0.68

Fonte: Autora do presente trabalho.

Tabela 4: Acurácia obtida para o conjunto de dados SyskillWebert.

	GFHF	RW	LP	Média
k-NN	**0.97 ± 0.01**	**0.97 ± 0.01**	**0.96 ± 0.01**	**0.97**
Mk-NN	0.92 ± 0.02	0.93 ± 0.01	0.84 ± 0.03	0.90
Epsilon	0.95 ± 0.01	0.94 ± 0.00	0.91 ± 0.03	0.93
NNK	0.92 ± 0.01	0.93 ± 0.01	0.87 ± 0.02	0.91
b-matching	0.60 ± 0.03	0.59 ± 0.01	0.51 ± 0.02	0.57
Ck-NN	0.85 ± 0.05	0.85 ± 0.03	0.71 ± 0.06	0.80
GBLP	**0.97 ± 0.01**	**0.97 ± 0.01**	0.91 ± 0.03	0.95
Sk-NN	0.85 ± 0.03	0.86 ± 0.02	0.68 ± 0.06	0.80

Fonte: Autora do presente trabalho.

Tabela 5: Acurácia obtida para o conjunto de dados Oh0.

	GFHF	RW	LP	Média
k-NN	**0.87 ± 0.01**	**0.85 ± 0.02**	0.62 ± 0.06	**0.78**
Mk-NN	0.83 ± 0.01	0.82 ± 0.01	0.62 ± 0.02	0.76
Epsilon	0.77 ± 0.01	0.76 ± 0.01	0.58 ± 0.02	0.70
NNK	0.83 ± 0.01	0.82 ± 0.01	0.62 ± 0.02	0.76
b-matching	0.46 ± 0.01	0.46 ± 0.01	0.41 ± 0.01	0.44
Ck-NN	0.66 ± 0.02	0.67 ± 0.02	0.55 ± 0.03	0.63
GBLP	0.84 ± 0.01	0.83 ± 0.01	**0.64 ± 0.02**	0.77
Sk-NN	0.72 ± 0.02	0.72 ± 0.02	0.55 ± 0.04	0.66

Fonte: Autora do presente trabalho.

Tabela 6: Acurácia obtida para o conjunto de dados Tr23.

	GFHF	RW	LP	Média
k-NN	**0.76 ± 0.05**	0.64 ± 0.07	0.52 ± 0.02	0.64
Mk-NN	0.73 ± 0.04	0.69 ± 0.04	0.57 ± 0.03	0.66
Epsilon	0.69 ± 0.06	0.66 ± 0.06	0.48 ± 0.06	0.61
NNK	**0.76 ± 0.04**	**0.71 ± 0.02**	**0.58 ± 0.03**	**0.68**
b-matching	0.42 ± 0.04	0.43 ± 0.05	0.39 ± 0.02	0.41
Ck-NN	0.66 ± 0.07	0.64 ± 0.04	0.53 ± 0.04	0.61
GBLP	0.73 ± 0.11	0.63 ± 0.05	0.52 ± 0.05	0.63
Sk-NN	0.65 ± 0.06	0.60 ± 0.04	0.47 ± 0.02	0.57

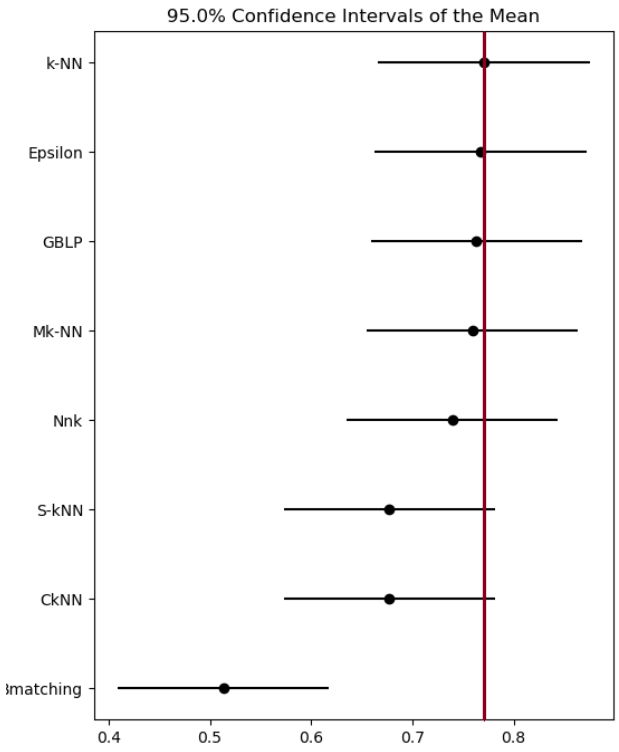
Fonte: Autora do presente trabalho.

A seguir, a Figura 22 representa a saída gerada ao ser rodada a análise estatística pela biblioteca Autorank nos resultados obtidos, utilizando as médias apresentadas na última coluna das Tabelas 2, 3, 4, 5 e 6. Esse teste foi utilizado para detectar as diferenças estatísticas no desempenho dos algoritmos. O Autorank realizou a comparação estatística e desenhou o diagrama Confidence Interval Plot (CI Plot). Para avaliar quais os grupos de algoritmos semelhantes estatisticamente, basta selecionar um algoritmo como ponto de referência e traçar uma reta vertical, todos os algoritmos cortados por essa reta fazem parte de um conjunto sem diferença estatística,

considerando o algoritmo de referência. O traço vermelho na figura demonstra esse processo. Os algoritmos posicionados mais acima possuem os melhores desempenhos.

Nota-se que, como já observado, o b-matching teve o pior desempenho dentre todos os algoritmos, seguido pelos C*k*-NN e S*k*-NN. Os algoritmos com melhor desempenho foram *k*-NN, Epsilon, GBLP e M*k*-NN, nessa ordem, sendo que o *k*-NN e o Epsilon tratam-se de alguns dos algoritmos mais tradicionais para a aplicação considerada. Nota-se que o algoritmo *k*-NN se destacou na base de dados SyskillWebert, tendo uma diferença de acurácia significativa para os outros algoritmos nessa base, o que pode ter influenciado a melhor posição do algoritmo nesse ranking.

Figura 22 – Comparação estatística dos resultados variando número de amostras geradas pelo Autorank com traço exemplificando grupo de algoritmos sem diferença estatística.

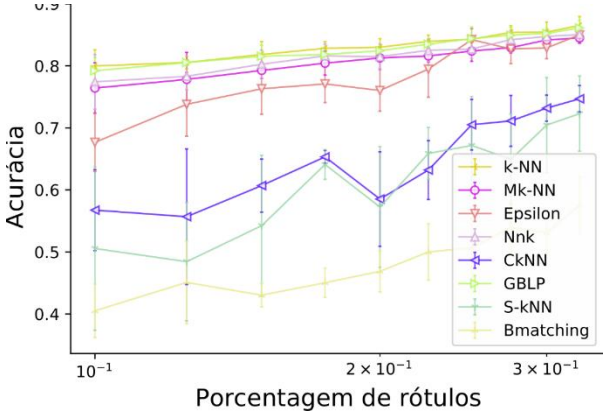


Fonte: Autora do presente trabalho.

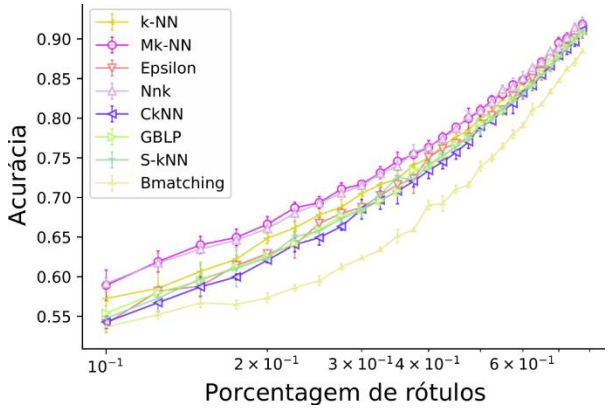
Por fim, a Figura 23 mostra uma comparação entre a porcentagem de dados rotulados utilizados na classificação e a acurácia obtida, para cada conjunto de dados.

O resultado visto nos gráficos, que mostra uma relação diretamente proporcional entre as duas variáveis, já era esperado, uma vez que quanto mais dados rotulados o algoritmo recebe, ele tende a conseguir deduzir os rótulos dos demais dados de maneira mais correta. Isso se verifica para todos os conjuntos de dados, com algumas variações pontuais.

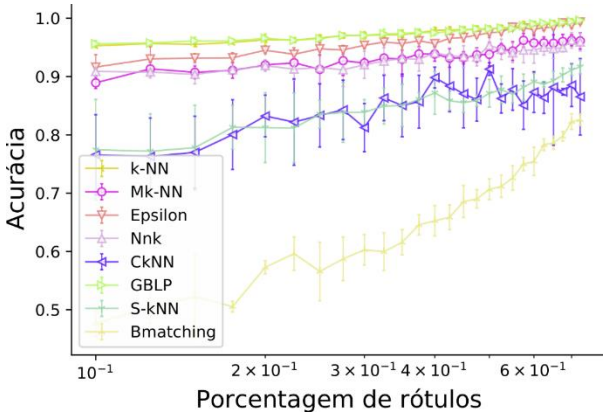
Figura 23 - Comparação entre a porcentagem de dados rotulados utilizados no treino e a acurácia obtida na classificação para os conjuntos de dados.



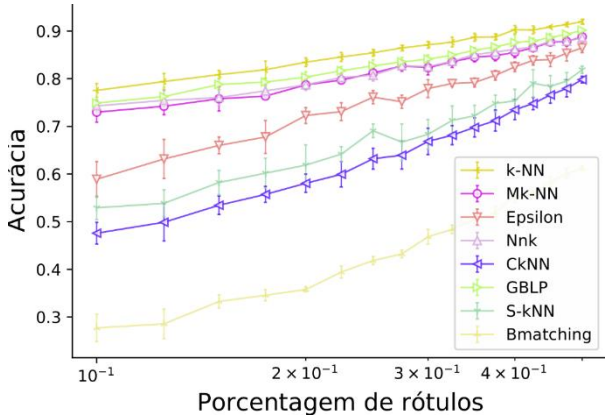
(a) CSTR



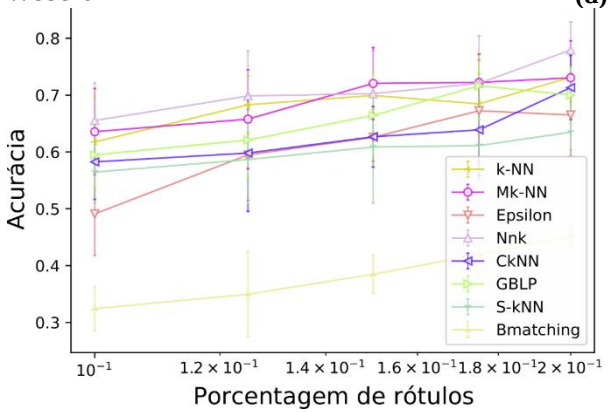
(b) Irish-Sentiment



(c) SyskillWebert



(d) Oh0



(e) Tr23

Fonte: Autora do presente trabalho.

5 CONCLUSÃO

Este trabalho compara os métodos de construção de rede na classificação semi-supervisionada de textos.

Tendo analisado o desempenho de oito algoritmos de construção de redes na análise de dados textuais, verificou-se que os que tiveram melhor desempenho foram o *k*-NN, o Epsilon, o GBLP e o M*k*-NN, com base na interpretação da análise dos dados obtidos a respeito não só da acurácia, mas também da proporção de φ-arestas, ou seja, de arestas que conectam dados de classes diferentes, e da análise estatística realizada.

Concluiu-se também que o formato dos conjuntos de dados, nos quais existem uma quantidade grande de palavras que aparecem poucas vezes e uma quantidade pequena de palavras que aparecem muitas vezes, bem como poucos documentos que apresentam muitas palavras e muitos documentos que apresentam poucas palavras, sugere que os algoritmos que geram redes regulares não têm bom desempenho ao serem utilizados com dados textuais, pois a topologia dessas redes não reflete os dados analisados, o que é reforçado pelo desempenho fraco que o algoritmo b-matching obteve.

Além disso, os algoritmos que usam a métrica cosseno também se sobressaíram, o que já era esperado, uma vez que há evidências de que essa métrica seja a melhor para dados textuais. Nota-se que o algoritmo S*k*-NN, embora use essa métrica, não gerou resultados muito bons. Os experimentos dão indícios de que isso se deva ao fato de que gera redes regulares. O NNK, que não usa métrica alguma de distância e sim uma função de distribuição, também não apresenta diferença estatística com os algoritmos que performaram melhor.

É válido ressaltar novamente que o desempenho observado em algumas situações pode ter sido influenciado pela métrica de distância. Por exemplo, o algoritmo b-matching, além de ter utilizado a distância euclidiana, também gera redes regulares. É possível que adaptar o algoritmo para utilizar a mesma métrica de distância adotada pelos algoritmos que obtiveram o melhor desempenho possa mitigar essa variação. Esse mesmo problema é observado no algoritmo C*k*-NN.

Portanto, em relação ao b-matching, devido a seu desempenho ter sido tão inferior em comparação com os outros algoritmos, verifica-se a necessidade de avaliá-lo novamente, utilizando a medida cosseno e otimizando a escolha da constante *b*, que indica o grau da rede. Além disso,

outra possibilidade seria usá-lo em conjunto com outro algoritmo de construção de redes, como o GBPL, pois pode ser interessante utilizar uma rede regular na predição de *links*.

Menciona-se também que a principal dificuldade no projeto foi encontrar os algoritmos de construção de redes implementados.

5.1 Trabalhos futuros

Para a continuação do trabalho realizado, existem algumas possibilidades, visando buscar resultados ainda melhores na comparação dos algoritmos de construção de redes.

Visto que estamos lidando com uma tarefa de aprendizado semi-supervisionado, uma possibilidade a ser considerada é usar os rótulos já na etapa de construção da rede, uma vez que temos alguns exemplos já rotulados inicialmente. Alguns algoritmos semi-supervisionados de construção de redes são: *Robust Graph that Considers Labeled Instances* (RGCLI) (BERTON; FALEIROS; VALEJO; VALVERDE-REBAZA, 2017) e *Semi-supervised K-associated graph* (BERTINI JR; LOPES; ZHAO, 2012).

Outra possibilidade seria considerar também algoritmos de classificação desenvolvidos para serem utilizados com redes bipartidas. Nesses algoritmos, as redes são carregadas diretamente a partir da matriz atributo-valor, não sendo necessário que haja de fato a construção da rede. Por exemplo, o algoritmo *Transductive Classification Using Propagation in Bipartite Graphs* (TPBG) é um algoritmo de classificação transdutivo que modela o conjunto de textos a partir de uma rede bipartida, de acordo com Faleiros et al. (2017).

Uma vez que no trabalho foi utilizado apenas a medida de distância cosseno em alguns algoritmos e outras distâncias em outros algoritmos, seria possível variar esse parâmetro nos experimentos, utilizando outras medidas, como euclidiana e *manhattan* para todos os algoritmos, verificando qual obtém melhores resultados.

Uma outra sugestão futura é analisar a distribuição de graus das redes construídas pelos algoritmos de construção de redes.

Por fim, seria interessante realizar o desenvolvimento de um *framework* para construção de redes (incluindo algoritmos não supervisionados e semi-supervisionados) com a linguagem de programação Python, pois verificou-se uma dificuldade considerável para encontrar esses

algoritmos implementados em Python. Apenas os algoritmos mais tradicionais, *k*-NN e Epsilon, estão disponíveis na biblioteca Scikit-Learn (SCIKIT-LEARN, 2021).

REFERÊNCIAS

ROSSI, Rafael G.; LOPES, Alneu de A.; REZENDE, Solange O. Optimization and label propagation in bipartite heterogeneous networks to improve transductive classification of texts. **Information Processing & Management**, v. 52, n. 2, p. 217-257, mar. 2016.

ERTEL, Wolfgang. **Introduction to Artificial Intelligence**. London: Springer, 2011.

RUSSELL, Stuart J.; NORVIG, Peter. **Artificial Intelligence**: A Modern Approach. Englewood Cliffs: Prentice Hall, 1995.

BURKOV, Andriy. **The Hundred-Page Machine Learning Book**. Quebec: [s.n.], 2019.

HONDA, Hugo; FACURE, Matheus; YAOHAO, Peng. Os Três Tipos de Aprendizado de Máquina. **Laboratório de Aprendizado de Máquina em Finanças e Organizações da Universidade de Brasília**, Brasília, 27 jul. 2017. Disponível em: https://lamfo-unb.github.io/2017/07/27/tres-tipos-am/. Acesso em: 26 set. 2021.

COMMON ML Problems. **Google Developers**, 2021. Disponível em: https://developers.google.com/machine-learning/problem-framing/cases. Acesso em: 31 out. 2021.

SANCHES, Marcelo Kaminski. **Aprendizado de máquina semi-supervisionado**: proposta de um algoritmo para rotular exemplos a partir de poucos exemplos rotulados. Dissertação (Mestrado em Ciências da Computação e Matemática Computacional) - Universidade de São Paulo, São Carlos, 2003.

PEREIRA, Silvio do Lago. Processamento de Linguagem Natural. **Instituto de Matemática e Estatística da Universidade de São Paulo**, São Paulo, 2011. Disponível em: https://www.ime.usp.br/~slago/IA-pln.pdf. Acesso em: 26 set. 2021.

JACKSON, Peter; MOULINIER, Isabelle. **Natural Language Processing for Online Applications**: Text Retrieval, Extraction and Categorization. Amsterdam / Philadelphia: John Benjamins Publishing Company, 2002.

VIJAYARANI, S.; ILAMATHI, J., NITHYA, S. Preprocessing Techniques for Text Mining: An Overview. **International Journal of Computer Science and Communication Networks**, vol. 5(1), p. 7-16, feb. 2015.

ALVARES, Reinaldo Viana. **Algoritmos de Stemming e o Estudo de Proteomas**. 2014. Tese (Doutorado em Engenharia de Sistemas e Computação) - Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2014.

KIBBLE, Rodger. **Introduction to natural language processing**. London: Goldsmiths University of London, 2013.

DAS NEVES, Rita de C. D. **Pré-Processamento no Processo de Descoberta de Conhecimento em Banco de Dados**. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2003.

CHIANG, David. Bags of Words. **University of Notre Dame**, Notre Dame, 22 oct. 2018. Disponível em: https://www3.nd.edu/~dchiang/teaching/nlp/2018/notes/chapter12v1.pdf. Acesso em: 11 out. 2021.

RAMOS, Juan. Using TF-IDF to Determine Word Relevance in Document Queries. In: **Proceedings of the First Instructional Conference on Machine Learning**, dec. 2003.

TRIPATHI, Mayank. How to process textual data using TF-IDF in Python. **freeCodeCamp**, 6 jun. 2018. Disponível em: https://www.freecodecamp.org/news/how-to-process-textual-data-using-tf-idf-in-python-cd2bbc0a94a3/. Acesso em: 26 set. 2021.

WANG, Bin; WANG, Angela; CHEN, Fenxiao; WANG, Yuncheng; KUO, C.-C. J. Evaluating Word Embedding Models: Methods and Experimental Results. **APSIPA Transactions on Signal and Information Processing**, 8, jan. 2019.

BENGIO, Yoshua; DUCHARME, Réjean; VINCENT, Pascal; JAUVIN, Christian. A Neural Probabilistic Language Model. **Journal of Machine Learning Research**, 3, p. 1137-1155, feb. 2003.

DE CARVALHO, Matheus H. **Estudo Comparativo dos Métodos de Word Embedding na Análise de Sentimentos**. Monografia (Bacharelado em Ciência de Computação) – Universidade Federal de Pernambuco, Recife, 2018.

LYNN, Shane. An introduction to word embeddings for text analysis. **Shane Lynn: Data science, startups, analytics, and data visualisation**, jan. 2018. Disponível em: https://www.shanelynn.ie/get-busy-with-word-embeddings-introduction/. Acesso em: 24 out. 2021.

BERTON, Lilian. **Construção de redes baseadas em vizinhança para o aprendizado semi-supervisionado**. Tese (Doutorado em Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo, São Carlos, 2016.

VALEJO, Alan D. B. **Refinamento multinível em redes complexas baseado em similaridade de vizinhança**. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo, São Carlos, 2014.

MAIER, Markus; HEIN, Matthias; LUXBURG, Ulrike. Cluster Identification in Nearest-Neighbor Graphs. **Proceedings of the 18th International Conference on Algorithmic Learning Theory**, 2007.

TALUKDAR, Partha P. Topics in Graph Construction for Semi-Supervised Learning. Technical Report, **University of Pennsylvania**, 2009.

BRITO, M. R.; CHÁVEZ, E. L.; QUIROZ, A. J.; YUKICH, J. E. Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection. **Statistics & Probability Letters**, v. 35, i. 1, p. 33-42, 1997.

BERRY, Tyrus; SAUER, Timothy. Consistent manifold representation for topological data analysis. **Foundations of Data Science**, 1(1), p. 1-38, 2019.

PARK, Kwangil; HONG, June; KIM, Wooju. A methodology combining cosine similarity with classifier for text classification. **Applied Artificial Intelligence**, 34, p. 396-411, 2020.

LIU, Chuan; WANG, Wenyong; TU, Guanghui; XIANG, Yu; WANG, Siyang; LV, Fengmao. A new centroid-based classification model for text categorization. **Knowledge-Based Systems**, 136, p. 15-26, 2017.

NGUYEN, Tam T.; CHANG, Kuiyu; HUI, Siu C. Word cloud model for text categorization. In: **Proceedings of the 11th IEEE International Conference on Data Mining**, Vancouver, Canada, 2011.

NEWMAN, M. E. The structure and function of complex networks. **SIAM Review**, 45(2), p. 167-256, 2003.

VEGA-OLIVEROS, Didier A.; BERTON, Lilian; EBERLE, Andre M.; LOPES, Alneu de A.; ZHAO, Liang. Regular graph construction for semi-supervised learning. **Journal of Physics: Conference Series**, 490(1), 2014.

BERTON, Lilian; LOPES, Alneu de A. **Network construction and applications for semi-supervised learning**. Porto Alegre: SBC, 2016.

SHEKKIZHAR, S.; ORTEGA, A. Graph Construction from Data by Non-Negative Kernel Regression. In: **2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**, p. 3892-3896, 2020.

VALVERDE-REBAZA, Jorge; VALEJO, Alan; BERTON, Lilian; FALEIROS, Tiago; LOPES, Alneu de A. A naïve bayes model based on overlapping groups for link prediction in online social networks. In: **Proceedings of the 30th Annual ACM Symposium on Applied Computing**, p. 1136-1141, 2015.

GRAHAM, R. L.; HELL, P. On the History of the Minimum Spanning Tree Problem. **Annals of the History of Computing**, v. 7, n. 1, p. 43-57, jan. 1985.

GUTIÉRREZ, Víctor A. L. **Classificação Semi-Supervisionada Baseada em Desacordo por Similaridade**. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo, São Carlos, 2010.

BERTON, Lilian; LOPES, Alneu de A.; VEGA-OLIVEROS, Didier A. A Comparison of Graph Construction Methods for Semi-Supervised Learning. **2018 International Joint Conference on Neural Networks (IJCNN)**, p. 1-8, 2018.

ZHU, Xiaojin; GHAHRAMANI, Zoubin; LAFFERTY, John. Semi-supervised learning using gaussian fields and harmonic functions. **Proceedings of the 20th International conference on Machine learning**, 03, p. 912-919, 2003.

ZHOU, Dengyong; SCHÖLKOPF, Bernhard. Learning from labeled and unlabeled data using random walks. **Joint Pattern Recognition Symposium**, Springer, Berlin, Heidelberg, p. 237-244, 2004.

JUNG, Alexander; HERO III, Alfred O.; MARA, Alexandru; JAHROMI, Saeed. Semi-supervised learning via sparse label propagation. **Journal of Machine Learning Research**, 1, 2017.

CASTILLO, Esteban; CERVANTES, Ofelia; VILARIÑO, Darnes. Text Analysis Using Different Graph-Based Representations. **Computación y Sistemas**, Ciudad de México, v. 21, n. 4, 2017.

CHONG, Yanwen; DING, Yun; YAN, Qing; PAN, Shaoming. Graph-based semi-supervised learning: A review. **Neurocomputing**, 408, p. 216-230, 2020.

OZAKI, Kohei; SHIMBO, Masashi; KOMACHI, Mamoru; MATSUMOTO, Yuji. Using the Mutual k-Nearest Neighbor Graphs for Semi-supervised Classification of Natural Language Data. In: **Proceedings of the Fifteenth Conference on Computational Natural Language Learning**, p. 154-162, 2011.

ROSSI, Rafael G.; MARCACINI, Ricardo M.; REZENDE, Solange O. Benchmarking text collections for classification and clustering tasks. Technical report. **Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo**, 2013.

KHAN, Arif; POTHEN, Alex; HALAPPANAVAR, Mahantesh. B-matching. **GitHub**, 2018. Disponível em: https://github.com/Exa-Graph/bMatching. Acesso em: 6 nov. 2021.

FLORES, Mauricio R.; COOK, Brendan; JACOBS, Matt; ETTEHAD, Mahmood. Graph-based Clustering and Semi-Supervised Learning. **GitHub**, 2020. Disponível em: https://github.com/jwcalder/GraphLearning. Acesso em: 5 nov. 2021.

DEMŠAR, Janez. Statistical Comparisons of Classifiers over Multiple Data Sets. **Journal of Machine Learning Research**, 7, p. 1-30, 2006.

HERBOLD, Steffen. Autorank: A Python package for automated ranking of classifiers. **Journal of Open Source Software**, 5(48), 2173, 2020.

PANDAS. **Pandas**: Python Data Analysis Library, 2021. Disponível em: https://pandas.pydata.org. Acesso em: 24 nov. 2021.

BERTON, Lilian; FALEIROS, Thiago de P.; VALEJO, Alan; VALVERDE-REBAZA, Jorge; LOPES; Alneu de A. RGCLI: Robust Graph that Considers Labeled Instances for Semi-Supervised Learning, **Neurocomputing**, v. 226, p. 238-248, 2017.

BERTINI JR, João R.; LOPES, Alneu de A.; ZHAO, Liang. Partially labeled data stream classification with the semi-supervised K-associated graph. **Journal of the Brazilian Computer Society**, 18, p. 299-310, 2012.

FALEIROS, Thiago; ROSSI, Rafael G.; LOPES, Alneu de A. Optimizing the class information divergence for transductive classification of texts using propagation in bipartite graphs. **Pattern Recognition Letters**, v. 87, p. 127-138, feb. 2017.

SCIKIT-LEARN. **Scikit-Learn**: Machine Learning in Python, 2021. Disponível em: https://scikit-learn.org/stable/. Acesso em: 10 nov. 2021.