Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Algoritmo de Fatoração de Matrizes Não-negativas para Aprendizado positivo não-supervisionado

Lucas Souza Sampaio Nunes

Dissertação apresentada como requisito parcial para
conclusão do Mestrado em Informática

Orientador
Prof. Dr. Thiago de Paulo Faleiros

Brasília
2024

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

Sa

Souza Sampaio Nunes, Lucas
Algoritmo de Fatoração de Matrizes Não-negativas para Aprendizado positivo não-supervisionado / Lucas Souza Sampaio Nunes; orientador Thiago de Paulo Faleiros. -- Brasília, 2024.
89 p.

Tese(Mestrado em Informática) -- Universidade de Brasília, 2024.

1. classificação de textos.. 2. aprendizado não-supervisionado. 3. positive unlabeled learning. 4. non-negative matrix factorization. 5. deep non-negative matrix factorization. I. de Paulo Faleiros, Thiago, orient. II. Título.

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Algoritmo de Fatoração de Matrizes Não-negativas para Aprendizado positivo não-supervisionado

Lucas Souza Sampaio Nunes

Dissertação apresentada como requisito parcial para
conclusão do Mestrado em Informática

Prof. Dr. Thiago de Paulo Faleiros (Orientador)
UnB

Prof. Dr. Li Weigang
UnB

Prof. Dr. Wallace Anacleto Pinheiro
UFRJ

Prof. Dr. Rodrigo Bonifácio Almeida
Coordenador do Programa de Pós-graduação em Informática

Brasília, 02 de setembro de 2024

Dedicatória

Dedico esta dissertação a Deus por me permitir ter saúde e condições de realizar esse curso de mestrado. Também agradeço ao meu pai Antônio Elder e à minha mãe Olguimeide, principais responsáveis por tudo que eu tenho e conquistei, por sempre incentivarem meus estudos e sempre estarem presentes. E ainda, à minha esposa, Roberta, por sempre me apoiar nas minhas iniciativas e entender todos os momentos em que estive ausente devido aos estudos decorrentes das disciplinas e da pesquisa relacionados ao curso de mestrado.

Agradecimentos

A Deus, por tudo o que me proporciona na minha vida.

À minha amada esposa Roberta, por todo apoio, suporte, motivação e compreensão durante estes anos de estudo e por entender o quanto este mestrado era importante para mim e para a nossa família e também por compreender os momentos de ausência devido aos estudos.

Ao meu orientador Prof. Thiago de Paulo Faleiros, por se mostrar sempre disponível para ajudar, seja nas questões técnicas do trabalho desenvolvido, seja nas questões formais e administrativas do mestrado, e também por sempre guiar o trabalho para um caminho relevante.

Aos meus amigos e colegas de trabalho Moreira e Beatriz, pelas dicas, sugestões, críticas e materiais, além de todo incentivo concedido.

Aos colegas de trabalho e principalmente aos meus superiores que sempre se mostraram compreensíveis dos afazeres do mestrado, me concedendo flexibilidade para lidar com a carga horária do curso.

Aos professores Li e Wallace, pelas considerações e discussões durante a qualificação e durante a defesa, que acrescentaram muito ao conteúdo e à forma do trabalho.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

“Roma ne fu pas faite toute en un jour.”

Resumo

A rotulagem de dados para treinamento de modelos de aprendizado de máquina está se tornando cada vez mais inviável devido ao alto volume de dados disponíveis e continuamente sendo produzidos. Portanto, pesquisas atuais se concentram na análise e investigação de técnicas de resolução do problema de *Positive Unlabeled Learning* (PUL), que podem produzir um desempenho satisfatório de classificação, mesmo com uma pequena porção de dados rotulados. Neste trabalho, é proposta uma adaptação estrutural do algoritmo de *Non-negative Matrix Factorization* (NMF), aplicada a problemas de PUL e denominada NMFPUL, a fim de aprimorar o desempenho da classificação de dados textuais. O NMF é uma técnica usada para a fatoração de matrizes e geralmente é utilizada para redução de dimensionalidade. Além disso, foi investigada uma variação do algoritmo NMF no aprendizado profundo, o *Deep Non-Negative Matrix Factorization* ou Deep NMF. Esta pesquisa aplica o algoritmo proposto em vários conjuntos de dados textuais, contendo milhares de documentos e termos, considerando diferentes quantidades de dados rotulados, variando de 1 a 30 documentos rotulados na classe positiva. Para os conjuntos de dados menores, o algoritmo proposto apresentou desempenho de classificação próximo às outras técnicas de ponta, enquanto, nos conjuntos de dados maiores, o desempenho do NMFPUL se destacou, obtendo uma melhoria de 10% a 30% em relação às outras técnicas, sendo a maior diferença observada quando há uma menor quantidade de documentos rotulados. O uso do NMF envolve a aplicação de uma função objetivo para convergir a matriz documento-palavra ao produto das matrizes documento-tópico e tópico-palavra. Essas técnicas de convergência podem ser utilizadas em métodos de aprendizado profundo, desdobrando as iterações do algoritmo em camadas da rede.

Palavras-chave: classificação de textos, aprendizado não-supervisionado, positive unlabeled learning, non-negative matrix factorization, deep non-negative matrix factorization

Abstract

The data labeling for machine learning models training is more and more impracticable, in a manual way, due to the high volume of data available and that is continuously produced. So, the current research stick to the analysis and investigation of *Positive Unlabeled Learning* (PUL) problem solving techniques, which can produces satisfactory classification performance, even having a small portion of data labeled. In this work, a structural adaptation to the *Non-Negative Matrix Factorization* (NMF) algorithm applied to PUL, denominated NMFPUL, is proposed in order to enhance the performance of text data classification. NMF is a technique used for matrix factorization and usually used to reduce dimensionality. This research applies the algorithm proposed in several text datasets, containing thousands of documents and terms, considering different amount of labeled data, varying from 1 to 30 labeled documents on the positive class. For the smallest datasets, the proposed algorithm had performance of classification close to those other state-of-the-art techniques, while, on larger datasets, the performance of NMFPUL stood out, having a 10% to 30% over other techniques, having the biggest difference when there are less quantity of labeled documents. The use of NMF involves applying a objective function to converge the matrix document-term to the product of document-topic and topic-term matrices. Those convergence techniques could be used in deep learning methods, unrolling the algorithm iterations into layers of the network. So, also, in this work, a variation of NMF for deep learning, the *Deep Non-Negative Matrix Factorization* or Deep NMF, is developed and applied to PU data, to compare with others state-of-the-art techniques in order to identify improvements to the performance of textual data classification.

Keywords: text classification, unsupervised learning, positive unlabeled learning, non-negative matrix factorization, deep non-negative matrix factorization

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

1.1 Contextualização

A incorporação de novas tecnologias no cotidiano das pessoas e instituições levou a um aumento exponencial do volume de dados gerados. Esse fato se intensificou biênio 2020/2021 devido à pandemia do COVID-19, pois uma grande porção das atividades humanas cotidianas de forma presencial foram substituídas por interações virtuais. Pode-se agregar a esse fator, o crescimento mundial da população de usuários com acesso a internet, alcançando mais de 63% da população mundial (Josh James, 2022). Indo mais fundo no contexto desse aumento, encontram-se parâmetros de complexidade como a existência de dados estruturados, semi-estruturados, não estruturados; conjuntamente às questões de organização, indexação, procura, análise, extração de informação desses dados (Naeem et al., 2021; Jagadish et al., 2014). Em complemento a isso, outras áreas atinentes à análise de dados crescem paralelamente como as estruturas para armazenamento de dados, o processamento de dados, o desenvolvimento de equipamentos de hardware capazes de transferir os dados necessários, entre outras áreas (Wu et al., 2014).

A maneira mais comum de armazenamento de dados é através do formato **textual** como, por exemplo em revistas, artigos, páginas da internet, mídias sociais, códigos e *logs* de aplicações, avaliações de produtos e serviços (*reviews*), entre outras, e o tratamento desses dados textuais é chamado de mineração de texto. Enquanto a mineração de dados lida com dados estruturados que são gerados em aplicações de software, sistemas ERP, planilhas, bancos de dados estruturados, e outras aplicações, a mineração de texto deve lidar com dados não-estruturados, ou seja, dados em formato textual como nos exemplos citados no início deste parágrafo (Li et al., 2022; Gôlo et al., 2021).

A classificação de texto é uma das tarefas mais importantes na descoberta de informações a partir de formas textuais. Podem-se citar as aplicações em Processamento de Linguagem Natural (PLN) como análise de sentimentos, semântica de pareamento de tex-

tos (*text matching*), modelagem de tópicos (Aggarwal and Zhai, 2012a,b). Usualmente, a maioria dos sistemas ou aplicações de categorização, armazenamento ou utilização de documentos e textos podem ser destrinchados em um fluxo direcionado (*pipeline*) de quatro fases: extração de características (*features*), redução de dimensionalidade, seleção de classificadores e avaliação do modelo (Kowsari et al., 2019).

Todo esse volume de dados gerado cria uma oportunidade para a área de aprendizado de máquina (*machine learning*): quanto mais dados gerados, maior é a capacidade de se desenvolver modelos de abstração, mas também promovem um problema na acurácia de classificação desses dados. Apesar de existir uma certa acurácia na classificação de texto de forma manual, ela não é viável em grandes volumes e está sujeita a erros e falhas humanas (Korde, 2012). Por isso, é desejável utilizar técnicas de aprendizado de máquina para automatizar essas tarefas visando entregar resultados mais confiáveis e agéis (van Engelen and Hoos, 2020).

Os algoritmos de mineração de texto podem usar o aprendizado de máquina supervisionado, semi-supervisionado ou não-supervisionado (Hassani et al., 2020). A forma mais usual de aprendizado de máquina é o *aprendizado supervisionado*, mas essa técnica demanda que um proporção grande dos dados estejam previamente rotulados para as respostas do supervisor sejam orientadas. Além disso, possuir um conjunto de dados de treinamento previamente rotulado e grande o suficiente para treinar bem o modelo normalmente não é simples. Assim, o aprendizado semi-supervisionado possibilita encontrar soluções para problemas com menor número de informações previamente conhecidas. No aprendizado semi-supervisionado, dados rotulados são combinados com dados não rotulados para realizar o aprendizado. Além desses, existem ainda as técnicas associadas ao aprendizado positivo e não-rotulado (*Positive Unlabeled Learning* - PUL), que se aproxima do aprendizado de máquina semi-supervisionado (de Paulo Faleiros et al., 2020).

A introdução de técnicas de aprendizado em classificação de texto tem como objetivo aprender um modelo de classificação por um conjunto de documentos já rotulados e tentar propagar esse aprendizado para documentos que não tenham sido rotulados ou classificados (Li et al., 2016). Vale ressaltar que a utilização do aprendizado de máquina tem sido bastante adequado para classificar textos de documentos, mas necessita de um grande número de documentos rotulados e pré-categorizados para alcançar uma acurácia aceitável (Nigam et al., 2000). Diante desse cenário, os algoritmos de classificação de texto a partir de um aprendizado de máquina não-supervisionado têm sido bastante explorados no sentido que esses devem abordar a classificação de texto através da categorização e similaridade das palavras contidas nos documentos (Haj-Yahia et al., 2019).

Um dos problemas mais estudados em aprendizado de máquina se trata da classificação binária, onde em um conjunto de dados, quase totalmente ou totalmente rotulado, deve-

se treinar um modelo para aprender a classificar os dados dentro das classes positivas ou negativas. O PUL é uma variante desse problema, e uma das suas principais diferenças com o problema de classificação binária é que ele pressupõe o uso de dados não-rotulados do conjunto de treinamento (Bekker and Davis, 2020). Um dos motivos de PUL estar sendo bastante estudado ultimamente é o fato de dados positivos e não-rotulados **(dados *PU*)** surgirem em diversas aplicações importantes. Vamos citar alguns exemplos de aplicações caracterizadas por dados positivos e não-rotulados.

- Na medicina, geralmente acontece a detecção de alguns casos positivos e uma grande presença de falsos-negativos, ou então um prontuário médico de um paciente indica qual doença aquele paciente possui, mas geralmente não indica quais doenças aquele paciente não possui. A utilização de PUL ajuda a detectar a presença de doenças através da definição dos poucos conhecidos como positivos e do restante dos dados como não-rotulados. O mesmo pode ser aplicado para pandemias, como a de COVID-19, onde muitos dos casos eram falsos-negativos. Também para identificação em biomedicina de genes de doenças (Bekker and Davis, 2020).
- Em páginas Web, dados PU gerados formam uma oportunidade de aplicação de PUL. O histórico de navegação de um usuário forma o conjunto de dados positivos, enquanto o restante das páginas não visitadas constituem o conjunto de dados não-rotulados. Assim, um sistema de recomendação pode utilizar desse conceito para recomendar páginas que podem ser de interesse (Jaskie and Spanias, 2019).
- A aplicação para detecção de notícias falsas (*fake news*) também é muito utilizada atualmente: de posse de poucas notícias sabidamente verdadeiras, consegue-se aumentar a desempenho de classificação de notícias falsas e verdadeiras (He et al., 2020).
- Propagandas personalizadas na internet utilizam métricas de páginas visitadas e cliques para indicar exemplos positivos de páginas e propagandas. Entretanto, outras páginas ou propagandas podem ser de interesse e não devem ser tratadas como exemplos negativos (Bekker and Davis, 2020).

Normalmente, a classificação de texto é aplicada através do aprendizado multi-classe, onde todos os documentos do *corpus* são rotulados em diversas classes. Entretanto, essa abordagem possui desvantagens, dado que considera-se que o corpus, ou seja, o conjunto de dados (*dataset*) esteja completamente rotulado, o que é humanamente inviável de se fazer, caso seja um conjunto de dados de uma aplicação do mundo real (Carnevali et al., 2021). Além disso, deve-se considerar, que, em alguns casos, os dados podem pertencer a apenas duas classes e também o fato de que, dado que a rotulação completa é inviável,

a consideração dos documentos não-rotulados no processo de aprendizagem é importante para a avaliação de resultados (Jaemin et al., 2022).

A questão envolvida em **classificar dados positivos e não-rotulados** surgiu no aumento acelerado de dados complexos, muitas vezes textuais, das mais diversas fontes e das aplicações mais complexas existentes, onde existem muitas características dos dados e muitas interdependências relacionais entre os dados em si, e é necessário classificar eficientemente esses dados. Geralmente, o custo para rotular dados é alto, e então necessita-se de métodos eficientes e eficazes de se classificar fontes de dados textuais ou outras fontes contendo uma pequena parcela de dados rotulados positivamente e uma grande parcela dos dados sem rótulos (Li et al., 2016). Um exemplo de PUL comparado a aprendizados semi-supervisionados e supervisionados é demonstrado na **Figura 1.1**.

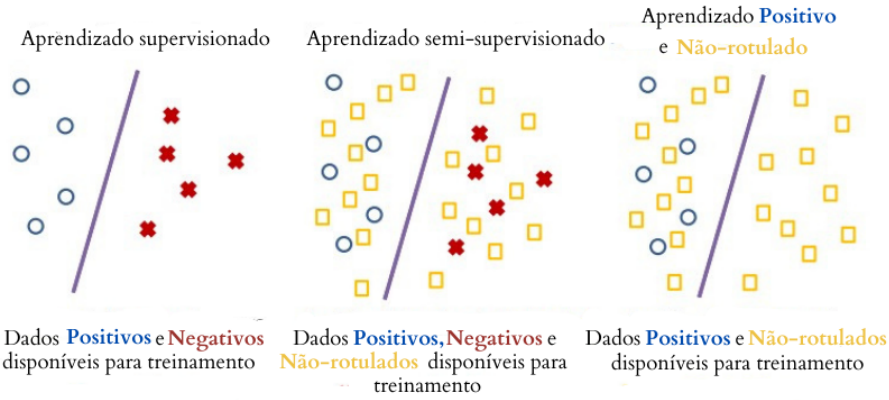


Figura 1.1: Exemplo de aprendizado supervisionado, semi-supervisionado e *positive unlabeled*. Fonte: Figura retirada de (Wu et al., 2021)

1.2 Lacunas

Como dados *PU* surgem em muitas aplicações, estudos em torno do aprendizado a partir desses tipo de conjunto de dados tendem a continuar sendo intensamente ativos. Por esse motivo, diversas lacunas, ainda não completamente exploradas, devem ser alvo de pesquisas e desenvolvimentos nos anos a seguir.

Algumas lacunas identificadas nos trabalhos mais recentes realizados sobre PUL indicam diversidade de questões a serem investigadas. Uma dessas lacunas refere-se a realizar a avaliação da desempenho de classificação de um algoritmo observando a presença de dados não-rotulados. Além disso, muitos trabalhos desenvolvem novas técnicas de classificação e colhem resultados a partir de uma gama pequena de conjunto de dados, onde

muitos trabalhos aplicam algoritmos de PUL em um grupo de dois a quatro conjuntos de dados, o que gera uma nova lacuna: aplicar uma técnica de classificação a um conjunto maior de conjunto de dados (Bekker and Davis, 2020; Qachfar et al., 2022).

Como o algoritmo de fatoração de matrizes não-negativas *(*Non-negative Matrix Factorization - *NMF)* é foco de estudo nesse trabalho, algumas lacunas na sua aplicação são bastante conhecidas. Uma dessas lacunas, abordada na Seção 2.2.2, refere-se à dificuldade da aplicação do *NMF* quando a interpretabilidade do dado é muito relevante. O *NMF* deve ser aplicado em dados não-negativos e esparsos, o que deve ser considerado como condição no pré-processamento dos dados textuais, embora não seja uma lacuna no presente estudo.

Existe ainda, embora seja uma área de aprendizado de máquina bastante ativa, escassez de trabalhos abordando aprendizado por redes neurais profundas em problemas de *PUL*. Em Xu et al. (2020), um dos trabalhos futuros a serem desenvolvidos diz respeito à aplicação de redes neurais para classificação de sentimento, assim como em Zhang et al. (2018), que propõe o estudo de outras técnicas de classificação como redes neurais. Como será abordado na Seção 2.2.3, a partir de algumas limitações do *NMF*, foram propostas abordagens de redes neurais, ou seja, utilizando estruturas multicamadas.

Assim, visando lidar com a dificuldade em se construir uma estrutura hierárquica ótima para atacar esses problemas, o aprendizado profundo (*deep learning*) foi aplicado junto ao *NMF*, pois auxilia no processo de dividir um problema com tarefas complexas em tarefas mais simples, culminando no desenvolvimento do algoritmo *Deep NMF*, onde se transformam as iterações do algoritmo em camadas de uma rede neural. Dessa forma, a aplicação do *Deep NMF* tem ganhado espaço na literatura, ainda demonstrando possuir algumas lacunas a serem atacadas. Muitas abordagens utilizam a *Distância Euclidiana* como função objetivo, entretanto ainda carece a aplicação de outras funções objetivo, como a de *Kullback-Leibler* no *Deep NMF* (W. Chen, 2022), o que é tratado na reconstrução de erro para o *framework* de Nasser et al. (2021). Outras medidas como o índice de Jaccard não possuem bons resultados quando aplicados a problemas de distribuição de frequências, sendo aplicados em distribuições de multiplicidade (Bonnici, 2020).

De forma resumida, as lacunas a serem abordadas no presente estudo são:

- Consideração dos dados não rotulados no processo de aprendizagem;
- Aplicação de uma técnica de classificação de textos para dados PU em uma gama maior de conjuntos de dados;
- Uso de técnicas de redução de dimensionalidade, como o NMF, de forma a auxiliar o processo de classificação (no caso do presente estudo, foi feita uma adaptação desse tradicional algoritmo para realizar a classificação em dados PU)

- Aprendizado por redes neurais em problemas de PUL, principalmente quando se aborda o algoritmo *Deep NMF*;
- Aplicação de funções objetivo diferentes da Distância Euclidiana, como a função de *Kullback-Leibler* tanto no NMF como no Deep NMF, pois essa métrica divergente quantifica a dissimilaridade entre duas distribuições de probabilidade e é adequada para ser adotada no NMF (Hien and Gillis, 2020).

E, assim, as seguintes questões de pesquisa foram formuladas e devem ser respondidas no trabalho:

1. Quais os métodos promissores para evolução da classificação semi-supervisionada de dados textuais considerando a utilização de métodos de PUL?
2. A aplicação do algoritmo NMF introduz melhorias na classificação semi-supervisionada de dados do tipo texto?
3. A aplicação de técnica de aprendizado profundo, como o algoritmo *Deep Non-Negative Matrix Factorization*, Deep NMF, resulta em resultados satisfatórios comparado ao estado da arte?

1.3 Hipótese

Baseado nos dados presentes nas três seções anteriores, chega-se à seguinte hipótese:

A hipótese do presente trabalho é que realizar modificações no método de redução de dimensionalidade, especificamente o NMF, para resolver problemas de ***PUL*** e propor uma abordagem multicamadas para aplicar o NMF, o *Deep NMF*, é uma estratégia para obter resultados competitivos com o estado da arte na aplicação de *PUL* para classificação de textos.

1.4 Objetivos

Este trabalho tem como objetivo investigar, analisar e comparar técnicas de resolução de problemas de aprendizado com exemplos positivos e não-rotulados, realizando adaptações estruturais no algoritmo de Fatoração de Matrizes Não-Negativas (*Non-Negative Matrix Factorization - NMF*) e aplicando a sua vertente para aprendizado profundo, a Fatoração Profunda de Matrizes Não-Negativas (*Deep Non-Negative Matrix Factorization - Deep NMF).*

Para atender este objetivo, os seguintes objetivos específicos são destacados:

- Modificar o método de redução de dimensionalidade, *NMF*, para ser aplicado como classificador em problemas de *PUL*;
- Comparar os resultados do NMF adaptado com outro(s) algoritmos do estado da arte;
- Implementar uma solução de aprendizado profundo através do Deep NMF na base de dados utilizada na versão adaptada do NMF.

1.5 Contribuições

Apresentam-se como contribuições do presente trabalho:

- O desenvolvimento de um método de PUL aplicável a dados textuais;
- Adaptação do método NMF para aplicação em problemas de PUL;
- Proposição de uma abordagem do método *Deep NMF* para aplicação em problemas de PUL.

1.6 Estrutura do Documento

Para o desenvolvimento do presente trabalho, a seguinte sequência de atividades foi realizada:

Primeiramente, foi realizada uma revisão dos conceitos de aprendizado em *positive unlabeled learning*. O Capítulo 2 fornece os principais conceitos dessa área, focando nas métodos baseados em *Positive Unlabeled Learning* (Seção 2.1), que são utilizados posteriormente no trabalho. Foram investigadas os mecanismos de rotulação, definições sobre os dados utilizados em uma metodologia usual de PUL, e detalhados alguns métodos de *PUL* para aplicação.

Após isso, é feita uma introdução a métodos de aprendizado para classificação de textos na seção 2.2, detalhando as formas de representação de texto, as técnicas mais adotadas de redução de dimensionalidade, aprofundando nas técnicas que serão foco desse estudo, como o *Non-Negative Matrix Factorization* e o *Deep Non-Negative Matrix Factorization.*

Para a questão de pesquisa definida na área da análise da aplicabilidade do PUL na classificação de textos, uma revisão sistemática de literatura foi realizada. O Capítulo 3 explica e detalha o método aplicado para a revisão de literatura, as questões que balizam esse estudo, os parâmetros adotados para a pesquisa, e as estatísticas e análise de resultados da revisão.

No Capítulo 4, são descritas a metodologia de trabalho proposta, indicando a abordagem de *Positive Unlabeled Learning* a ser seguida e as métricas para avaliação e os trabalhos relacionados.

No Capítulo 5, os resultados de desempenho de classificação além de outras métricas dos algoritmos NMFPUL e Deep NMF são avaliadas em comparação com outros algoritmos baseline, abordados no estado da arte.

Por fim, o Capítulo 6 apresenta as conclusões do presente trabalho, indicando também as possíveis limitações e abordagens para trabalhos futuros.

1.7 Limitações do Trabalho e Ameaças a Validade

O presente trabalho se limita ao escopo de PUL, averiguando técnicas que possuem capacidade de alcançar bons níveis de classificação de dados textuais. Uma das limitações do presente trabalho deve-se ao fato de que um único algoritmo de redução de dimensionalidade foi selecionado para adaptação visando o objetivo de classificar os dados, não utilizando de modo efetivo uma redução de dimensionalidade do problema, mas usando uma abordagem probabilística, que é possível através dos conceitos do NMF. Outros algoritmos da mesma linha, como a Análise de Componentes Principais (*Principal Component Analysis - PCA*), não caberiam na presente análise dado que esses algoritmos funcionam para reduzir a dimensionalidade do problema, o que não era foco da pesquisa. O NMF se diferencia desse último por ser aplicado em dados não-negativos e em dados esparsos, sendo características dos datasets usados na pesquisa, enquanto no PCA podem ter dados negativos e positivos misturados, as componentes principais tendem a ser densas, onde cada componente pode ser uma combinação das variáveis originais, o que pode dificultar a interpretação. Ou seja, no NMF, um valor baixo é associado a um tópico, enquanto um valor alto é associado a outro tópico, enquanto no PCA os dados são imersos no espaço, significando que um valor baixo não necessariamente significa um valor pior. Logo, pode-se haver uma expansão do trabalho ao se analisar outras técnicas similares, mas que possuam essa abordagem probabilística do problema, observando se elas podem resultar em desempenhos melhores, o que será levantado como possibilidade de trabalho futuro.

A metodologia utilizada levou em conta a aplicação da técnica NMFPUL, do *Deep NMF* e outros algoritmos do estado da arte em coleções de documentos de Rossi et al. (2013), onde de antemão os dados passaram por um processo de remoção de *stop words*, tokenização (separação dos textos em palavras), limpeza de dígitos irrelevantes e lematização das palavras (transformar palavras em forma original). Por esse motivo, aplicar o modelo proposto em um novo conjunto de texto requer que o processo seja copiado e apli-

cado, formatando os dados de forma a obter uma coluna com as palavras do documento e uma coluna com a classe correspondente daquele conjunto de palavras.

Capítulo 2

Classificação de textos usando Positive Unlabeled Learning

Aprendizado de máquina é uma das áreas da Inteligência Artificial e da Ciência da Computação que mais se desenvolveu, sendo destinada a desenvolver algoritmos que possuam a capacidade de aprender a partir de dados de entrada e, então, realizar tarefas de classificação, identificação, categorização, clusterização, entre várias outras tarefas (Shalev-Schwartz and Ben-David, 2014).

Algoritmos de Aprendizado de máquina podem ser classificados de diversas maneiras, sendo as principais: quanto ao modo de supervisão, de acordo com o *feedback* disponível para o processo de aprendizado; quanto à tarefa, de acordo com saída desejada do processo de aprendizado. Quanto ao modo de supervisão, existem os tipos: aprendizado supervisionado, aprendizado não-supervisionado, aprendizado semi-supervisionado, aprendizado por reforço (Mahesh, 2020).

O *aprendizado supervisionado*, sendo um dos mais comuns, possui aplicabilidade bastante ampla em problemas de classificação e regressão. Pode ser definido como o processo de treinamento de algoritmos, a partir de um conjunto de dados rotulados denominado conjunto de dados de treinamento, visando classificá-los. Assim, o algoritmo, durante o processo de aprendizado, é capaz de identificar se as respostas estão coerentes de acordo com a resposta esperada, realizando ajustes para que o erro seja minimizado. Algumas técnicas comumente utilizadas são a regressão logística, redes neurais artificiais, máquina de suporte vetorial, ou *Support Vector Machine*, Árvores de decisão, Redes Bayesianas ou *Naive Bayes*, K-Vizinhos Mais Próximos ou *K-nearest neighbors* (Mahesh, 2020).

No *aprendizado não-supervisionado*, não há disponível nenhuma informação prévia sobre o rótulo ou categoria do dado. Os algoritmos devem descobrir e identificar por si só, a partir dos dados, informações relevantes. Quando há novos dados entrantes, o algoritmo deve, a partir do aprendizado inicial, identificar a classe dos novos dados. Muito

usualmente, são aplicados em problemas de clusterização e redução de dimensionalidade, como em Análise de Componentes Principais, PCA, ou o *K-Means* (Mahesh, 2020).

Métodos de aprendizado por reforço assemelham-se aos métodos de aprendizado supervisionado, mas não são treinados a partir de dados de amostra apenas. Ao invés disso, esses métodos realizam o aprendizado a partir de tentativas e erros, gerando recompensas para o algoritmo quando acerta e propagando os erros recursivamente, visando corrigí-los (Shalev-Schwartz and Ben-David, 2014).

Já no *aprendizado semi-supervisionado*, utiliza-se uma combinação do aprendizado supervisionado e do aprendizado não-supervisionado. Fazendo uso de relações entre os dados não rotulados e rotulados para compensar a falta de rótulos (Silva, 2008), os métodos de aprendizado semi-supervisionado assumem que há, no conjunto de treinamento, dados rotulados e não-rotulados. A motivação para a pesquisa deste tipo de algoritmo está no fato de que, exemplos não rotulados são facilmente encontrados e mais baratos de serem coletados, comparado aos conjuntos de dados rotulados.

Uma nova vertente de métodos de aprendizado de máquina surgiu, o PUL, sendo esse mais próximo do aprendizado semi-supervisionado, mas com diferenças claras. Aprendizado positivo e não-rotulado explicitamente utiliza dados não-rotulados no processo de aprendizagem, sendo importante ressaltar que, geralmente, os dados que estão rotulados são de apenas uma classe. O que o diferencia dos demais é que apenas um pequena porção dos dados positivos são rotulados e nenhum dos dados negativos são rotulados, ou seja, deve-se realizar o treinamento do classificador a partir de dados positivos e não-rotulados, ou mais comumente conhecido como dados *PU*. É uma técnica útil para lidar com problemas em que coletar exemplos negativos é um desafio, mas é possível obter exemplos positivos confiáveis. Já no aprendizado semi-supervisionado, sendo uma especialização do aprendizado supervisionado, utiliza, quando existente, dados não-rotulados no processo de treinamento, mas, usualmente, alguns dados rotulados de todas as classes estão disponíveis (Bekker and Davis, 2020).

2.1 Positive Unlabeled Learning

2.1.1 Notações Matemáticas e Definições básicas

A Tabela 2.1 contém as notações que serão utilizadas ao longo deste capítulo de fundamentação teórica. Um conjunto de dados PU é representado como um conjunto de triplas, especialização de uma n-upla de 3 elementos, (x, y, l), onde x é um vetor de atributos, y é a variável que indica a classe e l, uma variável binária indicando se o exemplo é rotulado.

Tabela 2.1: Notações matemáticas adotadas no presente capítulo, adaptado de (Bekker and Davis, 2020)

Notação	**Descrição**
(x, y, l)	tripla de elementos de um dataset PU
x	Vetor de atributos
y	Variável indicando a classe do exemplo
l	Variável indicando a rotulação de um exemplo
X	Conjunto de vetores de atributos
Y	Conjunto de variáveis indicando a classe do exemplo
L	Conjunto de variáveis indicando a rotulação de um exemplo
α	Probabilidade da Classe principal, onde α =P(y = 1)
$e(x)$	Função de propensão, $P(l = 1\|y = 1, x)$
c	Frequência de rótulo
$f(x)$	Função de probabilidade
$f_+(x)$	Função de probabilidade dos positivos
$f_-(x)$	Função de probabilidade dos negativos
$f_l(x)$	Função de probabilidade dos rotulados
$f_u(x)$	Função de probabilidade dos não-rotulados
Pe_+	fator de penalização de ruído para a classe positiva
Pe_-	fator de penalização de ruído para a classe negativa
n	*noise* ou ruído

Para um dado exemplo, se ele é positivo ou pertence à classe positiva, então $y = 1$, e se ele é negativo ou pertence à classe negativa, $y = 0$. Em um *dataset* PU, se o dado é rotulado, temos que $l = 1$ e se não é rotulado, $l = 0$. Também, chamamos de classe principal a classe positiva, ou seja, a classe alvo do problema, aqui definida como $\alpha = P(y = 1)$, onde P indica a função probabilidade (Bekker and Davis, 2020).

Em um problema de PUL, temos que se um exemplo é rotulado, ele pertence à classe positiva, ou seja, podemos afirmar que $P(y = 1|l = 1) = 1$. Por outro lado, caso o exemplo não seja rotulado, ele pode pertencer tanto à classe positiva, classe principal, quanto à classe negativa.

2.1.2 Mecanismo de Rotulação

Para entender como os exemplos positivos rotulados são selecionados, devemos entender como esses exemplos são originados, dado o dataset original. Existem duas maneiras principais de identificar essa questão: os dados são oriundos de apenas um dataset, que é uma amostra independente e identicamente distribuída, i.i.d, da população real, chamado de cenário de conjunto único; ou os dados são oriundos de dois conjuntos de dados, sendo um deles apenas com os exemplos positivos da população e o outro conjuntos de dados, composto apenas por exemplos não-rotulados, uma amostra independente e identicamente distribuída, i.i.d, da população real, chamado de cenário de *case-control* (Bekker and Davis, 2020).

Além das maneiras mais relevantes para a presente pesquisa, citados acima, outros mecanismos de rotulação podem ser citados. No Cenário de Amostras Compostas (*Mixture Proportion Estimation*) onde os dados vêm de vários datasets diferentes (origens), mas com a rotulação de alguns dos dados positivos (com diferentes classes de origem) (Garg et al., 2024).

Para fins de estudo do mecanismo de rotulação, o foco será dado para o primeiro cenário definido, onde uma fração dos exemplos positivos são rotulados, seguindo a função de propensão de rotulagem, conforme (Equação 2.3). Considerando que as equações descritas nesse tópico foram adaptadas e retiradas de (Bekker and Davis, 2020). Uma fração c dos exemplos positivos é selecionada para ser rotulada, seguindo seus escores de propensão individuais e(x), e assim o conjunto de dados possui uma fração $\alpha.c$ de exemplos rotulados.

$$X \approx f(x) \tag{2.1}$$

$$X \approx \alpha f_{+}(x) + (1-\alpha) f_{-}(x) \tag{2.2}$$

$$X \approx \alpha e(x) f_{l}(x) + (1-\alpha e(x)) f_{u}(x) \tag{2.3}$$

onde f(x) é a função de probabilidade da população e e(x) é a função de propensão, que indica a probabilidade de um exemplo positivo ser rotulado.

A partir disso, podemos entender como mecanismo de rotulação pode ser aplicado para dados PU.

Inicialmente, definindo a função de probabilidade de um exemplo ser rotulado em relação à probabilidade de ser positivo:

$$f_l(x) = P(x|l = 1, y = 1) \tag{2.4}$$

$$f_l(x) = \frac{P(l = 1|x, y = 1)}{P(l = 1, y = 1)} P(x, y = 1) \tag{2.5}$$

$$f_l(x) = \frac{e(x)}{c} f_+(x) \tag{2.6}$$

Também entende-se que o exemplo é não rotulado caso ele seja um exemplo negativo ou caso seja positivo, mas não foi selecionado pelo mecanismo de rotulação para ser rotulado. Dessa forma, para permitir que haja aprendizado direto a partir de dados PU, deve-se compreender e definir a abordagem utilizada em relação ao mecanismo de rotulação e a distribuição dos exemplos nas classes. Para o mecanismo de rotulação, existem três abordagens principais, descritas a seguir.

Selecionado aleatoriamente - SAR

Também conhecida como Selecionado Aleatoriamente (*Selected at Random - SAR*), nessa hipótese, a seleção de exemplos positivos para serem rotulados depende totalmente dos seus atributos, o que torna essa vertente a mais generalista, pois entende que muitas aplicações do mundo real são influenciadas pelo bias presente, como a identificação de spam em e-mail depende da forma persuasiva em que o e-mail é construído e escrito ou sistema de recomendações dependem da ordem em que os primeiros produtos ou serviços são apresentados, enviesando as recomendações seguintes.

Dessa forma, a função de propensão de rotulagem é:

$$e(x) = P(l = 1|x, y = 1) = c \tag{2.7}$$

Como será abordado no tópico a seguir, o SAR é considerado como uma das variantes do SCAR de forma enfraquecida, tentando levar em conta a realidade de que o viés (*bias*) é intrínseco às aplicações reais (Jaskie and Spanias, 2019).

Selecionado completamente de forma aleatória - SCAR

Também conhecida como *Selected Completely at Random - SCAR*, nessa hipótese, os exemplos rotulados são um subconjunto do conjunto de exemplos positivos, ou seja, todo exemplo positivo tem a mesma probabilidade c de ser rotulado. Ao contrário do SAR, o SCAR assume que qualquer viés no conjunto de rotulados será transferido para o viés do modelo, e assim, utiliza a escolha randômica dos positivos para serem rotulados para

remover ao máximo o bias de seleção. Essa abordagem tem sido bastante comum nos estudos e pesquisas relacionados à aplicação de PUL.

Como exemplo, a decisão de se conceder financiamento para clientes, no SCAR, dentre aquelas aplicações que qualificam para um financiamento, será jogada uma moeda para escolher quais destas receberão o aceite. Assim, o viés de seleção é retirado, evitando transferi-lo para o modelo.

Aqui, a probabilidade de um exemplo positivo ser rotulado é análogo à frequência de rótulo c:

$$e(x) = P(l = 1|x, y = 1) = P(l = 1|y = 1) = c \tag{2.8}$$

E a probabilidade de um exemplo ser rotulado é proporcional à probabilidade dele ser positivo:

$$P(y = 1|x) = \frac{1}{c}P(l = 1|x) \tag{2.9}$$

Como citado no tópico anterior, quando definido o SAR, existem algumas iniciativas de se propor uma hipótese *enfraquecida* do SCAR, visando considerar a presença de algum viés, assumindo que o conjunto de rotulados a partir dos positivos não possui uma frequência de consistência, como proposto em Bekker and Davis (2018) e Bekker et al. (2019). Uma forma é aplicar a função de propensão de rotulagem como função dos atributos do conjunto de dados e o algoritmo de maximização de expectativa (Expectation Maximization - *EM*) para o problema de PUL (Jaskie and Spanias, 2019).

Além disso, há a abordagem de se minimizar a função de risco de classificação para lidar com o viés de seleção, ou seja, assume que a probabilidade de um exemplo ser rotulado como positivo é proporcionalmente similar à probabilidade do exemplo, de fato, ser positivo (Kato et al., 2019). E, por último, Kyrio et al. (2017) utiliza estimadores de risco não-negativos para produzir melhores resultados no que tange à utilização em soluções de aprendizado profundo.

Lacuna Probabilística

Sendo uma especialização do SAR, pois também se baseia no fato de que o mecanismo de rotulagem depende dos atributos, a Lacuna Probabilística considera que os exemplos positivos que se parecem ou são próximos de exemplos negativos são menos prováveis de serem rotulados. A função de lacuna probabilística (*Probabilistic Gap*), é definida conforme a fórmula:

$$e(x) = f(\Delta P(x)) = f(P(y = 1|x) - P(y = 0|x)) \tag{2.10}$$

Assim, conclui-se que quanto menor o valor da função de lacuna probabilística, menor a probabilidade do exemplo ser rotulado. Essa á uma propriedade bastante importante no estudo de problemas de PUL, e é comumente utilizada para se identificar os *Reliable Negatives* (RN), ou seja, aqueles exemplos que possuem a maior chance de serem realmente negativos, e podem ser identificados quando o valor da função de lacuna probabilítisca de exemplos não rotulados é menor do que o menor valor encontrado para a função de lacuna probabilística de exemplos rotulados.

2.1.3 Definições sobre os dados

Para realizar um estudo e utilizar métodos de *PUL* para classificar dados, deve-se levar em conta algumas condições em relação aos mesmos. Essas condições são basilares para se aplicar adequadamente em um problema e para simular uma condição real de aprendizado de máquina a partir de dados positivos e não rotulados.

Inicialmente, sendo uma condição já abordada anteriormente ao definir preliminarmente o que é PUL, deve-se considerar que a totalidade dos exemplos que não estão rotulados no conjunto de dados, pertencem à classe negativa dos dados, ou seja, nenhum exemplo que não esteja rotulado pertence à classe principal ou classe alvo. Essa é uma premissa básica para realizar o processo de treinamento e também para o mecanismo de rotulação.

E como consequência da primeira condição, tem-se que qualquer dado que esteja rotulado, pertence à classe positiva ou classe principal. Dessa forma, para o processo de treinamento do modelo, os exemplos rotulados da classe positiva podem ser importantes para identificar quais serão os exemplos mais prováveis de serem negativos, os negativos confiáveis ou (*Reliable Negatives*), RN.

Como outra assunção a ser considerada, é a de que deve haver uma forma clara de separação das classes, ou seja, deve haver um parâmetro ou conjunto de parâmetros que identifique perfeitamente a diferença entre as classes, durante o processo de treinamento. Em outras palavras, existe uma função f que separa perfeitamente os exemplos do conjunto de dados em duas classes.

A quarta assunção que devemos fazer é de que exemplos que estejam próximos entre si tem maior probabilidade de possuírem o mesmo rótulo, condição essa que é basilar para o método de PUL chamado de *Técnica em dois passos*. Essa propriedade, comumente denominada de *smoothness*, ou suavidade, é comumente utilizada em abordagens de grafos (Carnevali et al., 2021; Wu et al., 2021).

2.1.4 Métodos de *Positive Unlabeled Learning*

Existem diversos métodos que resolvem o problema de dados positivos e não rotulados, sendo cada um deles melhor endereçado para resolver diferentes problemas reais. Alguns desses métodos serão abordados nesse tópico, sem no entanto estressar todas as vertentes de aplicação, mas explicitando aqueles mais usuais e também aqueles que melhor se adequam ao problema em questão nessa pesquisa.

Aprendizado Enviesado

Também conhecido como *Biased Learning*, o Aprendizado Enviesado trata os exemplos negativos como não rotulados e sugere que exemplos positivos classificados erroneamente como negativos podem trazer graves problemas no processo de aprendizagem do modelo de classificação, o que é chamado de ruído na classe. As diversas abordagens desse método consideram que há ruído tanto na classe positiva quanto na classe negativa, entretanto definem que o nível de ruído é condicionado pela classe (Jaskie and Spanias, 2019).

Como já levantado anteriormente, ao abordar os mecanismos de rotulação, esse é um método que é adotado frequentemente junto com o mecanismo *SCAR*, pois consideram que o viés existente nos dados rotulados serão transferidos para o modelo. Sendo assim, essa abordagem realiza uma seleção randômica dos exemplos rotulados para diminuir ao máximo o viés de seleção. Para o processo de aprendizado, são colocados penalidades maiores para aqueles exemplos positivos erroneamente classificados. Dessa forma, usualmente são feitas aplicações de modelos de forma iterativa, para que penalidades sejam aplicadas no processo de classificação e, assim a eliminação dos ruídos seja maximizada, entregando uma classificação mais adequada ao final do processo.

Um dos algoritmos mais utilizados nesse tipo de método é o *Support Vector Machine - SVM*. A versão enviesada (*Biased SVM*) é aplicada tradicionalmente através da penalização da classificação de positivos e negativos, considerando a penalização distinta para cada classe. Em (He et al., 2020), o *Biased SVM* é aplicado principalmente em casos onde há um desbalanceamento dos dados, ou seja, existe uma porção consideravelmente superior de uma das classes frente à outra. Nesse mesmo estudo, é definido que a razão entre o fator de penalização dos exemplos positivos para os exemplos negativos equivale à razão entre o tamanho da amostra de exemplos negativos para os positivos.

$$Pe_+/Pe_- = A_-/A_+ \tag{2.11}$$

Onde Pe_+ é o fator de penalização para a classe positiva, Pe_- é o fator de penalização para a classe negativa, A_+ é o tamanho da população de exemplos positivos e A_- é o tamanho da população de exemplos negativos.

As versões mais recentes de aplicação desse algoritmo visam trazer a iteratividade, onde as penalidades são ajustadas a cada iteração, visando corrigir os erros de classificação. Por vezes, pesos são atribuídos aos dados não rotulados, onde a otimização dos pesos deve ser fator essencial para reduzir a complexidade computacional (Liu et al., 2022b).

Aprendizado em dois passos

Mais comumente denominado de *Two-step technique*, essa abordagem realiza o processo de classificação dos dados, segundo alguns parâmetros, através de dois passos.

Etapa 1 - Identificar os *Reliable Negatives - RN*: utiliza diversas técnicas distintas para identificar aqueles exemplos que mais possuem a probabilidade de serem negativos. Uma das técnicas identifica os exemplos que são muito distantes dos positivos, classificados como RN. Diversos métodos podem ser usados: *k-NN, Spy, Mapping-convergence(MC), 1-DNF*, etc.

Etapa 2 - Treinar o classificador através de um algoritmo semi-supervisionado ou de um algoritmo supervisionado: classificador deve utilizar os positivos, os RN e os não-rotulados para realizar o treinamento. Diversos algoritmos como SVM e o *Naives Bayes* são utilizados para rotular os exemplos restantes ainda não-rotulados.

Conforme descrito anteriormente, no tópico sobre a definição dos dados, a condição de suavidade dos dados é essencial para a utilização dessa abordagem, indicando que exemplos devem ser analisados para identificar mais diretamente aqueles exemplos mais prováveis de serem negativos, dado o rol de positivos que já estão rotulados.

Para o primeiro passo desse método, diversos algoritmos podem ser aplicados, como citado. Em Li and Liu (2003), o algoritmo *Rocchio* assume que todos os exemplos não rotulados são negativos e realiza o treinamento do classificador a partir dessa premissa. Então, após isso, aplica-se o classificador no conjunto inicial, sem rotulação total, para identificar aqueles que são realmente mais prováveis de serem negativos (Jiang et al., 2021). Já em Zhang (2024), o objetivo da aplicação da técnica em dois passos está em detectar e remover algumas entidades não-rotuladas com alta probabilidade de serem negativas como preparação para a aplicação da técnica de amostragem negativa no segundo passo do algoritmo.

Outra técnica comumente aplicada no primeiro passo dessa abordagem, é o *Spy*, onde alguns exemplos são escolhidos como espiões de forma randômica do conjunto de positivos, e utiliza-se os exemplos positivos, os exemplos não rotulados e os espiões para realizar as fases de treinamento e teste (Jiang et al., 2021; Liu et al., 2022a).

Outros exemplos de técnicas a serem aplicadas no primeiro passo são o *1-DNF*, *k-vizinhos mais próximos - KNN*, *Naive-Bayes*, *PGPU* (He et al., 2018).

Para o segundo passo dessa abordagem, onde algoritmos semi-supervisionados ou supervisionados são aplicados, a aplicação de variações do *Support Vector Machine* (Shuqin and Jing, 2019; Liu et al., 2022b; Yang et al., 2018) e do algoritmo de *Naive Bayes* são bastante usuais, seja qual for o método de PUL utilizado (Wu et al., 2020; Banerjee et al., 2018).

Incorporação da classe principal - SCAR

Como descrito no tópico 2.1.2, um dos obstáculos do aprendizado de dados positivos e não rotulados é lidar com o viés de seleção dos dados rotulados inicialmente. Por isso, diversos estudos lidam com o método de rotulação SCAR, realizando um enfraquecimento do mesmo, para que não haja viés na rotulação dos primeiros exemplos positivos, o que poderia levar a um viés também no modelo final. Na condição SCAR, a classe principal é conhecida e pode ser utilizada para fins de treinamento do classificador.

Dessa forma, seja considerando os não rotulados como negativos, seja modificando o conjunto de dados para utilizar a identificação prévia da classe principal, seja incoporando a classe principal no processo de treinamento, a assunção de que a proporção de exemplos rotulados dentre os positivos pode configurar uma metodologia de aprendizado a partir de dados positivos e não rotulados é considerada (Bekker et al., 2019).

Outras abordagens

Outras abordagens também se destacam, embora não se encaixam nas categorias anteriores. O conceito de *bagging* incrementou o desempenho de alguns classificadores, quando aplicados à problemas de PUL. Essa técnica realiza o processo de classificação através de geração randômica iterativa de subconjuntos dos dados de treinamento, como observado em (Qachfar et al., 2022). Em Bian et al. (2021), realiza-se o treinamento do classificador através do conjunto de dados positivos e do conjunto de dados não rotulados randomicamente selecionados.

As redes do tipo *Generative Adversarial Networks - GANs* realizam a modelagem das distribuições dos dados positivos e negativos. A ideia principal é a de utilizar uma rede generativa para com as distribuições possibilitar a geração de novas amostras desse conjunto (Hu et al., 2021).

2.2 Métodos de Aprendizado para Classificação de Textos

2.2.1 Representação de Texto

A representação textual indica a forma como o dado será formatado e apresentado. A aplicação de uma técnica de representação textual para conjuntos de dados em formato textual auxiliam no processo de aperfeiçoamento da classificação de textos. Geralmente, esses conjuntos de dados são compostos por documentos, contendo formato não estruturado. Dessa forma, para aplicação de determinados algoritmos de aprendizado de máquina, é necessário que seja feita uma transformação de modo a representar os dados textuais em vetores numéricos. Existem diversos mecanismos de representação textual, dentre os quais aqui serão descritos alguns que foram utilizados nos algoritmos que são objeto de estudo e implementação nessa pesquisa.

Bag of Words (BoW)

O Saco de Palavras ou *Bag of Words - BoW* é uma técnica que transforma textos em um vetor de comprimento fixo, onde cada entrada do vetor indica a ocorrência ou ausência de uma palavra constante do vocabulário do texto (Qader et al., 2019). Para cada documento de uma coleção, um vetor pode ser criado. Nesse sentido, não considera-se a repetição das palavras, ou seja, o vetor compreenderá as palavras do vocabulário, e não do *corpus*, onde esse último termo indica a totalidade das palavras constantes dos textos. Também não é considerada a ordem de ocorrência das palavras.

Essa técnica é indicada para uso em *corpus* menores, pois a sua aplicação em um *corpus* grande pode formar um vetor esparso, o que demanda recursos computacionais para a aplicação em algoritmos de aprendizado de máquina.

Term Frequency and Inverse Document Frequency (TF-IDF)

O TF-IDF pode ser definido como uma forma de calcular o quanto uma palavra é relevante para o conjunto de documentos ou *corpus*. Como está descrito no título, esse cálculo se resume na multiplicação de outros dois cálculos que definem a Frequência de Termos (*Term Frequency - TF*), que indica a quantidade de vezes que uma palavra aparece em um documento, e a Frequência Inversa de Documentos (*Inverse Document Frequency - IDF*), que indica a frequência de documentos que constam a palavra determinada (Yuntao et al., 2005).

O *TF* indica que em um documento d, essa frequência é a quantidade de vezes que uma palavra t aparece. Assim, podemos definir o *TF*, como sendo:

$$TF(t,d) = \frac{\text{contagem de t em d}}{\text{quantidade de palavras t em d}} \tag{2.12}$$

Em alguns casos também se utiliza a seguinte fórmula:

$$TF(t,d) = \frac{\text{contagem de t em d}}{\text{frequência do t mais comum em d}} \tag{2.13}$$

Assim, pode-se inferir que o *TF* atua como se alocasse um peso para um certo termo t em um documento d. Assim, cada documento pode ser representado como um vetor numérico onde cada entrada representa a frequência de aparição do termo no documento. Entretanto, o *TF* por si só não representa bem uma classe de documentos, pois uma palavra pode ter um frequência alta, mas disseminada em vários documentos.

Dessa forma, para superar esse obstáculo, o *IDF* insere um peso para palavras concentradas em poucos documentos.

A Frequência de Documentos (*Document Frequency - DF*) denota a quantidade de documentos que uma certa palavra está presente. Assim, o *IDF(t)* pode ser definido como:

$$IDF(t) = \ln \frac{(1+n)}{(1+DF(d,t))} + 1 \tag{2.14}$$

onde, DF(d,t) é quantidade de documentos que uma certa palavra está presente e n é a quantidade total de documentos da coleção.

Dessa forma, o *TF-IDF* é o produto do *TF* com o *IDF*, resultando em:

$$TF - IDF = TF(t,d) \cdot IDF(t) \tag{2.15}$$

Esse produto indica que um maior peso será dado a um termo t possua alta frequência em um documento e baixa frequência nos documentos da coleção. O resultado dessa implementação é uma matriz de documentos e termos, onde as células representam o valor de *TF-IDF* do termo na coleção.

Pode-se ainda destacar que essa técnica induz a utilização de redução de dimensionalidade, apesar de o próprio não realizá-lo integralmente (Zoya et al., 2021). Além disso, essa técnica não utiliza qualquer artifício de similaridade das palavras, considerando cada palavra de forma independente.

2.2.2 *Non-Negative Matrix Factorization*

A Fatoraçao de Matrizes não-negativas (*Non-Negative Matrix Factorization - NMF*) é um algoritmo ou conjunto de algoritmos onde uma matriz $V \in \mathbb{R}_{+}^{dxn}$ é fatorizada de forma a produzir duas matrizes: $W \in \mathbb{R}_{+}^{dxk}$ e $H \in \mathbb{R}_{+}^{nxk}$, onde as três matrizes possuem elementos não-negativos, como em:

$$V = WH^T \tag{2.16}$$

Considerando uma situação de análise sobre dados textuais, a matriz V indica os documentos constantes nas linhas e o vocabulário de palavras nas colunas. Já a matriz W indica os documentos com a sua relação de tópicos, e a segunda matriz, H, indica a contribuição de cada palavra nos tópicos, comumente chamado de saco de palavras (*bag-of-words*) (Lee and Seung, 2000).

Usualmente, os elementos da matriz V indicam ausência ou presença da palavra ou termo no documento, sendo essa uma classificação binária, ou indicam a frequência de aparições das palavras nos documentos, onde usualmente se usa o cálculo bastante utilizado em mineração de texto, *TF-IDF*, já definido na Equação 2.15, e definido abaixo com variáveis adaptadas para o contexto de palavras e documentos no NMF.

$$w_{n,d} \quad = \quad tf_{n,d} \quad \times \quad log(\frac{N}{df_n}) \tag{2.17}$$

onde:

$w_{n,d}$ é a palavra n no documento d

$tf_{n,d}$ é a frequência da palavra n no documento d

df_n é o número de documentos contendo a palavra n

N é o número total de documentos

O NMF é bastante utilizado há algum tempo e que possui extensa pesquisa acadêmica e aplicação prática realizada. Ele se diferencia de outro métodos de fatorização pela não-negatividade dos dados, se opondo à Decomposição em Valores Singulares (*Singular Value Decomposition - SVD*), por exemplo. O NMF tem como objetivo a extração de *features* e a redução de dimensionalidade. Usualmente, o NMF é comparado à utilização da Análise de Componentes Principais (*Principal Component Analysis - PCA*), pois ambos atuam na redução de dimensionalidade. Entretando, o NMF se diferencia desse último por ser aplicado em dados não-negativos e em dados esparsos. Esse algoritmo tem sido amplamente aplicado em pesquisas da área médica, mas necessitando de adaptações: o

fato do NMF ser um algoritmo não-supervisionado introduz algumas limitações em situações onde a interpretabilidade é muito relevante e crítica, como em pesquisas biomédicas. Dessa forma, tem sido ultimamente aplicada uma técnica de mascaramento durante o processo de decomposição do algoritmo NMF (Lin and Boutros, 2020).

Além disso, aplicar o NMF implica em resolver o problema de encontrar W e H, de forma que $V \approx WH$. Essa aproximação é medida com a aplicação de uma função objetivo, como a *distância euclidiana*. Neste estudo, pretendemos usar a **divergência de Kullback-Leibler** (divergência KL) como medida do erro de reconstrução, por ser um das funções objetivo mais usadas com o NMF (Hien and Gillis, 2020), e também possui uso evolutivo em soluções de *Deep NMF*, como o trabalho utilizado com base de estudo para o desenvolvimento do algoritmo *Deep NMF* dessa pesquisa (Nasser et al., 2021).

A divergência de Kullback-Leibler é dada por:

$$D(V||WH) = \sum(V \cdot \log\left(\frac{V}{WH}\right) - V + WH) \tag{2.18}$$

E a chave para o algoritmo NMF usando a divergência KL é encontrar as regras de atualização para W e H que diminuam a divergência KL a cada iteração, até que se estabilize. Portanto, aplicamos regras de atualização multiplicativas às matrizes W e H, que são:

$$W = W \odot \left(\frac{V}{WH + \epsilon} H^T\right) / \left(M_1 H^T\right) \tag{2.19}$$

$$H = H \odot \left(W^T \frac{V}{WH + \epsilon}\right) / \left(W^T M_1\right) \tag{2.20}$$

onde:

- $\odot$ denota a multiplicação elemento a elemento,
- M_1 é uma matriz com todos os elementos sendo 1,
- / denota divisão elementar,
- W^T denota a matriz transposta de W,
- ϵ é uma constante para evitar divisão por zero.

Essas regras de atualização são aplicadas de forma iterativa, começando com matrizes W e H iniciais aleatórias e não negativas, até que a mudança na divergência KL fique abaixo de um limite pré-determinado ou o número máximo de iterações seja alcançado. Esse processo de atualização das matrizes através de uma divergência é chamado de Atualização Multiplicativa (MU).

2.2.3 *Deep Non-Negative Matrix Factorization*

Como citado no tópico anterior, o NMF possui algumas limitações de aplicação, como a necessidade de adaptar o algoritmo em diversos componentes para aplicação prática; não produzir bons resultados em conjuntos de dados pequenos; ter dificuldade de lidar com conjuntos de dados poucos esparsos, ou seja, que possuem grande proporção de conteúdo, o que deixa a matriz V menos esparsa, quando se considera a estrutura do NMF com uma única camada.

Para isso, foram desenvolvidas versões do NMF, utilizando redes neurais, para usar uma estrutura multicamadas, o que caracterizou os algoritmos chamados de ***Multi-layer NMF***. Eles se caracterizam por combinar NMF e redes que derivam as iterações das matriz base (W) e da matriz de features (H) em várias camadas e então fatoriza a matriz H para construir a estrutura da rede (W. Chen, 2022). Usualmente, múltiplas camadas são utilizadas, através de métodos de gradiente descendente ou propagação retroativa, para penalizar ou premiar acertos ou erros do algoritmo através das camadas, corrigindo quando necessário, gerando um melhor e mais refinado processo de aprendizado (W. Chen, 2022).

Entretanto, ainda existe uma dificuldade em se construir uma estrutura hierárquica ótima para esses problemas. Nesse ponto, o aprendizado profundo possui a capacidade de dividir uma tarefa complexa em tarefas menores, extraindo informação das matrizes W e H para resolver problemas de classificação (Wang and Zhang, 2023).

Como consequência disso, têm sido utilizadas diversas formas iterativas do NMF adaptados para soluções localmente eficientes. E isso foi um ponto de partida para a utilização de aprendizado profundo (***deep learning***), onde se transformam as iterações do algoritmo em camadas de uma rede neural sendo chamado de *Deep NMF*. Pode ser dito que o *Deep NMF* é o *Multi-layer NMF* com o uso do Ajuste Fino (*Fine-tuning*), onde os pesos de um modelo pré-treinado são transferidos para um teste em dados novos (Flenner and Hunter, 2017).

Assim, a partir disso, se desenvolveram técnicas para aplicar o NMF juntamente com aprendizado profundo seja em modelagens supervisionadas, não-supervisionada ou para aplicação em *PUL* (Nasser et al., 2021). Dentre as evoluções do NMF no sentido de formação de redes neurais, as vertentes *"Deep"*, podemos citar as seguintes partições: *Deep NMF*, *Constrained Deep NMF*, *Generalized Deep NMF*, *Multi-view Deep Matrix Factorization*, a associação entre NMF e redes neurais profundas (W. Chen, 2022), Deep Unfolding NMF (Nasser et al., 2021).

Entretanto, todas as formas de aplicação do Deep NMF possuem uma estrutura básica. Como definido na Seção 2.2.2, as matrizes W e H são obtidas a partir da matriz V, de entrada. Isso compõe a primeira camada do processo de aprendizagem. Para formar a

estrutura hierárquica da rede neural, a matriz H, agora chamada de H_1 decompõe para formar as matrizes W_2 e H_2, formando assim uma estrutura de duas camadas. Replicando essa lógica até o número máximo de camadas, obtém-se uma estrutura multicamadas, onde a matriz de entrada V decompõe em $L+1$ fatores, conforme a **Figura 2.1** e as equações a seguir, extraídas de W. Chen (2022):

$$V = W_1 W_2 W_3 ... W_L H_L \tag{2.21}$$

onde $i = 1, 2, 3..., L$, e:

$$H_2 \approx W_2 W_3 ... W_L H_L; H_3 \approx W_3 ... W_L H_L; H_{L-1} \approx W_L H_L \tag{2.22}$$

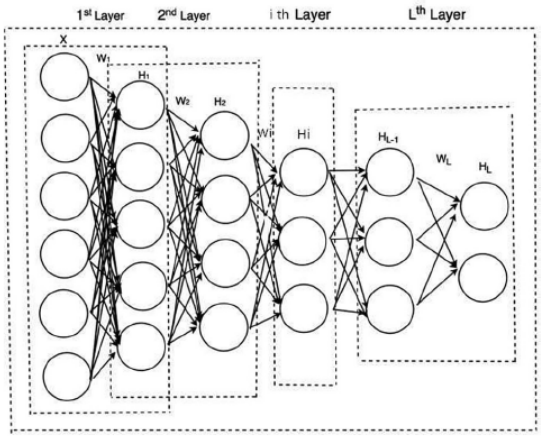


Figura 2.1: Estrutura de rede de uma abordagem básica do Deep NMF . Fonte: Figura retirada de (W. Chen, 2022)

Esse passo pode ser considerado como a etapa de pré-treinamento do modelo de aprendizado profundo. Nos algoritmos de camada única que utilizam o método NMF, algumas funções-objetivo são utilizadas para otimização do problema, buscando minimizar o erro e atingir uma convergência, como os algoritmos de *Atualização Multiplicativa*. Na estrutura multicamadas também é necessária a aplicação de técnicas de otimização, chamado de estágio de *Ajuste Fino*, como pode ser observado na função objetivo, extraída de W. Chen (2022):

$$D = \frac{1}{2}\left\|V - W_1 W_2 ... W_L H_L\right\| \tag{2.23}$$

2.2.4 Método de desdobramento para o NMF

No contexto de aplicação do *Deep Unfolding NMF*, diversas técnicas de desdobramento (unfolding) foram desenvolvidas visando a aplicação de um *framework* de deep learning

para o NMF. Essas técnicas provém uma conexão entre algoritmos iterativos e redes profundas (Monga et al., 2021). O *framework* para este tipo de abordagem realiza o desdobramento dos passos iterativos do algoritmo em camadas da rede profunda, o que permite transformar de maneira direta um algoritmo iterativo de aprendizado em um método de aprendizado através de uma rede neural (Hershey and Weninger, 2014).

Muitas aplicações do NMF utilizam um esquema onde, alternadamente, uma das matrizes W ou H é mantida fixa, enquanto otimiza a outra matriz, na decomposição $V \approx WH$, usualmente aproximado através de Atualização Multiplicativa (MU). Algoritmos iterativos geralmente possuem convergência lenta e alto custo computacional quando aplicados a matrizes com altos valores de dimensões (Kim and Park, 2011). Dessa forma, uma abordagem com *unfolding* no NMF em um aprendizado profundo pode proporcionar ganhos de desempenho, além de evitar altos custos computacionais.

Através do contexto de aplicação de técnicas de desdobramento para tornar o uso do NMF em uma rede de várias camadas, visando resolver problemas de classificação, Nasser et al. (2021) propôs um método de *unfolding* para o NMF, levando a um algoritmo de aprendizado profundo, o Deep NMF, para aplicação em problemas de dados PU.

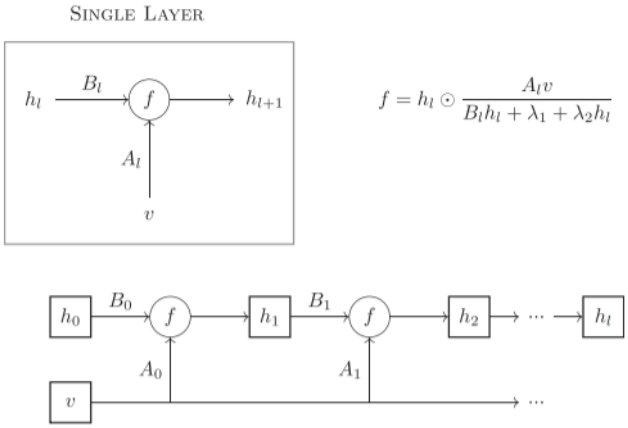


Figura 2.2: Estrutura de uma rede com desdobramento para o NMF
. Fonte: Figura retirada de (Nasser et al., 2021)

Em uma visão conceitual, o algoritmo se configura com a otimização de uma coluna h, da matriz H, enquanto permite-se variar a matriz W, sendo parte dos parâmetros da rede que serão aprendidos. Verifica-se, a partir da Fig 2.2, que h é formado por uma transformação dos valores da camada anterior, onde $A = W^T$ e $B = W^T W$, e v são dados de entrada, sendo f a função de atualização de cada camada, conforme Eq.2.24, extraída de Nasser et al. (2021).

$$h_{l+1} = h_l \odot \left(\frac{W^T V}{W^T W h_l + \lambda_1 + \lambda_2 h_l} \right) \tag{2.24}$$

O modelo de Deep NMF é construído com múltiplas camadas de unidades de aprendizado não supervisionadas, cada uma responsável por uma etapa de atualização multiplicativa orientada pelas normas de Frobenius. Essas camadas trabalham em conjunto para refinar de forma iterativa as representações dos documentos e as suas características, com a adição dos termos de regularização L1 e L2 para impor esparsidade e suavidade, respectivamente. Essa regularização busca mitigar o sobreajuste e aprimorar a capacidade do modelo de generalizar para novos dados não vistos.

No processo de treinamento da rede, a matriz H é atualizada iterativamente, a partir dos dados da matriz V e da própria matriz H. Um método estocástico de gradiente de descida baseado na estimativa adaptativa é aplicado na rede, mais especificamente, através do otimizador *Adam*. Então, executa-se um ajuste de mínimos quadrados não negativos (*NNLS*) para cada recurso, atualizando os pesos com base na saída atual e na matriz de entrada V. Com isso, obtém-se como resultado do método a matriz atualizada final H, obtida através do processo iterativo, e a matriz aprendida W, sendo essa a matriz de pesos ajustada.

Capítulo 3

Revisão de Literatura

3.1 Abordagem da Revisão Sistemática de Literatura

A grande quantidade de problemas relacionados à necessidade de se utilizar bases de dados onde a maior porção dos dados não estão rotulados se traduziu em motivos para se realizar uma pesquisa de classificação de textos a partir do PUL.

Assim, realizou-se uma revisão sistemática de literatura, visando extrair diversas informações qualitativas e quantitativas relativas ao assunto objetivo desse estudo. A escolha de utilizar uma abordagem de revisão sistemática de literatura foi baseada no objetivo de se extrair e sintetizar os estudos científicos de boa qualidade no tópico descrito anteriormente. Essa definição pode ser chamada de evidência (Kitchenham et al., 2009). Assim, a revisão sistemática do presente estudo deve ser baseada em evidências de bons artigos, estudos, livros e periódicos científicos para classificação de textos a partir de PUL.

A presente revisão sistemática de literatura foi realizada tomando como base a revisão da literatura existente, sumarizando artigos que servem de base para justificar o estudo e também para identificar lacunas no estado da arte, e também uma revisão independente, com o intuito de extrair dados e estatísticas para melhor delimitar os artigos da revisão de literatura.

Dentro do contexto de revisão independente, foram feitos dois tipos de revisão com propósito descritivo. O primeiro tipo é a síntese narrativa de texto, onde dados são extraídos dos artigos, determinando as características relevantes de cada um, organizando as análises do estudo em subgrupos. E o outro tipo de revisão aplicada foi a de revisão de escopo, tendo como foco a extração da maior quantidade de dados relevantes o quanto possível (Xiao and Watson, 2019).

3.2 Orientação da Pesquisa

Os protocolos que serão utilizados para orientar este estudo serão composto pelos itens: Questões de Orientação da Pesquisa; Bases para busca de artigos; Critérios de Seleção; Critérios de Qualidade; Estratégia de extração de informações e Segmentação de artigos.

3.2.1 Questões de orientação da pesquisa

A pesquisa foi orientada a partir dos questionamentos levantados, explicitados a seguir, onde existe uma questão principal e questões secundárias que nortearão o estudo.

Questão primária: Quais os métodos promissores para evolução da classificação semi-supervisionada de dados textuais considerando a utilização de métodos de PUL?

A partir da questão principal, questões secundárias foram definidas visando especializar a pesquisa.

Questões secundárias:

- Quais métodos de Classificação de texto, de forma geral, são utilizados?
- Quais métodos de PUL (*Positive Unlabeled Learning*) são usados?
- Quais metodologias de redução de dimensionalidade em texto são usadas?
- Existem abordagens de aprendizado profundo e redução de dimensionalidade aplicadas a problemas de PUL?
- Quais métodos de PUL (*Positive Unlabeled Learning*) obtém melhores resultados?
- Quais são as bases de dados mais utilizadas?

3.2.2 Bases para busca de artigos

Os critérios de busca dos artigos das bibliotecas mais importantes para a área de conhecimento que está sendo pesquisada foram os definidos a seguir.

- **Bases de dados:** Scopus [1], Web of Science [2], ACM Digital Library [3], IEEE Xplore [4], Periódicos CAPES [5], SpringerLink [6], Science Direct [7].

[1] https://www.scopus.com/
[2] https://clarivate.com/webofsciencegroup/solutions/web-of-science/
[3] https://dl.acm.org/
[4] https://ieeexplore.ieee.org/Xplore/home.jsp
[5] https://www-periodicos-capes-gov-br.ez54.periodicos.capes.gov.br/index.php/acervo/lista-a-z-periodicos.html
[6] https://link.springer.com/
[7] https://www.sciencedirect.com/

- **Ano de Publicação dos artigos:** 2018 - 2024
- **String de busca:** (“positive unlabeled learning” OR “positive unlabelled learning” OR “positive and unlabeled learning” OR “positive and unlabelled learning” OR “pu learning” OR “positive unlabeled” OR “positive unlabelled”) AND (“text” OR “textual” OR “text classification”)

Deve-se levar em conta que as buscas nas bases de dados foram feitas aplicando a *string* de busca nos campos de título, resumo e palavras-chave, quando a motor de busca ofertava essa opção. Na base de dados IEEE Xplore, a busca foi realizada em todo o conteúdo textual do artigo, retornando um volume maior de artigos, mas sendo a maioria desses com tema ou foco distinto do relativo a esse estudo.

3.2.3 Critérios de seleção

O primeiro critério de seleção se baseia em, após utilizar a *string* de busca nas bases de dados, e observando o ano de publicação, realizar uma leitura do título, resumo e palavras-chave do artigo e tentar identificar se o artigo é atinente ao foco dessa pesquisa. Caso a leitura inicial não seja suficiente para averiguar a concordância com o tema, é realizada uma leitura superficial do processo metodológico, experimentos e resultados obtidos. Se esses passos não resultassem em uma adequação do artigo com o tema da pesquisa, o artigo é rejeitado. Caso a sentença anterior seja afirmativa, o artigo deve passar pelos critérios de exclusão a seguir, para que seja incluído no rol de artigos selecionados. Os artigos que estejam de acordo com, ao menos, um dos critérios abaixo, não são selecionados.

- Publicação pelo(s) mesmo(s) autor(es) em outra publicação: título e resumo similares;
- Publicação escrita em idioma diferente do inglês;
- No caso de publicações duplicadas, manter apenas uma;
- Publicação que não usa explicitamente o PUL ou que não avalia qualquer método de PUL;
- Publicação que não utiliza datasets de texto;
- Publicação de data anterior ao critério de ano de publicação da subseção anterior.

3.2.4 Critérios de qualidade

Após a aplicação dos critérios de seleção de artigos, foi realizada a aplicação de alguns critérios de qualidade, visando identificar, dentre os artigos selecionados, aqueles que têm

maior atinência com o estudo a ser realizado. Os quesitos de qualidade levantados têm como objetivo indicar quais das publicações produzem estudos mais aprofundados e com mais detalhes de avaliação. Os critérios são descritos abaixo:

- A publicação compara diferentes algoritmos ou abordagens de PUL;
- A publicação utiliza mais de um dataset de texto para experimentos;
- A publicação utiliza mais de uma métrica de avaliação;
- A publicação não foca em *One-class classifiers*.

3.2.5 Estratégia de extração de informações

Concomitantemente à aplicação dos critérios de qualidade, onde uma leitura completa do artigo foi realizada, houve a extração de diversas informações indispensáveis para responder os questionamentos levantados que orientam a presente pesquisa. As informações que foram definidas como importantes para o processo de extração se basearam na percepção de que muitos estudos e experiências de pesquisas realizadas são basilares para suportar um novo estudo e também para inspirar novas abordagens, assim como para evitar o desenvolvimento de trabalhos já consolidados. Dessa forma, as informações elencadas abaixo serão importantes para melhor definir o problema a ser estudado e também para melhor orientar a metodologia e métricas e métodos de avaliação.

- Título da publicação;
- Ano da publicaçao;
- Idioma da publicação;
- País da publicação;
- Jornal da publicação;
- Domínios da aplicação;
- Nomes das bases de dados;
- Idiomas das bases de dados;
- Tamanho das bases de dados;
- Qual a representação de texto utilizada?
- Qual a quantidade de dados rotulados utilizada nos experimentos?
- É aplicado alguma técnida de redução de dimensionalidade?

- Qual o método de classificação utilizado?
- Qual o técnica de PUL é utilizada?
- Qual(is) a(s) métrica(s) de avaliação é(são) utilizada(s)?
- Qual o método de avaliação utilizado?
- Trabalhos futuros.

3.2.6 Categorização de artigos

A partir de todas as informações e da base de artigos selecionados, foi construída uma categorização das publicações baseada no tipo de metodologia de *Positive Unlabeled Learning* que foi aplicada. As definições das categorias foram definidas no capítulo de Fundamentação Teórica e refletem a abordagem relativo à rotulação dos dados e de tratamento dos mesmos. Foram formadas 5 categorias, podendo haver interseção entre os grupos.

1. Artigos que focam em *one-class classifiers*;
2. Artigos que aplicam métodos de redução de dimensionalidade;
3. Artigos que usam o método PUL *Two-step techniques*;
4. Artigos que usam o método PUL *Biased Learning*;
5. Artigos que usam outros métodos PUL como *Generative Adversarial Network* - GANs, Incorporação da Classe principal.

3.3 Resultados da revisão sistemática

Consolidando todos os dados resultantes das ações e critérios adotados, descritos na seção anterior, podemos extrair diversas informações relevantes para o estudo atual, como dados estatísticos de algoritmos mais utilizados e domínios mais abordados nas publicações.

Inicialmente, a Figura 3.1 demonstra os passos de busca, filtro e seleção de artigos, detalhados na Seção 3.2.

No processo de busca, foram encontrados um total de 1547 artigos, seguindo o padrão da string de busca definida na Subseção 3.2.2. Entretanto, apenas 445 possuiam assunto, de acordo com título e resumo dos artigos, aderente a atual linha de pesquisa. Logo, dos 1547 artigos resultantes da string de busca, 1102 artigos foram rejeitados. De acordo com os critérios de seleção, outros 327 artigos foram rejeitados e 118 foram selecionados, sendo 37 desses, entretanto, duplicados, sendo assim removidos da base de artigos final. Adicionalmente, foram incluídos como aceitos 6 artigos do tipo *survey*, que serão utilizados

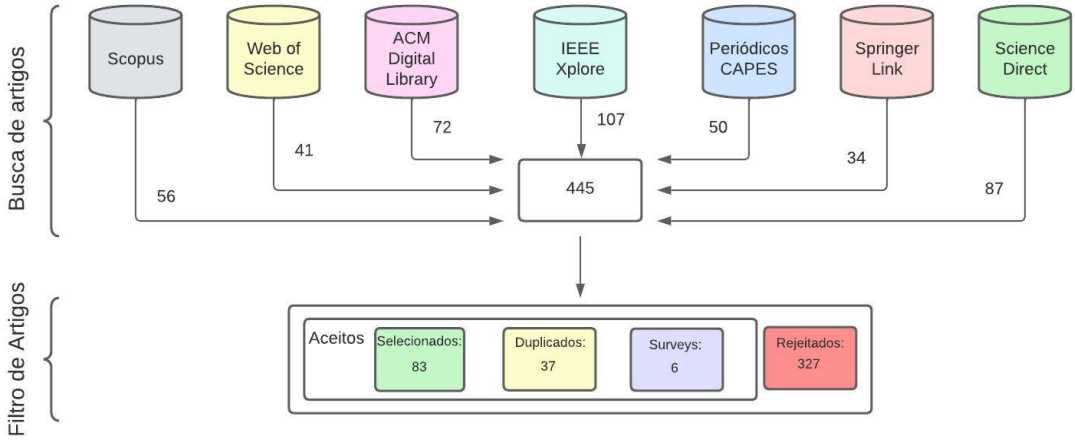


Figura 3.1: Organograma de seleção de artigos

para fins de fundamentação teórica e verificação das melhores práticas e também das técnicas mais aplicadas nos últimos anos.

A partir da base de 83 artigos selecionados, foi feito uma extração dos dados, conforme critérios da **Subseção 3.2.5**. A extração foi feita realizando-se uma leitura completa dos artigos e, assim, armazenando todos os dados em uma base de dados. Destaca-se ainda que juntamente com os dados extraídos a partir dos critérios de qualidade (Seção 3.2.4), uma base de artigos destacados foi formada, sendo esses artigos alvos de estudos mais detalhados para a produção metodológica para o presente trabalho. De posse dos dados da base selecionada de 83 artigos, foram feitas diversas análises quantitativas e qualitativas, visando responder a questão primária e as questões secundárias. Estas análises são apresentadas nas subseções a seguir.

3.3.1 Análise temporal e geográfica

Sendo orientado pelo critério levantados na Subseção 3.2.2, onde os anos de seleção dos artigos devem estar dentro do período de 2018 a 2024, temos os resultados obtidos na **Tabela 3.1** e pela **Figura 3.2**.

Em uma análise temporal, podem ser destacadas as produções de artigos nos anos de 2020 e 2021, 18 e 19 respectivamente, mesmo em um período de pandemia. Já no ano de 2024 a quantidade de 04 artigos, deve-se ao fato desta análise ter sido feito em meados de julho de 2024.

Analisando geograficamente a produção de artigos com base nos países, verifica-se que a China possui a maior quantidade de artigos, sendo responsável por mais de um terço

Ano	Quantidade de Artigos
2018	10
2019	09
2020	18
2021	19
2022	13
2023	10
2024	04

Tabela 3.1: Artigos selecionados por ano de produção.

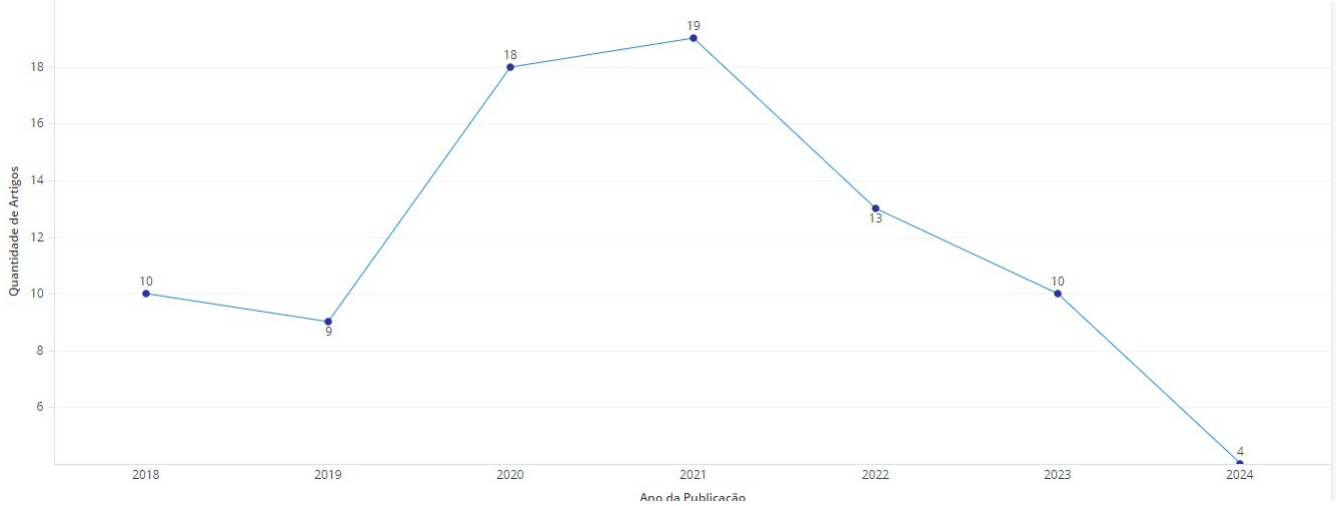


Figura 3.2: Produção de artigos por ano

(*37,35%*) dos artigos selecionados. Após a China, os países que mais produziram artigos foram os Estados Unidos da América, com 12 artigos (14,46%), Brasil, com 7 artigos (8,43%) e Austrália, com 6 artigos (7,23%). Pode-se concluir também que China e Estados Unidos da América, juntos, correspondem a mais da metade dos artigos selecionados (51,81%), conforme verificado na **Figura 3.3**.

No ano de 2018, pela primeira vez, a China se tornou o país com maior produção de artigos científicos do mundo, passando a produção realizada pelos Estados Unidos da América (Tollefson, 2018). Isso demonstra que outros centros estão crescendo nas produções científicas, trazendo maior diversidade de ideias e investimento no que tange a desenvolvimento na ciência de tecnologia. Isso é comprovado com a amostra de artigos selecionados nesta revisão sistemática de literatura.

3.3.2 Análise dos conjuntos de dados

Observando os dados extraídos da base de artigos selecionados no que se refere aos datasets e aos métodos de representação de texto utilizados, podemos tirar algumas conclusões.

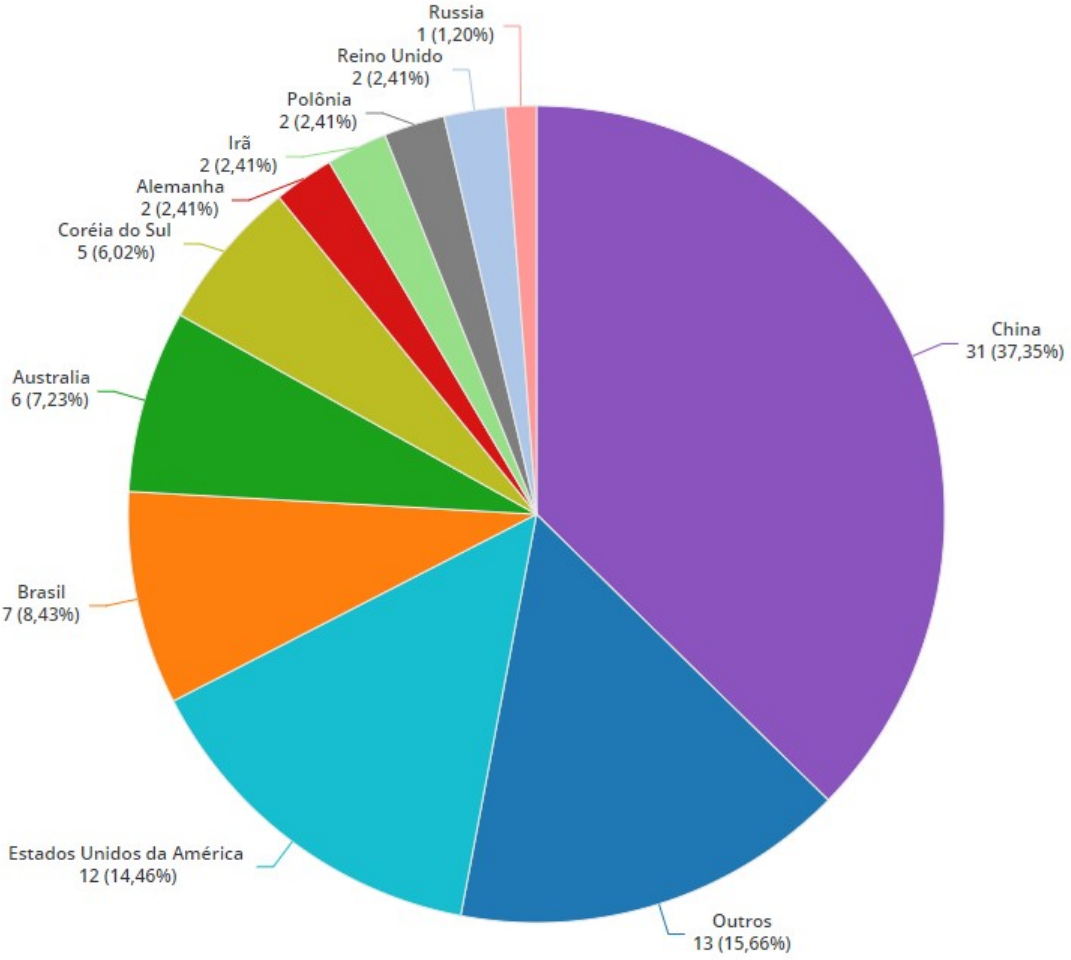


Figura 3.3: Produção de artigos por país

A **Figura 3.4** mostra a distribuição de idiomas das bases de dados utilizadas nas publicações. Percebe-se facilmente que a maior porção de dados utilizados estão na língua inglesa, podendo ser explicado pelo fato desse idioma ser tratado como idioma universal. Outros idiomas como chinês, português, alemão, espanhol, maltês e russo aparecem com pouca aplicação nos idiomas das bases de dados aplicadas. Mesmo a China possuindo a maior proporção de artigos selecionados, conforme **Figura 3.3**, muitos desses artigos utilizam bases de dados em língua inglesa para os seus estudos.

A partir dos conjuntos de dados utilizados no rol de 83 artigos selecionados, pode-se inferir que existe uma gama de conjuntos de dados grandes que são abordados quando se trata de classificação de texto e PUL. Percebe-se que os 5 primeiros conjuntos de dados da **Tabela 3.2** foram utilizados em experimentos em cerca de um terço dos artigos selecionados. Embora tenhamos a conclusão apontada anteriormente, é direto dizer que

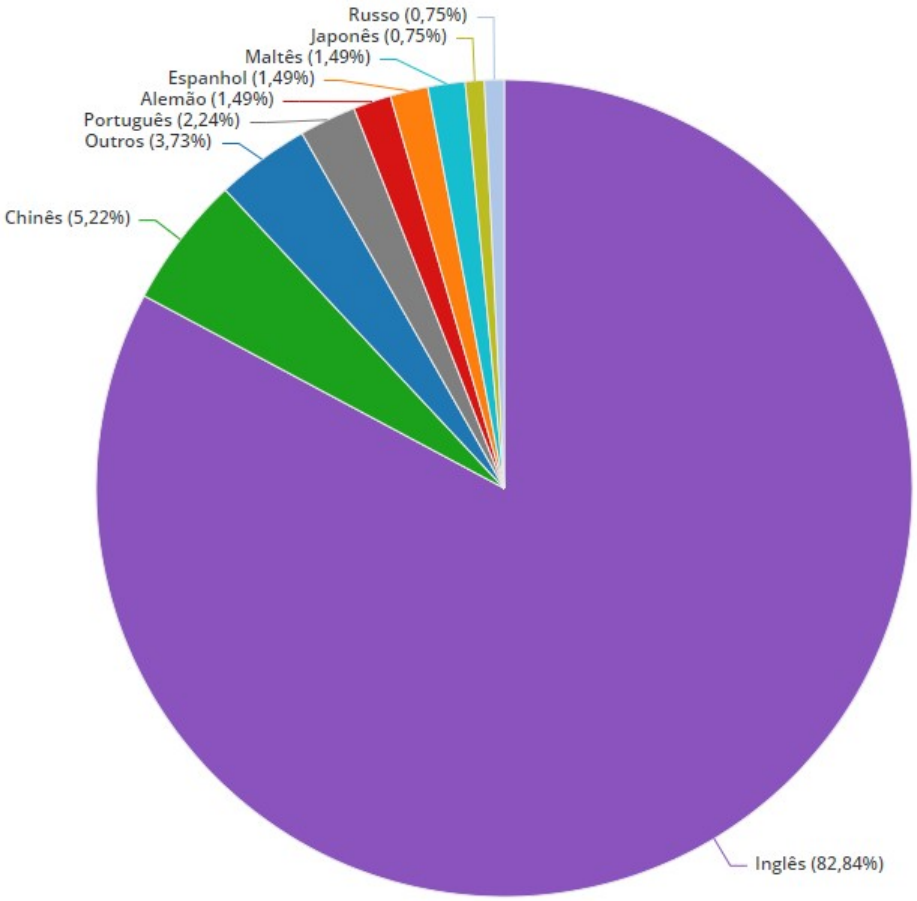


Figura 3.4: Idiomas das bases de dados utilizadas nos artigos

a variedade de assuntos de aplicações são refletidas pela variedade de conjuntos de dados e dos seus domínios de aplicação, como veremos posteriormente neste capítulo.

Assim, podemos verificar a existência de conjuntos de dados dos domínios mais variados, como: notícias, páginas web, bases de dados de artigos científicos, redes sociais, forums, bases de dados de avaliações de produtos e serviços, bases de documentos médicos, entre outros. E alguns desses domínios compõem alguns conjuntos de dados que são frequentemente aplicados em pesquisas, como dados de notícias em geral, o *20 Newsgroup*, que possui dados de diversas fontes de notícias, o Citeseer, que possui dados de produções científicas, o IMDB e *Wikipedia* que trazem dados de texto em páginas *web*.

Um outro fator a ser analisado neste tópico é a forma de representação dos dados textuais nos artigos selecionados. Como existe o foco de tratar de dados no formato de texto neste estudo, é importante entender como que as representações de texto são utilizadas nos diferentes contextos de domínios de aplicação.

A **Figura 3.5** mostra que os três primeiros grupos de representação de texto aparecem

Dataset	Quantidade de Artigos
News dataset	10
20 Newsgroup	7
Wikipedia	6
Forums dataset	5
Twitter dataset	5
Articles dataset	4
UCI dataset	4
CiteSeer	4
Cora	4
IMDB	4
Social networks dataset	3
Amazon reviews	3
Yelp reviews	3
Cora-ML	2
CSTR	2
Elec dataset	2
FEVER	2
RG-65	2
SimLex-999	2
Steam reviews	2
text8	2
WordNet	2
WordSim-353	2
YAGO	2
Outros	42

Tabela 3.2: Conjuntos de dados dos artigos

em mais da metade dos artigos, o que demonstra que são formas de representar o dado textual bastante utilizadas quando aplicadas em PUL. O método de representação *BoW* ou método TF-IDF aparecem em 21 artigos, sendo portanto praticamente utilizado em mais de 25% das publicações selecionadas, conforme pode ser notado na **Figura 3.6**.

Ainda pela **Figura 3.6**, mais de 70% dos artigos selecionados utilizam um desses métodos: *BoW*, *GloVe*, TF-IDF, Grafos, *Word2Vec* e BERT.

3.3.3 Análise dos domínios de aplicação

Os domínios de aplicação tem uma relação forte com as bases de dados utilizadas nos artigos. Alguns estudos aplicam diversas bases de dados de domínios distintos e outros estudos utilizam uma linha de pesquisa mais teórica, não especificando um aplicação em domínio, mas a maioria dos artigos seguem uma linha de aplicação em uma área pré-determinada.

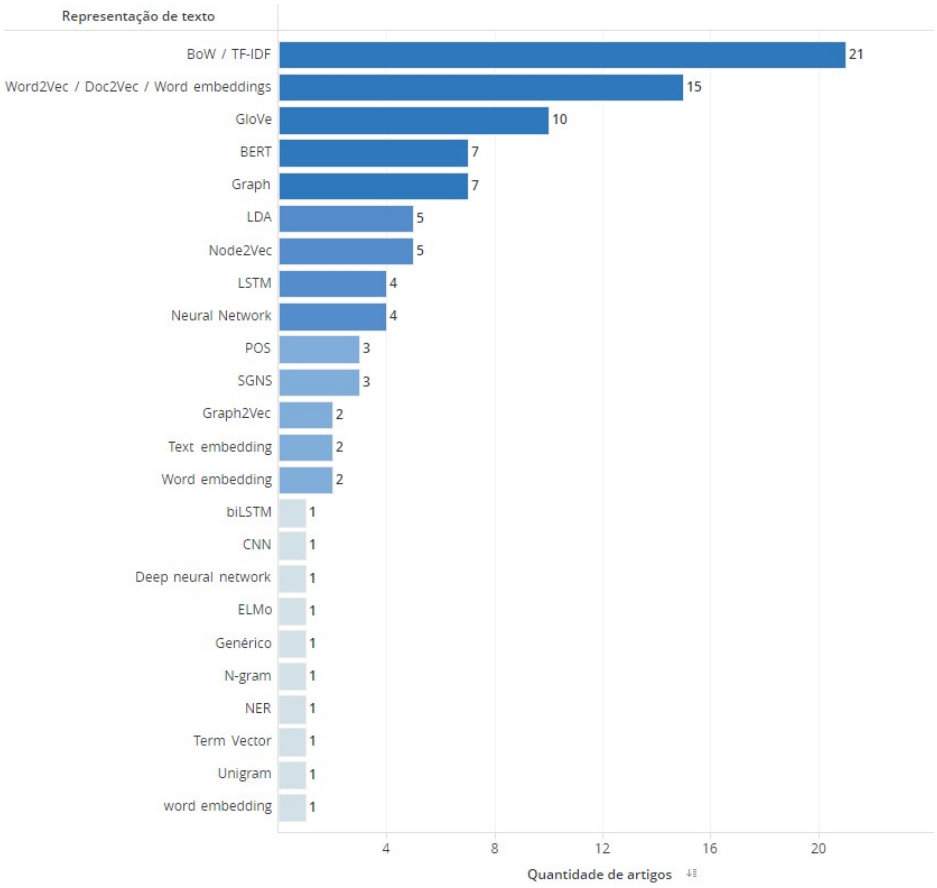


Figura 3.5: Métodos de representação de texto em quantidade

Na **Figura 3.7**, observa-se que muitos artigos em Aprendizado positivo e não-rotulado fazem estudos de forma mais teórica, buscando otimizar, de forma matemática, o desempenho de classificação de textos. Mas, ainda assim, a maioria das publicações selecionadas se atém ao domínio de aplicação, estudando maneiras de otimizar o uso das técnicas de PUL a um problema real e específico daquele domínio.

Domínio - Genérico

Jiang et al. (2018) analisa a representação de texto *word embedding* em idiomas com baixos recursos, idiomas onde a quantidade de pessoas que a falam é grande, mas o volume de conjuntos de dados disponíveis para treinamento de modelos em inteligência artificial ainda é pequeno. Nesse contexto, o autor aplica PUL para fatorizar a matrix de co-ocorrências, dado que a mesma é esparsa pelo fato de não se conhecer a co-ocorrência de muitos pares de palavras. Jungmaier et al. (2020) também aborda situações onde há volume de dados textuais menor, como em idiomas poucos usuais para modelos de aprendizados de máquina. Ele utiliza PPMI, *positive pointwise mutual information*, adicionando uma suavização de *Dirichlet* e compara com a abordagem através de PUL.

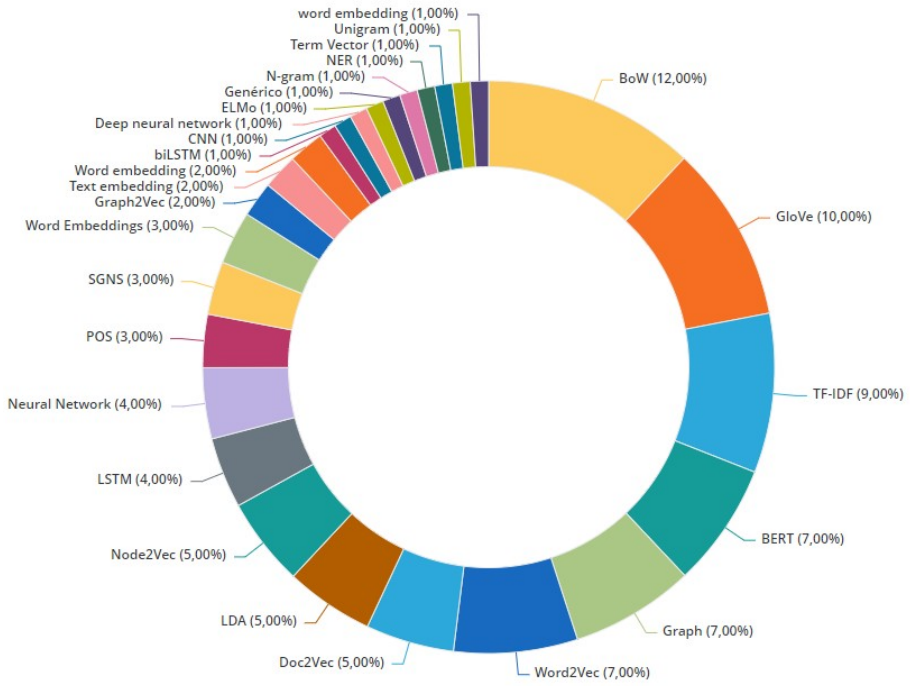


Figura 3.6: Métodos de representação de texto em porcentagem

Uma das motivações mais comuns ao se aplicar PUL para problemas de classificação é o de que no aprendizado supervisionado há risco de classificação errônea da classe principal. Existem então abordagens para avaliar numericamente o erro do classificador apenas com os dados disponíveis e assim otimizá-lo (Chen et al., 2020), podendo ser aplicado em qualquer domínio. Também são estudadas abordagens para lidar com risco de erros na classificação dos exemplos não-rotulados dos conjuntos de dados em positivos, quando forem negativos. Por isso, utilizam uma abordagem semi-supervisionada realizando a atualização dos pesos (*reweighting*) de forma supervisionada (Wang et al., 2022).

Existem abordagens para tratar de conjuntos de dados onde há múltiplas classes principais, ou seja, muitas classes são classificadas como positivas. Esse problema se chama de ***Multi-Class Positive Unlabeled Learning***, MPU, e é estudado em (Park, 2022). Nesse estudo, é proposto uma abordagem de 2 passos, conhecida como *two-step technique*, onde a primeira fase consiste em identificar aqueles exemplos negativos com maior probabilidade de o serem através da aplicação de alguma função de classificação, sendo aplicada no estudo em questão o método de KNN, *k-nearest-neighbors*. Ainda na linha de multi-classes, existe a linha de pesquisa na utilização de cadeias de classificadores, conhecido como *classifier chains*, de forma a se adaptar a um framework de um problema de PUL, já que a cadeia de classificadores transforma justamente um problema de multi-classes em vários problemas menores de rótulos binários, que é o problema que o PUL resolve. Os métodos CCPU e CCPUW aplicam essa metodologia (Teisseyre, 2021).

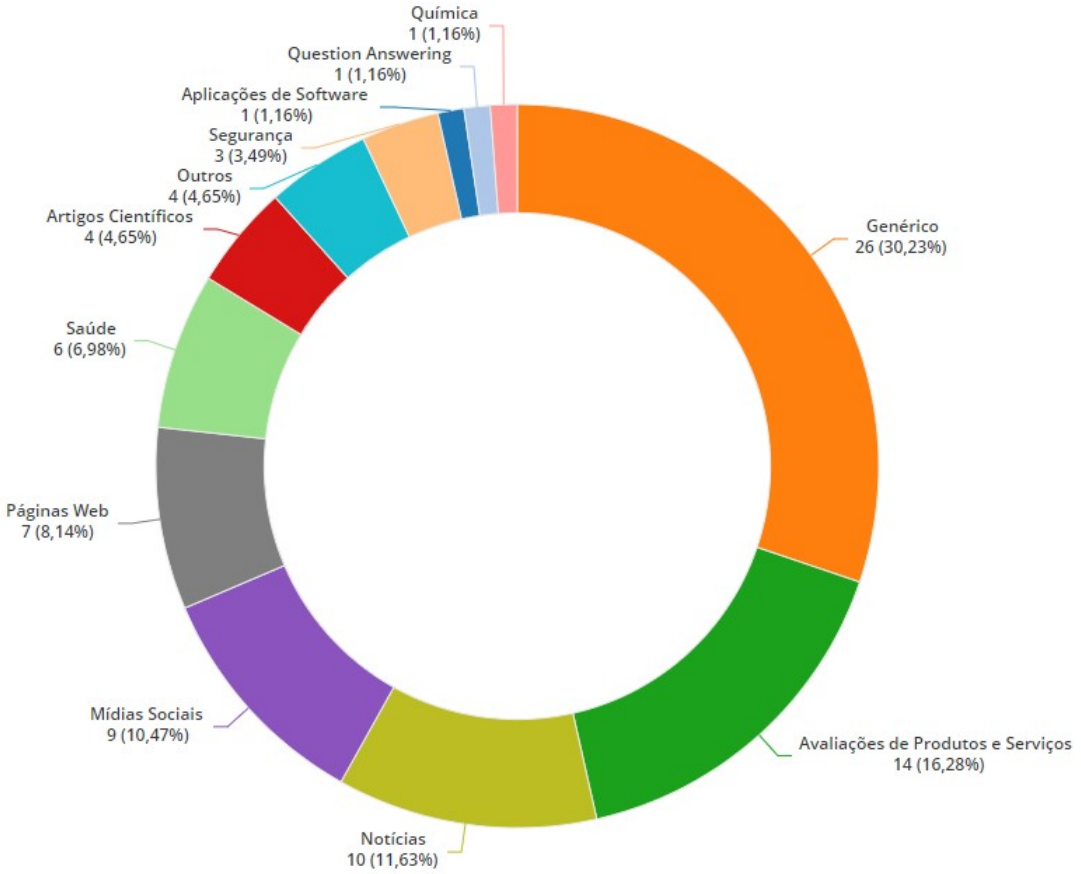


Figura 3.7: Domínios de aplicação dos artigos

Sendo o método de propagação de rótulos (*SCAR - selected completely at random*) o mais utilizado em estudos, outras linhas de pesquisa têm abordado o fato dele desconsiderar o viés de seleção em aplicações reais. Assim, o modelo *Generative PUL* introduziu uma linha de métodos sem utilizar o SCAR de forma pura para propagação de rótulos, sendo usada a função risco de estimação como forma de criação de exemplos virtuais para treinar o classificador (Na et al., 2020).

O campo de estudo em PUL está crescendo tão rapidamente que já é quase impraticável identificar o algoritmo mais indicado para uma dada tarefa. Em (D. Saunders and A. Freitas, 2020), se desenvolveu um modelo de aprendizado de máquina, *Auto-ML*, que seleciona o algoritmo mais indicado dado um conjunto de entradas e um conjunto predefinido de algoritmos. A técnica ***Biased SVM***, técnica comumente utilizada em problemas com dados positivos e não-rotulados, pode ser aplicada em ranqueamento de páginas ou em ranqueamento de documentos e assuntos (Pham et al., 2019). Em Liu et al. (2022a), a essência de métodos de *transfer learning* e PUL são combinados para transformar um classificador fraco em um classificador forte, através de uma otimização iterativa.

Mesmo em questões de processo de linguagem natural, existem diversas possibilidade

de aplicação de PUL, como em Zheng et al. (2021), onde se aplica o algoritmo *Spy-PU* para solucionar e reduzir o impacto negativo dos dados não-rotulados na aplicação de Reconhecimento de Entidade Mencionada (*Named Entity recognition*) em problemas de escassez de dados no tratamento da língua chinesa.

O estudo de Grafos de Conhecimento (*Knowledge Graph*) é amplo em diversas aplicações de inteligência artificial. Sendo assim, diversas lacunas e caminhos para otimizações surgem à medida que novas aplicações também são continuamente utilizadas. Algumas abordagens possuem limitações devido ao fato de que dependem de amostragem de casos negativos no conjunto de dados, o que nem sempre é possível. Assim, a aplicação da abordagem em 2 passos de PUL para identificar de forma automática e iterativa os negativos mais confiáveis da amostra tem sido uma das formas de se otimizar esses métodos (Shi et al., 2021) (Niu et al., 2018). Algumas iniciativas de solução envolvem aplicar o PUL quando não se tem informações suficientes para definir qual é a classe principal. O GRAB, *Graph-based Risk minimization with iterative Belief propagation*, utiliza essa premissa para treinar um classificador para, de forma eficaz, classificar os dados textuais sem conhecer a classe principal daqueles dados (Yoo et al., 2021).

Em Wu et al. (2021), desenvolve-se um algoritmo de redes neurais em grafos que é alimentado tanto pelas relações de nós próximos quanto a entre nós distantes entre si. Além disso, utilizar técnicas de PUL para aplicar um estimador de risco de propagação de rótulos no grafo, para que não haja grandes taxas de erro de rotulação de nós mais distantes.

Domínio - Avaliação de produtos e serviços

A avaliação de produtos, serviços e outros nas diversas páginas web e em aplicativos introduziram uma fonte constante e exponencial de geração de dados textuais. Entretanto, por vezes, as avaliações são usadas para promover um produto ou serviço de forma fictícia, também chamado de *spam* ou ainda, são apenas textos aleatórios que possui relação com aquele item. Para isso, diversos estudos voltados para a detecção de avaliações fictícias foram desenvolvidos e ainda são estudados atualmente. Dentre eles, Shuqin and Jing (2019) realiza a integração dos textos de comentários junto com comportamento do usuário para modelar um classificador, dado que os textos de comentários unicamente podem limitar a identificação de *fake reviews*. Assim, utilizou-se o algoritmo MPINPUL, *Mixing Population and Individual Nature PU learning*, que aplica o *K-means* para o exemplo negativo mais confiável, depois expande o conjunto de exemplos positivos a partir de outros exemplos negativos confiáveis (*reliable negatives* - RN).

Ainda no escopo de deteção de *spam*, ou avaliações falsas, em perfis de produtos ou serviços, (Zhang et al., 2018) enfatiza a importância de se aproveitar os dados que es-

tão rotulados no conjunto de dados, e não apenas focar naqueles dados não-rotulados. No estudo, é abordado um método parcialmente supervisionado, PSGD, que aplica PUL através da técnica de dois passos, realizando primeiramente a identificação de casos negativos confiáveis, aplicando posteriormente técnicas tipicamente supervisionadas, como *Naive-Bayesian* e *Expectation-Maximization* para treinar o classificador. Bian et al. (2021) também trilha o mesmo caminho, ao indicar a importância de se identificar os comentários fictícios dado que isso pode significar queda nas vendas de algum produto ou serviço, resultando em perdas financeiras para a empresa. Entretanto, no estudo é entregue como resultado a formatação de um framework para, mais rapidamente e com menos esforço, identificar avaliações reais ou fictícas de jogos em plataformas.

Em Jeffrey et al. (2020) é feita uma análise sistemática genérica das plataformas de avaliações de jogos e propõe um método para verificação geral dos textos. A partir do estudo de He et al. (2020), é possível treinar um classificador, *Biased-SVM*, através de técnica de PUL e densidade do comportamento do usuário e obter resultados satisfatórios. Já Wu et al. (2020) foca, dentro do problema de detectar avaliações fictícias em páginas de produtos, em resolver a diversidade de características que os usuários fictícios possuem ao inserir esses tipos de comentários nas plataformas. É proposto um método híbrido de detecção de *spam* baseado em PUL que considera tanto as características do usuário quanto a relação entre o usuário e o produto ou tipo de produto, injetando exemplos positivos no processo de treinamento do classificador para o mesmo conseguir apurar e identificar diferentes tipos de ações desses usuários.

Uma outra linha de estudos é, identificar dentro dos textos de comentários, aquelas palavras que possuem semântica de opinião e aquelas que possuem apenas contexto complementar no texto. Nesse quesito, além da abordagem tradicional, é necessária a abordagem de análise de sentimento. Wang et al. (2020) aborda esse problema a partir de PUL e análise de sentimento em duas etapas: no primeiro, através de análise de sentimento, identificar as palavras alvo do estudo, e no segundo passo separar as duas classes de palavras através de PUL.

Xu et al. (2019) utilizaram a técnica de **treinamento adversário** que pode ser definida como a realização do treinamento do classificador através de informações sabidamente erradas. Essa é uma técnica bastante usada em PUL, quando se deseja treinar o modelo com exemplos classificados de forma enganosa, visando adaptar o classificador para identificar falsos positivos. Xu et al. (2020) fez estudo na mesma direção, mas incorporou mecanismos de atenção e *Long short-term memory - LSTM*, avaliando o desempenho e efeito em *word embeddings*. Ambos os trabalhos citados em treinamento adversário aplicaram suas metodologias em conjuntos de dados textuais contendo avaliações, como: *Yelp, Rotten tomatoes, IMDB*. Uma variação do treinamento adversário é aplicar uma metodo-

logia preditiva visando também obter um bom *recall* ao se identificar exemplos negativos confiáveis através de PUL, e não apenas os exemplos positivos, como é feito ao se gerar exemplos positivos com o ***Generative Adversarial Networks - GANs*** (Hu et al., 2021).

Banerjee et al. (2018) aborda os conjuntos de dados formados de textos de avaliações de produtos e serviços a partir da identificação de casos de falhas de segurança do produto, problemas éticos, *recall* de produtos, etc. Abordam um método em 2 estágios, onde no primeiro escolhe casos representativos de ambas as classes e no segundo usa resultados do primeiro para executar uma classificação supervisionada. Yang et al. (2020) utiliza *GANs* para gerar dados de treinamento e identificar aqueles comentários que se relacionam com reclamações do produto, principalmente através de algoritmo de PUL. Já Gharahighehi et al. (2022) aplica PUL para resolver problemas de *cold-start* em sistemas de recomendação.

Domínio - Mídias Sociais

Egorov et al. (2020) propõe a construção de um classificador binário a partir da deteção dos relacionamentos dos usuários em uma rede social, identificando quem são os pais e filhos de cada relacionamento entre usuário e comentário, curtida, inscrição em grupo, etc. Entretanto, nesse estudo é utilizada uma abordagem do tipo orientada a uma classe, ou seja, onde se deseja identificar os objetos apenas de uma classe.

Em Zhang and S. Yu (2018), é estudado a composição de usuário em diversas redes sociais, ou seja, como o comportamento desse usuário pode ser descrito a partir das diversas ações em textos (comentários) nas múltiplas mídias sociais. Para isso, utiliza-se o método conhecido como *broad learning*, ou aprendizado amplo, visando agregar dados de redes sociais heterogêneas e treinar um classificador a partir disso. Muric et al. (2020) também se baseia na correlação de comportamento de um usuário em diversas redes sociais e, para isso, desenvolve um *framework* de simulação de agentes dirigido por dados, visando predizer o comportamento e as ações de um usuário a partir de alguma outra ação.

Por vezes, conjuntos de dados possuem múltiplas classes e, por isso, devem ser abordados a partir de técnicas de classificação que levem em conta esse fator. Liu et al. (2019) ataca esse problema a partir de multi-tarefas que vão, individualmente, entender cada par de classes, aplicando técnicas de PUL. Após isso, agrega os resultados que o classificador fornecer para identificar discursos de ódio, racismo, preconceito religioso em plataformas sociais.

Kaur et al. (2021) atua no seu estudo na detecção de contas de usuários que sejam maliciosas ou que atuem de forma fictícia nas redes sociais, o que descredibiliza a motivação de outros usuários utilizá-las. Os autores desse trabalho abordam essa questão a partir

dos algoritmos *One-class SVM*, *Isolation Forest*, ou seja, tratando como um problema de classificação unária.

Domínio - Notícias

Muitos estudos no domínio de notícias se relacionam com identificação de assuntos, detecção de fake news, categorização de opiniões em notícias, entre outros ramos. Wang et al. (2021) busca, em seu estudo, um classificador que tenha a capacidade de identificar, em um extenso conjunto de dados textuais de notícias, notícias que sejam relacionadas a um caso ou assunto em questão. Como notícias relacionadas a um assunto podem ser originárias de diversos periódicos, jornais, revistas diferentes com distintos estilos de escrita, pode ser complexo identificar os tópicos apenas através de técnicas tradicionais de modelagem de tópicos. Para isso, no estudo, foi proposta a utilização de PUL juntamente com modelagem de tópicos, especificamente com *variational autoenconder - VAE*, que extrai tópicos dos casos não-rotulados.

Certos trabalhos propõem uma variação de PUL que utiliza informação privilegiada, informação que é apenas disponibilizada durante a fase de treinamento do classificador, e também pesos de similaridade, pesos que são gerados para os casos não-rotulados de acordo com a similaridade com cada classe após a extração dos casos negativos mais confiáveis, como em Liu et al. (2022b). A adição dessas duas técnicas são importantes para auxiliar o classificador a lidar com uma proporção grande de dados não-rotulados.

No estudo de técnicas para identificação de notícias falsas, existem algumas abordagens sendo estudadas nos últimos anos. Gôlo et al. (2021) foca, no seu estudo, em identificar as melhores maneiras de se representar o dado textual e indica que esse passo do aprendizado é essencial para a construção de um bom classificador de notícias. A detecção de notícias falsas requer uma boa estruturação do conjunto de dados e de realizar uma **modelagem de tópicos** adequada. O seu estudo indica que técnicas de representação de textos unimodais, como o saco de palavras (*bag-of-words*), podem ser substituídas por abordagens multimodais. No seu estudo, aplica o *text embedding* e a extração de informações de tópicos dos textos, indicando que esse método dual auxilia na identificação de notícias falsas.

A abordagem metodológica utilizada por Caravanti de Souza et al. (2021) utiliza, primariamente, PUL com uma técnica de propagação de rótulos, chamada de **PU-LP**. O algoritmo principal identifica documentos (notícias) de interesse e aqueles que não são de interesse no conjunto de dados não-rotulados. Após, é utilizada a técnica de propagação de rótulos para rotular os documentos restantes. O estudo foi feito realizando diversas comparações: inicialmente é feita a comparação do PU-LP com algoritmo de PUL tradicional, o *Rocchio Support Vector Machine*, e com versões de algoritmos para aprendizado

de uma classe, como *k-Means, k-Nearest Neighbors, One-Class Support vector Machine*. Além disso, realizou o estudo comparativo performático de modelos de representação de texto, com *bag-of-words* e *Doc2Vec*.

Ainda como evolução do estudo e desenvolvimento do algoritmo **PU-LP**, Souza et al. (2024) utiliza mecanismo de atenção através do algoritmo GNEE (*Graph Attention Neural Event Embedding*), que integra regularização e mecanismo de atenção, para, a partir de um valor de até 10% de notícias falsas rotuladas, alcançar uma melhora no desempenho de classificação de notícias.

Domínio - Páginas Web

Ferretti et al. (2018) estuda técnicas de *positive unlabeled learning* e *one-classe learning* aplicada a predição de falhas de verificabilidade na página web *Wikipedia*. Dentre os tipos de falhas, o problema de artigos que não possuem citações suficientes para verificação foi utilizada como foco para a utilização de PUL e avaliação. Dentre as técnicas aplicadas estão o ***Biased-SVM*** e ***centroid-based balanced SVM***.

Muitas empresas que possuem vendas pela internet, sempre buscam aumentar as suas receitas através da venda de mais produtos e serviços e também manter os clientes comprando na sua loja. Uma das estratégias mais comuns e efetivas e realizar a venda cruzada (*cross-selling*), técnica de venda que busca aumentar a receita adquirida de cada cliente e diminuir a taxa de *churn* dos clientes, ou seja, reduzir a perda de clientes. Entretanto, essa estratégia deve ser utilizada tendo como alvo aqueles clientes certos que possuem propensão a comparem mais produtos ou serviços. Essa é uma estratégia de negócios bastante aplicada na indústria de seguros, entretanto os dados de vendas dessas empresas geralmente possuem uma pequena cesta de produtos, são dados muito desbalanceados, pois a maioria dos clientes não possuem informações de compra cruzada (*cross-purchase*), e não há informação relevante e confiável suficiente para identificar se os clientes não são propensos a realizar a compra cruzada ou se apenas não houveram oportunidades. Sidorowicz et al. (2022), em sua pesquisa, aplica técnicas de PUL em conjunto com **modelagem de tópicos** e ***feature engineering*** sobre dados não-estruturados de perguntas e respostas de clientes e dados demográficos. O método utilizado é o de, iterativamente, identificar dados da classe positiva dentro de amostras do subconjunto de dados não-rotulados.

Carvalho et al. (2018) alcança bons resultados tanto em termos de desempenho quanto em eficácia ao utilizar algoritmos de Maximização da Expectativa, *Expectation-Maximization (EM)*, para a detecção de páginas web que são réplicas ou muito similares a outras, o que pode gerar uma experiência ruim do usuário ao realizar pesquisas em sites de busca. Para isso, são utilizados exemplos não óbvios de réplicas para treinar o classificador, tornando-

o eficiente ao lidar com exemplos reais, o que o adequa ao escopo de dados positivos e não-rotulados.

Domínio - Saúde

Alguns estudos buscam aplicar os conceitos de PUL em assuntos ligados à saúde, como na bioinformática, documentos médicos, entre outras áreas. Abed et al. (2019) reúne Aplicações em Computação de Alta Performance (HPCA) com dados textuais biomédicos com o intuito de eliminar ambiguidades na mineração de textos médicos. Os autores utilizam a técnica de PUL chamada de ***Positive Unlabeled Disease gene Identification - PUDI*** na construção do classificador que irá classificar textos médicos e eliminar as ambiguidades na detecção de doenças através dos genes.

Yang et al. (2018) propõe como metodologia da sua pesquisa a construção de um framework, chamado de AdaSampling, *Adaptative Sampling*, para aplicação em PUL e em conjuntos de dados com erros de classificação de rótulos, ou comumente chamado de *label noise*. Os autores aplicam a metodologia em obstáculos existentes na bioinformática.

Jacovi et al. (2021) define que problemas de recuperação de documentos de classe já conhecida a partir de uma coleção de documentos não-rotulados podem ser considerados como problemas de PUL. Os autores aplicam soluções de redes neurais em PUL para conjuntos de dados de publicações médicas, como o *PubMed*, para demonstrar a eficiência do método proposto.

3.3.4 Análise dos algoritmos

Nesta seção, a análise será realizada sobre as técnicas de classificação utilizadas, as abordagens de métodos de PUL aplicadas, as publicações que utilizam alguma técnica de redução de dimensionalidade e artigos sobre o método *Deep NMF*.

Abordagens de PUL

Em relação às **abordagens de PUL** utilizadas e desenvolvidas na base de artigos da revisão sistemática de literatura, pode-se concluir que muitas técnicas são elaboradas a partir da adaptação de outras técnicas, concedendo assim uma denominação a essas técnicas. Algumas técnicas se destacam pela sua utilização em diversas publicações, indicando técnicas consolidadas e com base metodológica, de testes e de experimentação:

- nnPU - *Non-Negative Positive Unlabeled*
- uPU - *Unbiased Positive Unlabeled*
- GenPU - *Generative Positive Unlabeled*

- PUL-LP - *Positive Unlabeled Learning - Label Propagation*
- RC-SVM - *Rocchio Support Vector Machine*

Outras técnicas possuem novas denominações, visto serem desenvolvidas a partir da ótica de metotologia do autor da publicação referida, como: PE-PUC (*Positive Unlabeled Document Enlarger*), PUL-LELC (PUL através da extração de exemplos positivos e negativos confiáveis e formação de micro-clusters).

Métodos de classificação

A partir da definição de revisão de escopo, relatada no início deste capítulo, foi possível extrair os dados relativos aos **métodos de classificação** utilizados nas publicações da base de artigos selecionados, quando esses fossem utilizados como parte da metodologia de PUL ou quando eram usados para fins de comparação de desempenho entre algoritmos.

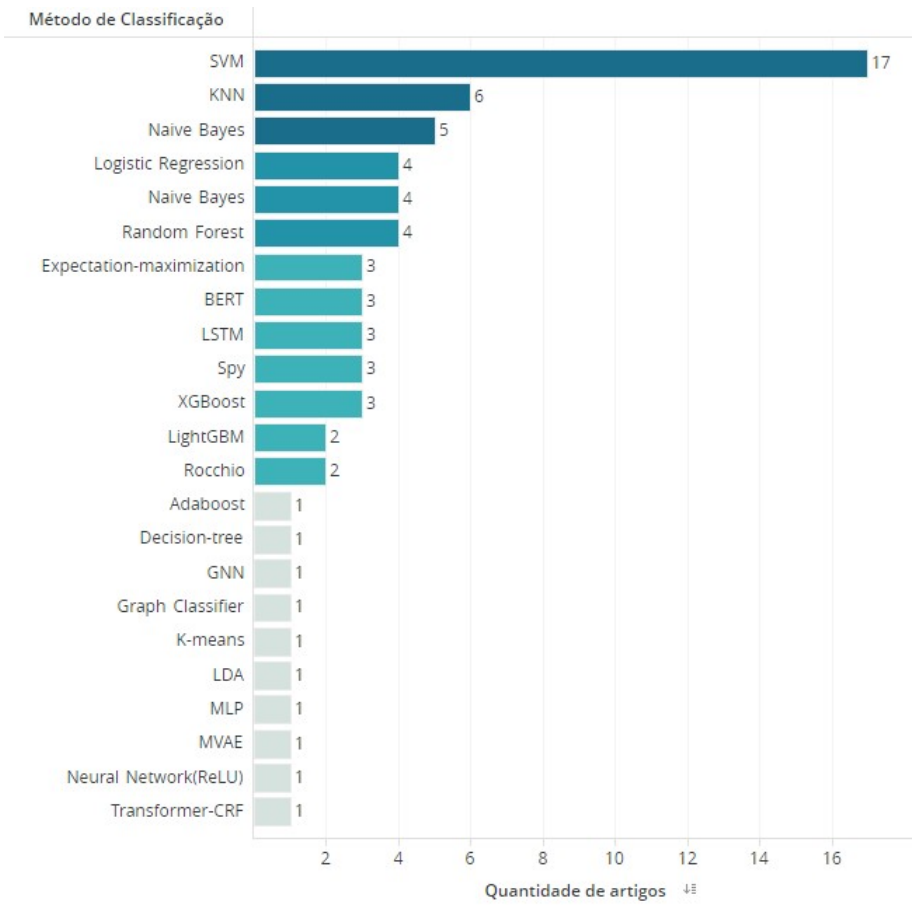


Figura 3.8: Métodos de classificação em quantidade de artigos em que foi aplicado

Alguns métodos, bastante tradicionais na literatura mais geral de tecnologia da informação, surgem como métodos também frequentemente utilizados na comparação com

técnicas genéricas de PUL ou como parte integrante do método em duas fases de PUL. Conforme a **Figura 3.8**, a técnica de Máquina de Vetores de Suporte (*Support Vector Machine*) surge como técnica muito usual em artigos que possuem como foco a aplicação de técnicas de PUL. Outros algoritmos de classificação também aparecem com boa frequência nas publicações selecionadas: *Naive Bayes*, *K-Nearest Neighbors*, Regressão Logística, *Expectation-Maximization*, *Long Short Term Memory*, *Random Forest*. Como método mais recente, pode-se citar o NPULUD, técnica de classificação para Aprendizado Positivo e Não-rotulado baseado em Vizinhança usando Árvores de Decisão (*Neighborhood-Based Positive Unlabeled Learning Using Decision Tree*) (Ghasemkhani et al., 2024).

Redução de dimensionalidade

A análise de algoritmos que utilizam alguma técnica de PUL mostram a diversidade de abordagens e aplicações, sendo, entretanto, muito particular a utilização de cada adaptação para a finalidade da pesquisa. A aplicação de técnicas de **redução de dimensionalidade** ou de **fatorização de matrizes** também são aplicadas visando melhorar a aplicação de alguma técnica de PUL.

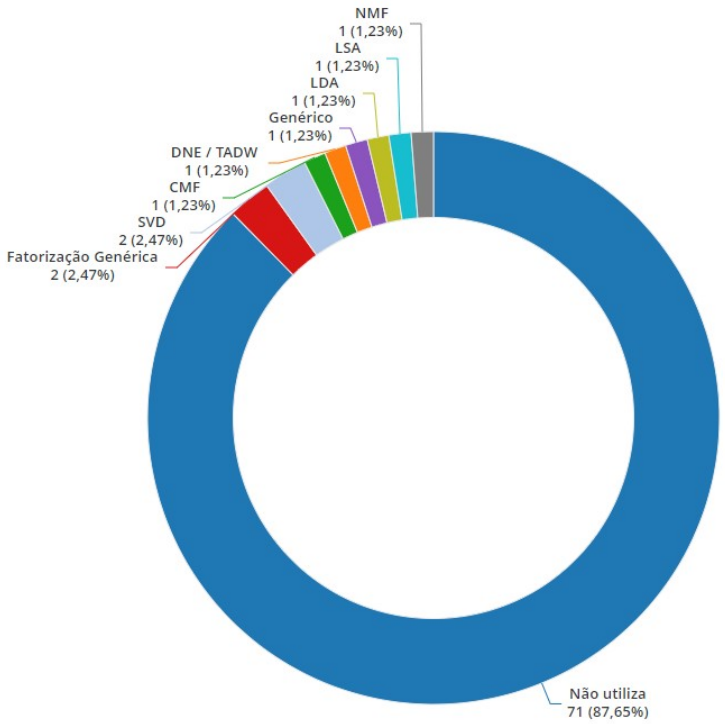


Figura 3.9: Artigos selecionados que aplicam técnica de redução de dimensionalidade

Através da Figura 3.9, verifica-se que a maioria dos artigos selecionados, ou seja, artigos focados em técnicas de PUL para classificação de texto, não aplica técnica de redução de

dimensionalidade, seja para experimentos, seja para desenvolvimento da metodologia do algoritmo.

Os artigos Jiang et al. (2018) e Tanielian and Vasile (2019) aplicam técnicas genéricas de fatorização de matrizes, com objetivo de minimizar o erro na construção dos vetores de palavras incorporadas e para treinar o modelo a partir dos exemplos positivos, respectivamente. Em Jungmaier et al. (2020), a matrix construída de co-ocorrências das palavras é muito grande e esparsa. Para obter uma representação vetorial de textos densa e com poucas dimensões foi aplicada a técnica de redução de dimensionalidade SVD, (*decomposição em valores singulares*), que consiste na fatoração de uma matriz em três matrizes, sendo duas unitárias e outra diagonal contendo os valores da matriz original.

Liu et al. (2019) aplica LDA, *Latent Dirichlet Allocation* para realizar a identificação de tópicos nos textos, e assim relacionar diferentes documentos a partir de tópicos. Essa aplicação pode ser importante para a representação vetorial dos documentos. Zhang et al. (2021) utiliza a técnica TADW para representação de texto em nós a partir da fatorização indutiva de matrizes. Banerjee et al. (2018), em sua pesquisa, aplica o LSA, *Latent Semantic Analysis*, técnica de processamento de linguagem natural que extrai o relacionamento entre documentos e palavras de textos. No LSA, como a matriz gerada é esparsa, é realizada uma aproximação de baixo nível para tornar a matriz densa e mais facilmente computável.

Outra técnica aplicada é o CMF, *Fatorização Coletiva de Matriz*, que realizar a fatorização da matriz de interação e das matrizes complementares (Gharahighehi et al., 2022). Em Kaur et al. (2021), o NMF foi utilizado para extrair os tópicos mais frequentes dos documentos e para auxiliar na classificação das classes.

Nota-se que uma pequena porção das publicações selecionadas abordou a utilização dos algoritmos de redução de dimensionalidade ou de fatorização de matrizes. No presente estudo será realizada a análise de uma versão adaptada do NMF para fins de verificação de desempenho de classificação de dados.

Deep NMF

Quando se trata da utilização do NMF a partir de uma visão de aprendizado profundo, não houve identificação de artigos dentro da string de busca definida, concluindo-se assim que não há estudos ou pesquisas que apliquem o *Deep NMF* ou alguma adaptação do NMF para aprendizado profundo (*deep learning*) a problemas de PUL.

Entretanto, alguns artigos relacionam a aplicação do *Deep NMF* a problemas de classificação de texto ou para modelagem de forma genérica do NMF em redes neurais. Flenner and Hunter (2017) introduz o assunto ao indicar que redes neurais profundas precisam de uma grande quantidade de dados observados e características (*features*) com representa-

ções formatadas. Desses obstáculos, pode-se utilizar o NMF, que produz bons resultados ao entender os dados e entregar uma modelagem de tópicos estruturada. Assim, o arcabouço (*framework*) proposto pelos autores mostra que é possível combinar a interpretabilidade de métodos de representação de tópicos, como o NMF, com a acurácia de redes neurais profundas.

Outras abordagens indicam que redes neurais não-supervisionadas podem ser usadas para aprender a partir de documentos e construir uma boa representação deles. Wang and Zhang (2023) indica que o modelo proposto de *Deep NMF*, no seu artigo, de forma não-supervisionada, pode melhorar a eficácia de descoberta de tópicos, além de reduzir a complexidade computacional, um dos obstáculos de métodos de modelagem de tópicos. A ideia que diferencia a abordagem de *Deep NMF* da abordagem de NMF é a de realizar a fatorização das matrizes em múltiplos fatores não lineares.

Nasser et al. (2021) levanta a grande evolução da aplicação de modelos de redução de dimensionalidade, como o NMF, para o domínio biológico e de saúde. Uma das questões é a de que algoritmos como o NMF geralmente possuem uma predisposição para alcançarem um solução que é ótima localmente. Nesse quesito, a aplicação do NMF a partir de várias camadas, como é feito em técnicas de redes neurais, pode resolver esse problema e levar a uma solução que seja ótima de maneira global ou que consiga encontrar soluções mais próximas da melhor solução possível.

Entretanto, muitos estudos levantam os obstáculos em se utilizar o *Deep NMF* para processos de classificação, clusterização, como *overfitting*, complexidade computacional, perda de informação nas camadas, entre outros. Uma das propostas de estudo é de se utilizar regularização através de *embedding*, como abordado em Moayed H. (2024). Nesse estudo, dados de entrada são integrados com versões ruidosas dos dados em canais, para posterior extração das features e agregação em uma única *feature* para performar a tarefa de classificação.

Mahmoodi et al. (2024) também aborda a dificuldade do *overfitting* em modelos de redes neurais, e realiza um estudo em que usa NMF para reconstruir redes esparsas e *Deep NMF* para reformatar a estrutura hierárquica de um grafo.

Em W. Chen (2022), diversos tipos de algoritmos de *Deep NMF* foram categorizados e descritos, mostrando a evolução do estudo dessa área do conhecimento específica dentro de algoritmos de redução de dimensionalidade e de redes neurais. Para fins do estudo atual, a categoria de *Basic Deep NMF* serão mais adequadas para adaptar o método para problemas de PUL, objetivo final dessa pesquisa.

3.3.5 Outras análises

Nesta seção, a análise será realizada sobre as métricas de avaliação mais frequentemente adotadas, além dos resultados obtidos para a categorização de artigos, definido na **Seção 3.2.6**.

Métricas de avaliação

No que tange à avaliação dos resultados, temos a utilização de diversas **métricas de avaliação** para fins de mensuração do desempenho, acurácia e eficiência dos algoritmos desenvolvidos.

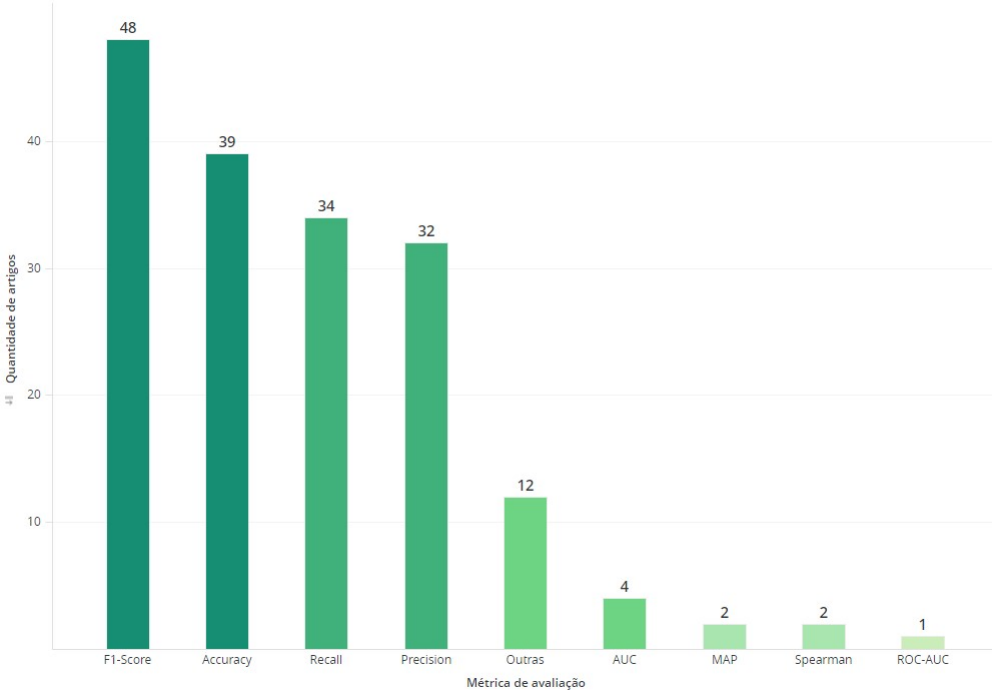


Figura 3.10: Métricas de avaliação aplicadas nos artigos selecionados

Como explicitado na Figura 3.10, as métricas mais utilizadas na base de artigos selecionados são: **Precisão**, métrica que indica quantas predições positivas corretas do total de predições positivas; ***Recall***, medida que indica quantas predições positivas corretas do total de casos positivos na base de dados; **Acurácia**, métrica que indica o número de predições corretas dentro do total de predições; **F1-Score**, que combina as métricas de *Recall* e *Precisão* através de uma média harmônica, pois essa dará maior peso, caso alguma das métricas possua valor muito baixo (Goutte and Gaussier, 2005).

São métricas bastante usuais em avaliação de algoritmos de classificação, por isso serão objeto de aplicação nessa pesquisa.

Categorização dos artigos

Conforme foi definido na Seção 3.2.6, os artigos selecionados foram categorizados de acordo com a metodologia utilizada de aplicação de PUL. Seguem abaixo as categorias.

1. Artigos que focam em *one-class classifiers*;
2. Artigos que aplicam métodos de redução de dimensionalidade;
3. Artigos que usam o método PUL *Two-step techniques*;
4. Artigos que usam o método PUL *Biased Learning*;
5. Artigos que usam outros métodos PUL como *Generative Adversarial Network* - GANs, Incorporação da Classe principal.

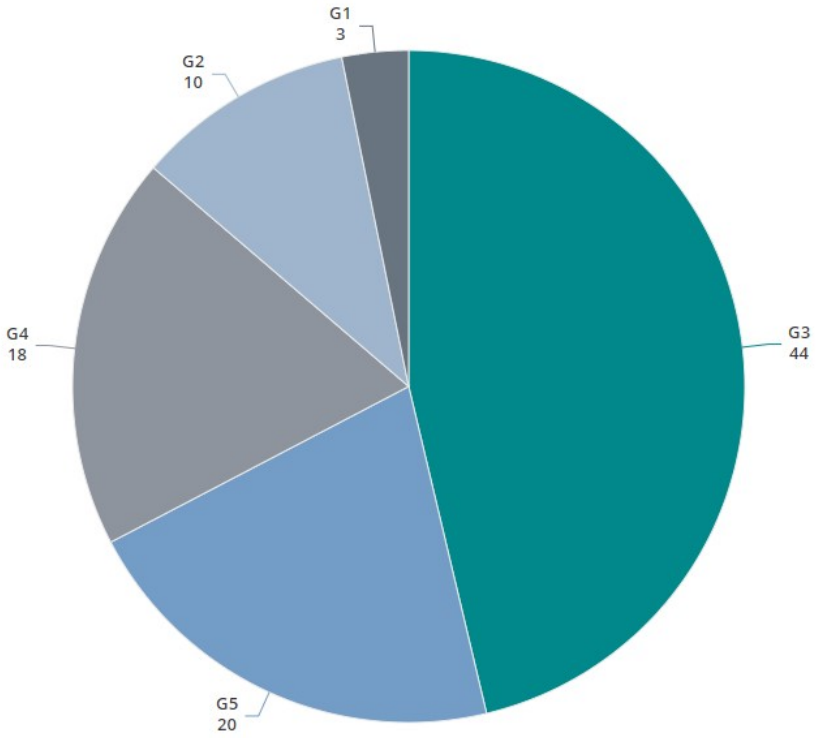


Figura 3.11: Quantidade de artigos em cada categoria de método PUL

Conforme a Figura 3.11, a categoria 3, onde a técnica de duas fases é aplicada foi a mais comum dentro da base de artigos selecionados. Nota-se que apenas 12,34% dos artigos (10 artigos) aplicaram algum método de redução de dimensionalidade, o que demonstra pouca exploração desse tópico em conjunto com a abordagem de *Positive Unlabeled Learning*, conforme também foi visto na seção que tratou de técnicas de redução de dimensionalidade.

3.4 Algoritmos de PUL na literatura

Como analisado nas seções anteriores desse capítulo, especialmente o que foi destacado na 3.3.5, a técnica de dois passos é a mais frequentemente aplicada, sendo alvo principal do nosso estudo. Algumas abordagens que usam a técnica de dois passos, são Unbiased Positive-Unlabeled (UPU) (Yang et al., 2020), Spy Expectation-Maximization (Spy-EM) (He et al., 2020), Non-negative Positive-Unlabeled (nnPU) (Ji et al., 2023), Positive and Unlabeled Learning by Label Propagation (PU-LP) (Jaemin et al., 2022; Ma and Zhang, 2017), Rocchio Support Vector Machine(RC-SVM) (Li and Liu, 2003), Label Propagation for Positive and Unlabeled Learning (LP-PUL) (Carnevali et al., 2021).

No algoritmo ***Unbiased Positive-Unlabeled (UPU)***, a proposição feita é de que os dados não-rotulados sejam considerados como uma combinação com pesos dos dados positivos e negativos (Kyrio et al., 2017). Em Yang et al. (2020), o algoritmo *uPU* utiliza uma relação entre os dados positivos e os não-rotulados para realizar uma combinação de pesos dos dados. Algumas abordagens também transformam o problema em um problema de otimização convexa ao aplicar funções de perda convexas aos dados positivos e não-rotulados (Du Plessis et al., 2015).

Já o método ***Spy Expectation-Maximization (Spy-EM)*** consiste das etapas de esperança, onde as amostras são rotuladas, e de maximização, onde todo o conjunto de dados é usado para criar parâmetros para o modelo (He et al., 2020). Esse método busca encontrar os dados pertencentes à classe negativa que são mais confiáveis.

O algoritmo ***Non-negative Positive-Unlabeled (nnPU)*** realiza uma correção nas estimativas de risco para os dados negativos para garantir que o risco estimado empiricamente para as amostras negativas seja sempre positivo (Ji et al., 2023). Esse tipo de método utiliza o conceito de estimador de risco não-negativo.

O **RC-SVM** é baseado no SVM e no Rocchio (Li and Liu, 2003), utilizando este último para produzir protótipos tanto para os dados positivos quanto para os dados não rotulados. Documentos são selecionados para formarem o conjunto de treinamento, que é composto por documentos rotulados da classe positiva (classe de interesse) e por documentos não-rotulados que podem pertencer à classe positiva ou não, ou seja, o conjunto de treinamento T é formado por $T = \{D_1, D_2, ..., D_l, D_{l+1}, ...D_{l+u}\}$, onde D_l documentos são rotulados, e os D_u documentos são não-rotulados, e que para problemas do tipo PUL, $\left\|D_u\right\| \gg \left\|D_l\right\|$. Os documentos que são mais similares aos dados não rotulados do que aos dados positivos são definidos como documentos negativos confiáveis, ou *Reliable Negatives - RN*, a partir de um classificador *Rocchio*. Então, em uma segunda etapa, a Máquina de Suporte de Vetores (*Support Vector Machine*) é aplicado de forma iterativa para construir e estruturar o classificador (Caravanti de Souza et al., 2021), a partir dos dados rotulados positivos e dos documentos negativos classificados anteriormente (RN).

O **PU-LP** (Jaemin et al., 2022) é um algoritmo baseado em grafo que localiza os nós, a partir do conjunto de nós não rotulados, que possuem uma pontuação de similaridade menor, com base na matriz de similaridade criada através do *k-Nearest Neighbors*, e os define como negativos confiáveis (RN). Em seguida, os nós com uma pontuação de similaridade mais alta são atribuídos ao conjunto de nós positivos, utilizando isso para rotular o restante dos documentos, transformando-o em um problema positivo-negativo. Como na denominação do algoritmo, rótulos podem ser propagados por caminhos no grafo tal que a presença de vários nós vizinhos em comum, ambos os nós pertencerão, com alta probabilidade, à mesma classe (Caravanti de Souza et al., 2021).

Por fim, o **LP-PUL**, também considerando uma abordagem baseada em grafo, consiste em três etapas: construir um grafo de documentos baseado em similaridade; em seguida, usar o grafo construído para inferir documentos negativos confiáveis com base na vizinhança dos nós; e, por fim, usar documentos positivos e negativos para aplicar um mecanismo de propagação de rótulos aos documentos não-rotulados (Carnevali et al., 2021). Deve-se considerar que, na construção do grafo, os relacionamentos entre nós devem acontecer, prioritariamente, entre aqueles objetos mais similares, usando, por esse motivo, a similaridade por cosseno, como medida padrão, para medição da similaridade dos objetos.

Quando tratamos da aplicação do algoritmo **NMF** em problemas de classificação de texto em situações onde temos apenas dados PU, a literatura não fornece muitos trabalhos, constando apenas alguns trabalhos que usam o NMF como técnica de redução de dimensionalidade ou em modelagem de tópicos, o que se mostra muito mais comum em pesquisas recentes, como é visto em Kaur et al. (2021), onde NMF é usado em um contexto de extração de características em modelagem de tópicos como uma métrica de similaridade das características (*features*). Já em Li et al. (2016), o NMF é utilizado para explorar os tópicos/clusters das características dos textos. Por se tratar de uma técnica não-supervisionada, o trabalho incorporou dados supervisionados nas funções de divergência aplicadas no NMF. Então, quando se trata de adotar o NMF como técnica de classificação de dados, não foram encontrados artigos que tratassem dessa abordagem, principalmente quando se trata de problemas de PUL em dados do tipo texto.

3.5 Considerações finais

A partir de toda a análise descrita anteriormente neste capítulo, pode-se destacar os seguintes tópicos:

- A abordagem mais frequente no que tange à aplicação de PUL se traduz na utilização de técnicas em duas fases, onde na primeira, usualmente, se identifica aqueles exem-

plos negativos mais confiáveis a partir de alguns exemplos positivos e de exemplos não-rotulados e, posteriormente, realiza-se a classificação do restante dos exemplos.

- Não houve uma exploração, em termos de pesquisa, extensa para utilização de técnicas de redução de dimensionalidade em problemas de PUL. Mas, ainda assim, podemos destacar a aplicação de alguns métodos, como o NMF, o LSA e o LDA.
- O método de classificação de dados textuais empregado em boa parte dos artigos selecionados foi o SVM, sendo seguido por *Naive Bayes* e KNN, mostrando que técnicas tradicionais de aprendizado supervisionado são usualmente aplicados em avaliação de modelos de PUL.
- A predominância do uso de bases de dados no idioma inglês ainda é grande frente aos outros idiomas. E a utilização de bases de dados voltados para classificação de notícias falsas, identificação de avaliações reais de produtos e serviços e análise de publicações científicas são as mais comuns. Também destaca-se o estudo de novas metodologias de aplicação genérica para PUL.
- Muitos dos artigos que abordam a vertente em redes neurais do NMF, o *Deep* NMF, foram publicados nos últimos três anos, o que demonstra que essa abordagem tem sido estudada atualmente nas mais diversas possibilidades de aplicação e com distintos processos de formulação.

Capítulo 4

Desenvolvimento do Trabalho

Com o objetivo de responder às questões levantadas na Seção 1.2, a partir da hipótese levantada na Seção 1.3, o presente capítulo apresenta a proposta de um algoritmo adaptando o NMF para classificação de dados do tipo texto em problema de *Positive Unlabeled Learning* e também apresenta a proposta de um algoritmo utilizando aprendizado profundo, o *Deep NMF*, para melhorar o processo de classificação do primeiro algoritmo proposto, através da avaliação do desempenho de classificação e de eficiência no uso de recursos computacionais.

4.1 Datasets

Os experimentos foram aplicados em 17 conjuntos de dados: coleções de textos compostas por uma coleção de termos e uma classe para cada documento (Rossi et al., 2013). Essas coleções estão relacionadas às áreas: documentos médicos, documentos científicos, artigos de notícias, documentos TREC (imobiliários) e páginas web. Pode ser observada uma variabilidade na quantidade de documentos, quantidade de palavras e quantidade de classes dentro do conjunto de *datasets*, permitindo obter-se uma avaliação menos enviesada, sendo possível aplicar em outras coleções de textos com maior possibilidade de manter o nível de desempenho. Algumas informações sobre os conjuntos de dados estão resumidas na **Tabela 4.1**.

4.2 Algoritmos Baselines

Os experimentos foram realizados considerando diferentes parâmetros e algoritmos do estado da arte em técnicas de PUL e *One-Class Learning*, como reportado em Carnevali et al. (2021). Além dos algoritmos propostos, NMFPUL e *Deep NMF*, outros algoritmos do estado da arte foram selecionados para fins de comparação dos resultados.

Tabela 4.1: Características das coleções de documentos

Dataset	Domínio	# Documentos	# Palavras	Média de palavras	# Classes
CSTR	Relatórios Científicos	299	1726	54.27	4
Fbis	Artigos de notícias	2463	2001	159.24	17
Oh0	Documentos médicos	1003	3183	52.5	10
Oh5	Documentos médicos	918	3013	54.43	10
Oh10	Documentos médicos	1050	3239	55.64	10
Oh15	Documentos médicos	3101	54142	17.6	10
Re0	Artigos de notícias	1504	2887	51.73	13
Re1	Artigos de notícias	1657	3759	52.70	25
SyskillWebert	Páginas Web	334	4340	93.16	4
Tr11	Documentos TREC	414	6430	281.66	9
Tr12	Documentos TREC	313	5805	273.60	8
Tr21	Documentos TREC	336	7903	469.86	6
Tr23	Documentos TREC	204	5833	385.29	6
Tr31	Documentos TREC	927	10129	268.50	7
Tr41	Documentos TREC	8778	7455	19.54	10
Tr45	Documentos TREC	690	8262	280.58	10
WAP	Páginas Web	1560	8461	141.33	20

Inicialmente, foram selecionados, a partir das técnicas usadas em Carnevali et al. (2021), os três algoritmos que utilizam a técnica de dois passos para problemas de *PUL* com os melhores resultados nesse trabalho citado, que são: **RC-SVM**, **PU-LP** e **LP-PUL**. Os experimentos utilizados nesse trabalho para os algoritmos selecionados, detalhadamente descritos na **Seção 3.4**, foram aproveitados de Carnevali et al. (2021).

Para fins de comparação de desempenho do algoritmo, também foi abordada a utilização de uma técnica de *One-Class Learning*, que utiliza todos os documentos positivos para construir um classificador, enquanto todos os outros documentos pertencem à classe negativa (Ta et al., 2019). Neste trabalho, foi utilizado um algoritmo baseado em **K-Means**, no qual os documentos positivos são divididos em grupos e cada grupo possui um centróide, que é calculado com base na média dos vetores de documentos do grupo. A similaridade de cada novo documento é comparada aos centróides para definir a classe à qual o documento pertencerá. Também form aproveitados os resultados obtidos nos mesmos conjuntos de dados do tipo texto em Carnevali et al. (2021).

4.3 NMFPUL - Non-negative Matrix Factorization para Positive Unlabeled Learning

O método proposto, *Non-negative Matrix Factorization for Positive Unlabeled Learning* (NMFPUL), aplica uma representação numérica vetorial aos dados de texto, da mesma forma que o modelo de saco de palavras e tf-idf, e adapta o conceito de NMF para classificar dados não rotulados a partir de dados PU. Seja V a matriz documento-termo,

nl o número de documentos rotulados, m o número máximo de iterações de Atualização Multiplicativa, t a tolerância para o erro e k o número de tópicos, o algoritmo é construído através de quatro etapas principais:

1. Construção da **matriz documento-termo** aplicando o modelo de *bag of words* e **TF-IDF**.

2. Formação das matrizes documento-tópico e tópico-termo na primeira iteração. Os documentos rotulados são selecionados utilizando o mecanismo de **Selected Completely at Random (SCAR)**, escolhendo aleatoriamente exemplos positivos para rotulagem.

3. Para garantir que documentos **positivos** rotulados possuam o maior valor na primeira posição (primeiro tópico), na matriz W, o primeiro valor, ou seja na primeira posição da linha $d(i)$, seja atribuído o maior valor da matriz W inteira. Um valor positivo, diferente de zero, ϵ é atribuído para todas as dimensões restantes, sendo representado por um valor muito baixo. Utilizamos o valor 0.001 no nosso experimento para garantir uma magnitude baixa. Essa operaçào foi denominada de *suppress*. Esse passo melhora a separação entre documentos positivos e documentos não rotulados.

4. Classificação dos documentos não rotulados utilizando as matrizes documento-tópico e tópico-palavras atualizadas, a partir de um processo iterativo tendo como função objetivo a divergência de *Kullback-Leibler*.

Ao combinar os princípios do NMF com os requisitos do aprendizado através de dados PU, o método aqui proposto, NMFPUL, busca classificar dados não rotulados no framework PU de forma efetiva. Veja a descrição no Algoritmo 1.

Inicialmente, é realizada uma etapa de pré-processamento seguindo a seguinte sequência: primeiro, as classes e o próprio texto são incorporados a variáveis para que o processo de vetorização através do TF-IDF seja executado; em seguida, uma classe é escolhida aleatoriamente para ser a classe positiva, enquanto as demais são definidas como negativas; a partir da quantidade de documentos rotulados definidos inicialmente, os índices da classe positiva são embaralhados e os primeiros nl documentos são definidos como os documentos rotulados positivos, e os documentos positivos restantes são definidos como não rotulados.

O algoritmo NMFPUL recebe como entrada a matriz documento-termo V e os índices dos documentos rotulados. Alguns valores são importantes de serem inicializados antes de instanciar o algoritmo: o número de tópicos que a parte do algoritmo de NMF deve usar para construir as matrizes W e H, o número de documentos rotulados da classe positiva,

Algoritmo 1 Non-negative Matrix Factorization para Positive Unlabeled Learning

Input: matrix representation V, number of labeled documents nl, maximum interations m, tolerance t, number of topics k

Output: Matrices W, H

Function `NMFPUL`(V, nl, k, m, t, $\epsilon = 0.001$)**:**

```
V ⟵ Document-word matrix created by TfidfVectorizer ;
Initialize W and H with random values;
positive_class_index ⟵ 0;
for n in range(1, m + 1) do
    Update W and H according to Multiplicative Update;
    W[labeled, 0] ← max(W);
    W[labeled, 1 :] ← ϵ;
    Calculate Kullback-Leibler divergence between V and W × H as error;
    if error < t then
        break
    else
        continue to next iteration
    end
end
return W, H
```

End Function

o número máximo de iterações para a convergência, a tolerância que pode interromper o processo iterativo se a divergência de *Kullback-Leibler* alcançá-la antes do número máximo de iterações.

4.4 Deep NMF

O método de Deep NMF proposto utiliza o método de desdobramento para o NMF, utilizando conceitos abordados em Nasser et al. (2021), para aplicação em *PUL*. O método proposto utiliza as matrizes de saída W e H, conforme abordado na Seção 2.2.4, resultantes do Deep NMF para realizar iterações no método adaptado do NMF, o NMFPUL. A descrição do método pode ser conferida no Algoritmo 2.

De forma mais detalhada, cada camada da rede neural, treinada por iterações, tem como objetivo formar o algoritmo *Deep* NMF para reduzir o erro de reconstrução da matriz V, formar as matrizes finais W e H e, posteriormente, utilizar o algoritmo NMFPUL para realizar a Atualização Multiplicativa das Matrizes e classificar os documentos do *dataset* através da comparação das classes originais de cada documento com as classes preditas através do algoritmo.

No NMFPUL, foi detalhado que o algoritmo necessita iniciar com valores randômicos para as matrizes W e H, o que pode gerar uma dificuldade na convergência do algoritmo, além da possibilidade de consumo de mais tempo e recursos computacionais. Dessa forma,

Algoritmo 2 Procedimento de Treinamento para uma rede não-supervisionada de um modelo Deep NMF

Input: Input tensors V_{tns} and H_{tns}, initial weights W_{init_tns}, num_layers, $network\_train\_iterations$, $n\_components$, $features$, learning rate lr, regularization parameters l_1, l_2, $positive\_class\_indices$

Output: Trained model $deep\_nmf$, training costs $dnmf\_cost$, trained weights $dnmf\_w$, output matrix H_out

Function `TrainDeepNMF(`V_{tns}, H_{tns}, W_{init_tns}, num_layers, $network\_train\_iterations$, $n\_components$, $features$, lr, l_1, l_2, $positive\_class\_indices$`):`

- Build the architecture $deep\_nmf$ with or without regularization using $UnsuperNet(num\_layers, n\_components, features, l_1, l_2)$;
- Initialize model $deep\_nmf$ and parameters $dnmf\_w$;
- Initialize $optimizerADAM$;
- Prepare input tensors (H_{tns}, V_{tns});
- **for** i *in* $range(0, network\_train\_iterations-1)$ **do**
  - Compute output H_out from $deep\_nmf$
  - Calculate loss using $cost\_tns(V_{tns}, dnmf\_w, out, l_1, l_2)$
  - Perform backpropagation
  - Update model parameters using $optimizerADAM$
  - Modify output matrix H based on $positive\_class\_indices$
  - Update $dnmf\_w$ using non-negative least squares (NNLS)
  - Append current loss to $dnmf\_cost$
  - Append test performance to $dnmf\_cost$
- **end**
- **return** $deep\_nmf$, $dnmf\_cost$, $dnmf\_w$, H_out

End Function

a geração de matrizes mais adequadas para iniciar o processo de Atualização Multiplicativa pode gerar melhores resultados no que tange ao desempenho de classificação e também no quesito de economia de recursos temporais e computacionais.

Assim como no NMFPUL, no método proposto de *Deep NMF* foi feita a incorporação de um mecanismo para considerar explicitamente os documentos da classe positiva durante o treinamento. Isso é alcançado modificando a matriz de características decomposta para enfatizar as características associadas à classe positiva, orientando assim o modelo a aprender representações que são discriminativas das classes de interesse. Durante o processo iterativo da rede, a partir dos índices das classes positivas definidos como entrada do método, é realizada a supressão dos elementos correspondentes a esses índices para possuírem os maiores valores da matriz de saída.

Assim, pode-se sumarizar o método nas etapas:

1. Formulação das **camadas da rede neural**, através das matrizes V e H
2. Formulação do **algoritmo de treinamento** (*Deep NMF*) e execução.

3. Utilização das **matrizes W e H** resultantes do treinamento da rede como entrada para o método NMFPUL.

4. Execução do **processo iterativo do NMF adaptado (NMFPUL)** para classificação dos documentos nas classes.

4.5 Avaliação

4.5.1 Estrutura dos Experimentos

Um cenário simulado de aprendizado positivo-negativo (PUL), aplicando o processo a coleções de texto multiclasse, usando os datasets descritos na **Secão 4.1**, foi considerado, pois foi adotada uma abordagem iterativa, através do processo de Atualização Multiplicativa, conforme resumido na Figura 4.1. Durante cada iteração, uma única classe da coleção de textos foi designada como classe positiva, enquanto as demais foram designadas como classe negativa. Após isso, foi definida uma variável para selecionar a quantidade de documentos rotulados na classe positiva selecionada. Os valores para esse parâmetro foram $D_+ = 1, 5, 10, 20, 30$ e os documentos rotulados foram selecionados aleatoriamente, conforme definido na Seção 2.1.2, por meio do mecanismo de rotulagem SCAR. Os documentos restantes, ou seja, aqueles pertencentes à classe negativa e os positivos restantes, foram deixados sem rótulo.

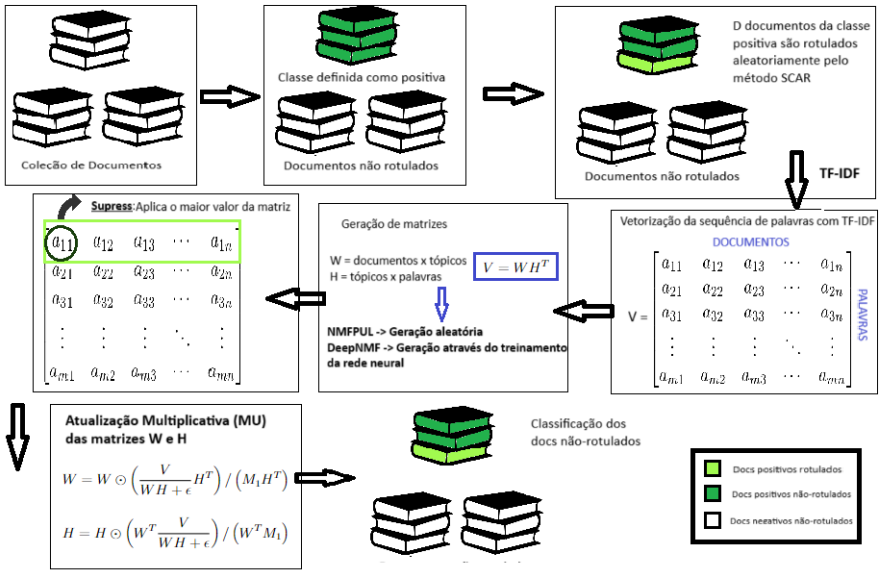


Figura 4.1: Esquema com a estrutura do experimento realizado no NMFPUL e Deep NMF até a avaliação

Para os algoritmos propostos NMFPUL e *Deep NMF*, algumas especificidades foram consideradas (Carnevali et al., 2021). Inicialmente, a matriz documento-palavra V foi

construída através da vetorização, aplicando a transformação TF-IDF, da sequência de palavras de cada documento da coleção de documentos.

Para o NMFPUL, após definir a classe positiva e os documentos positivos rotulados aleatoriamente, as matrizes W e H são inicializadas randomicamente, tendo o parâmetro do número de dimensões latentes k definindo as dimensões dessas matrizes. Já para o Deep NMF, o processo iterativo inicializa com a matrizes W e H obtidas do processo de treinamento da rede neural.

A função *suppress* é aplicada para garantir que os documentos positivos rotulados na matriz W no primeiro valor da matriz, ou seja, na primeira posição da linha $d(i)$, possuam o maior valor. Essa função atribui o maior valor da matriz W inteira para esses documentos positivos rotulados, ou seja, ela força o valor mais alto na primeira posição da linha correspondente para os documentos rotulados.

No NMFPUL, como função objetivo usada para calcular a aproximação das matrizes, foi aplicada a divergência Kullback-Leibler (KL) (Hien and Gillis, 2020). Essa métrica divergente quantifica a dissimilaridade entre duas distribuições de probabilidade e é adequada para ser adotada na execuçao do NMFPUL e *Deep NMF*.

Como os algoritmos **NMFPUL** e ***Deep NMF*** são construídos usando os parâmetros do NMF, é necessário definir o número máximo de iterações para que os algoritmos convirjam. Assim, foi definido um número máximo de 100 iterações ($m = 100$) para esse experimento, para verificar a convergência do algoritmo, visando analisar o erro definido e em que momento seria apropriado parar as execuções para evitar iterações desnecessárias.

Dado que a seleção aleatória de documentos para o conjunto positivo pode influenciar o resultado da classificação, foram realizados 10 testes, selecionando documentos rotulados diferentes a cada vez. Uma média simples dos resultados desses testes foi calculada para mitigar o efeito da aleatoriedade. Além disso, para a consistência dos experimentos, toda essa configuração foi executada três vezes para cada conjunto de dados, alterando o parâmetro ($[micro, macro, weighted]$) do F1 score, sendo melhor detalhado no tópico seguinte. Em posse do resultado de *F1 Score* para as três instâncias, é calculado o valor médio do *F1 Score*, sendo esse o resultado final de desempenho e classificação do algoritmo.

4.5.2 Critérios de Avaliação

A partir da configuração definida em 4.5.1, a avaliação de desempenho de classificação dos algoritmos selecionados e propostos foi avaliada através da métrica *F1 Score*. As instâncias $[micro, macro, weighted]$ foram usadas para calcular o F1 score, extraindo o resultado dos algoritmos **NMFPUL** e ***Deep NMF*** trêz vezes, onde:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{4.1}$$

$$Precision = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{4.2}$$

$$Recall = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{4.3}$$

onde TP (Verdadeiros Positivos) é o número de documentos positivos corretamente classificados como positivos, FP (Falsos Positivos) é o número de documentos negativos classificados como positivos e FN (Falsos Negativos) é o número de documentos positivos classificados como negativos.

Os mesmos critérios são utilizados para o algoritmo Deep NMF, pois na formulação do presente trabalho, a rede neural fornece como saída matrizes W e H que serão usadas como entrada no processo de Atualização Multiplicativa do NMFPUL, sendo parte da execução do algoritmo Deep NMF, como detalhado na Seção 4.4. Dessa forma, também tem-se os resultados do desempenho de classificação do Deep NMF usando a mesma formulação detalhada anteriormente.

Além dos critérios do desempenho de classificação dos algoritmos, será considerada a avaliação da convergência do algoritmo, visando verificar se houve algum ganho no que tange à velocidade de execução do algoritmo, sem que haja perda de desempenho, e também no menor uso de recursos computacionais para o processo de classificação dos documentos. Nesse caso, é realizado o cálculo do erro e da diferença entre erros consecutivos de iterações para verificar se esse valor atinge um patamar menor que um valor de tolerância definido, no caso definido em $[tol = 1e - 3]$. Caso esse valor atinja um valor menor que a tolerância, o algoritmo pode cessar as iterações para aquela execução.

Para os algoritmos **NMFPUL** e ***Deep NMF***, cada execução contém um processo de cem (100) iterações, no processo de Atualização Multiplicativa, para verificar a convergência do algoritmo, através da análise do erro e em que momento seria apropriado parar as execuções para evitar iterações desnecessárias.

Capítulo 5

Resultados

Foi realizada uma comparação entre o *F1-Score* do *NMFPUL*, do *Deep NMF* e dos outros algoritmos (*baselines*) nos mesmos conjuntos de dados. Foram utilizados os resultados produzidos em Carnevali et al. (2021) nas coleções de documentos selecionadas, descritas na Seção 4.1 para realizar a análise do desempenho da classificação. Além dos resultados dos algoritmos *baselines*, os resultados para o *NMFPUL* e o *Deep NMF* foram extraídos seguindo o processo detalhado na Seção 4.5. Nas tabelas 5.1 e 5.2, alguns resultados podem ser vistos, enquanto a totalidade dos resultados podem ser visualizados na Figura 5.1. As tabelas completas com todas as bases de dados e resultados estão no **Apêndice I**.

Tabela 5.1: Valores de F1 Score para diferentes algoritmos na Coleção de Documentos CSTR

Algoritmo	***# Documentos Rotulados - CSTR***				
	1	5	10	20	30
K-Means	0.49	0.61	0.68	0.65	0.51
PU-LP	0.55	**0.74**	**0.78**	0.7	0.69
RCSVM	0.02	0.12	0.29	0.52	0.39
LP-PUL	0.61	0.69	0.77	**0.79**	**0.8**
NMFPUL	**0.632**	0.679	0.715	0.735	0.741
Deep NMF	0.457	0.577	0.710	0.725	0.762

Tabela 5.2: Valores de F1 Score para diferentes algoritmos na Coleção de Documentos Oh10

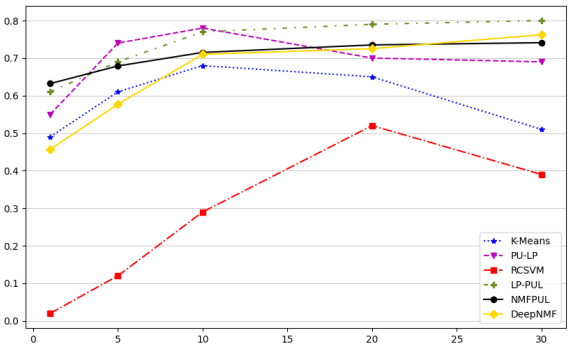
Algoritmo	***# Documentos Rotulados - Oh10***				
	1	5	10	20	30
K-Means	0.37	0.52	0.60	0.61	0.61
PU-LP	0.22	0.40	0.47	0.54	0.50
RCSVM	0.01	0.11	0.28	0.45	0.53
LP-PUL	0.44	0.56	0.60	0.62	0.62
NMFPUL	0.713	0.714	0.716	0.725	0.727
Deep NMF	**0.751**	**0.758**	**0.783**	**0.812**	**0.817**

É satisfatório afirmar que, a partir dos dados das métricas de desempenho dos diversos algoritmos analisados, que as abordagens do NMFPUL e do Deep NMF superam a maioria dos *baselines* na maioria das coleções de documentos e, em algumas delas, obteve um resultado satisfatório e próximo ao desempenho das outras técnicas apresentadas. Por exemplo, nos conjuntos de dados com menor quantidade de documentos e classes, os algoritmos *NMFPUL* e *Deep NMF* apresentam desempenho similar às outras técnicas comparadas.

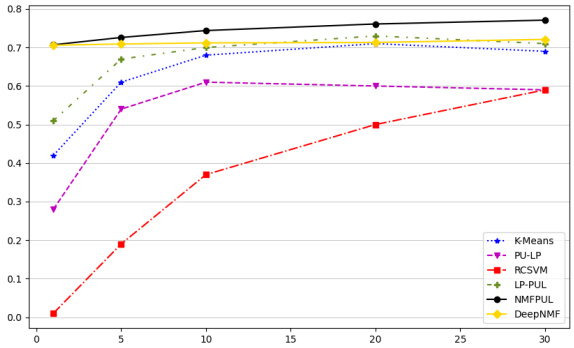
No entanto, quando são analisados conjuntos de dados maiores, o *NMFPUL* e o *Deep NMF* superam todos os algoritmos, especialmente quando há um número pequeno de documentos positivos rotulados. E isso é compreensível, uma vez que o NMF tradicionalmente funciona melhor quando a quantidade de dados usada não é pequena (Lin and Boutros, 2020). Portanto, embora o *NMF* reduza a esparsidade e os dados ruidosos quando aplicado em sua forma original, ele precisa de um volume mínimo de dados para produzir bons resultados.

Além disso, pode ser apontado, a partir do conceito apresentado para o *Deep NMF* e da metodologia utilizada para implementá-lo, que a utilização de um método de aprendizado profundo, através de redes neurais, usando uma adaptação do método *Deep NMF* para aplicação em dados do tipo PU, trouxe ganho de desempenho na classificação de documentos, obtendo resultados próximos ou melhores daqueles que o NMFPUL demonstrou, na maioria dos conjuntos de dados analisados.

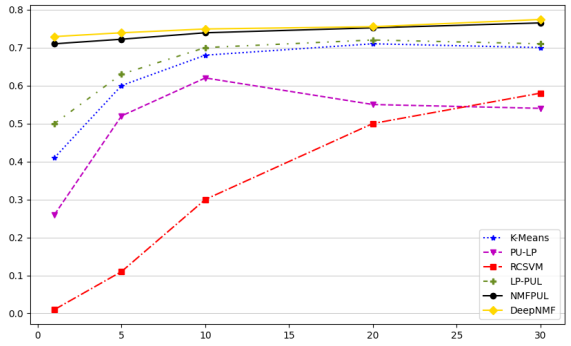
Dessa forma, é possível afirmar que a abordagem utilizada nesta pesquisa obteve um resultado satisfatório na evolução do desempenho de classificação do uso da método NMF para classificação de dados textuais em um contexto de dados positivos e não rotulados. Considerando algoritmos do estado da arte aplicados a esse contexto, o NMFPUL e o Deep NMF demonstraram ser técnicas promissoras para a classificação de documentos e palavras de um texto em classes.



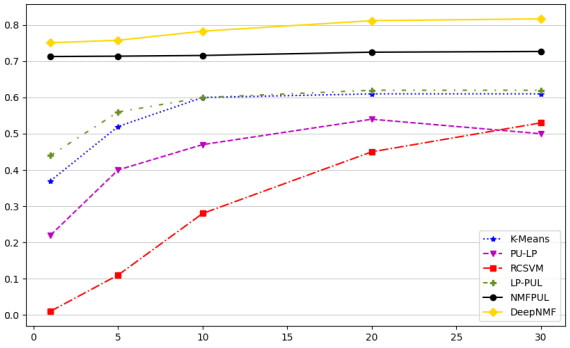
(a) CSTR



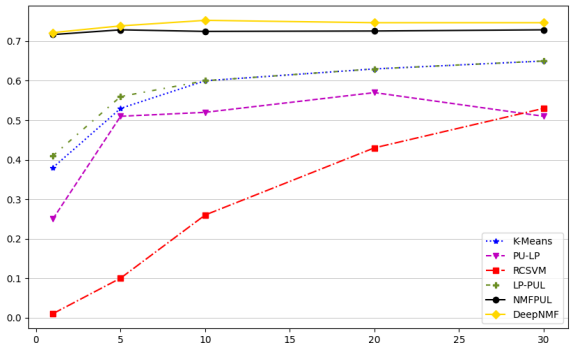
(b) Oh0



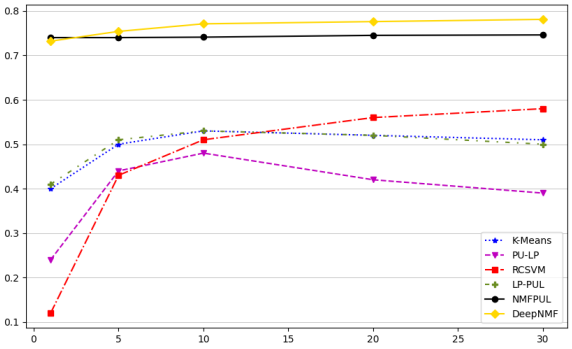
(c) Oh10



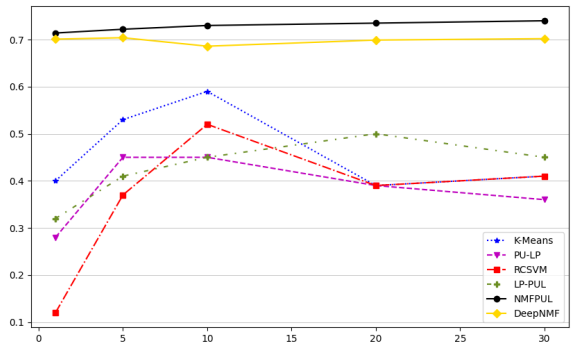
(d) Oh5



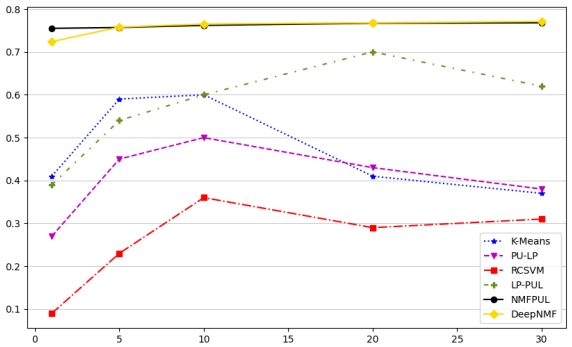
(e) Oh15



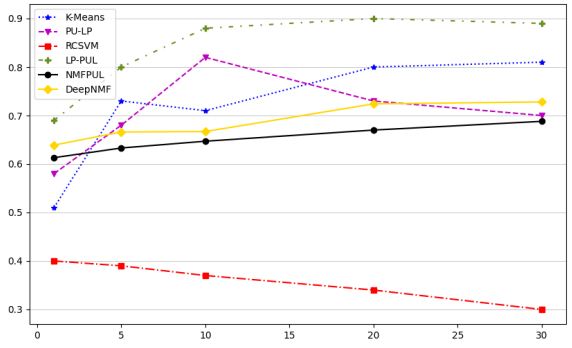
(f) Fbis



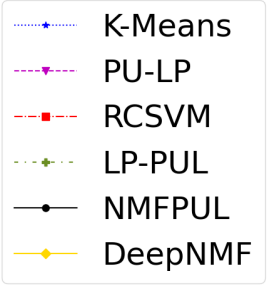
(g) Re0

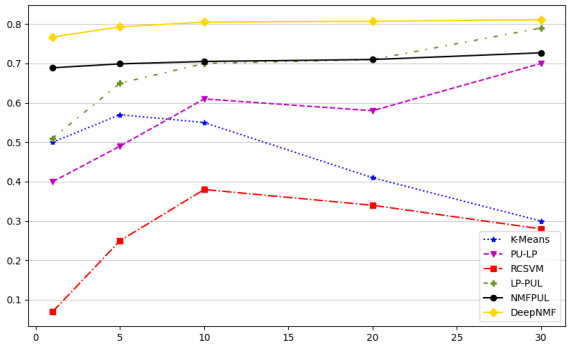


(h) Re1

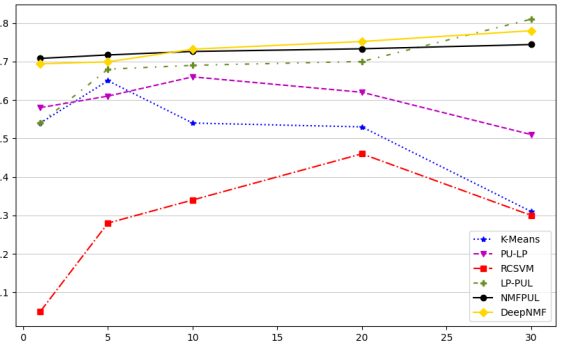


(i) SyskillWebert

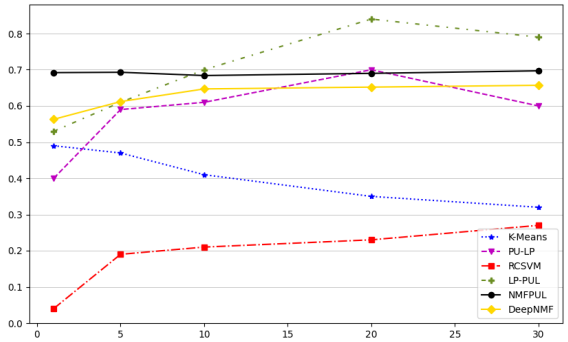




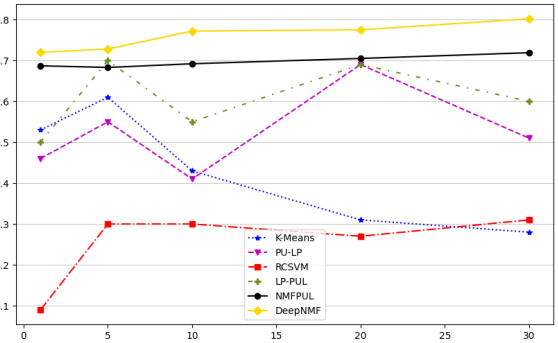
(a) Tr11



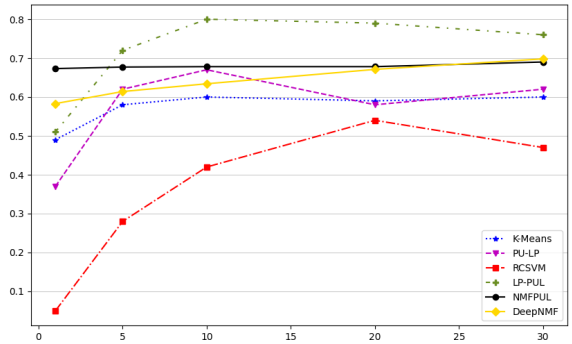
(b) Tr12



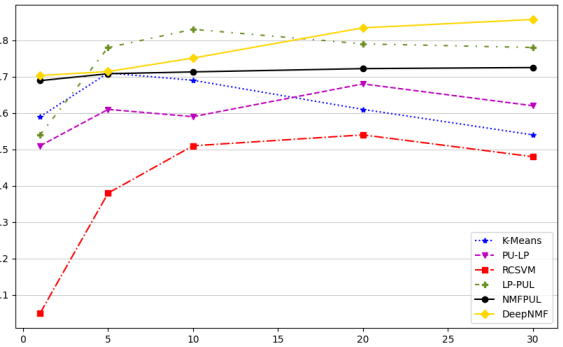
(c) Tr21



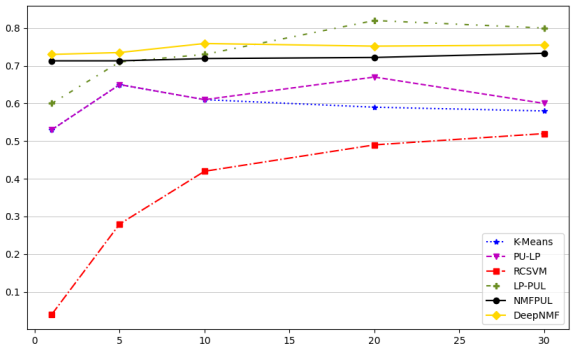
(d) Tr23



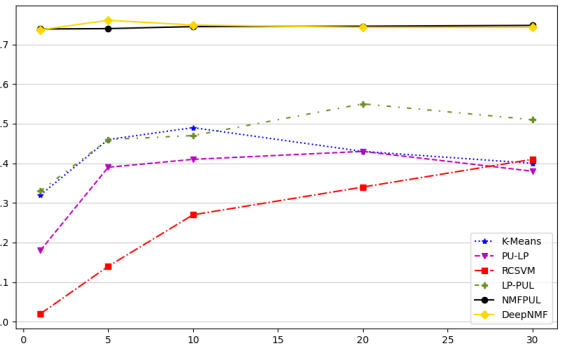
(e) Tr31



(f) Tr41



(g) Tr45



(h) Wap

Figura 5.1: Desempenho de classificação para os algoritmos definidos em comparação com o NMFPUL e o Deep NMF para as coleções de documentos selecionadas. O eixo X representa o número de documentos rotulados e o eixo Y representa o valor do F1 Score

Os dados de desempenho de classificação, medidos pelo *F1-Score*, foram analisados utilizando testes estatísticos para determinar se as diferenças no desempenho dos algoritmos selecionados são estatisticamente significativas. O teste de diferença de significância estatística (SSD – *Statistically significant differences (SSD)* considerou os diversos conjuntos de dados deste experimento. Esse é um teste recomendável de ser aplicado, visto que diferentes algoritmos podem ter diferentes comportamentos quando aplicados em diferentes contextos e em múltiplos datasets. Por isso, os testes são usados quando há um algoritmo controle, no nosso caso os algoritmos *NMFPUL* e o *Deep NMF*, e múltiplas coleções de dados.

Os valores de desempenho, que estão detalhados no Apêndice I, foram submetidos inicialmente ao **Teste de Friedman** (Pereira et al., 2015), usado para detectar diferenças significativas nos resultados de vários algoritmos sobre múltiplos conjuntos de dados, com 95% de confiança. Além disso, também foi aplicado o **teste *post-hoc* de Nemenyi** (Liang et al., 2011), que utiliza as diferenças significativas encontradas entre algoritmos para compará-los e identificar quais diferem significativamente entre si (Herbold, 2020).

Na **Tabela 5.3**, são apresentados os resultados do teste estatístico por meio do Ranking médio (*Average ranking* (*AR*)) e Ranking geral (*General ranking* (*GR*)). Os algoritmos NMFPUL e Deep NMF quase sempre obtiveram o melhor e o segundo melhor ranking de forma geral. O algoritmo Deep NMF foi o algoritmo que obteve a melhor ranking considerando as diversas quantidades de documentos rotulados por classe positiva, obtendo primeiro lugar no ranking exceto em um dos casos, quando o número de documentos rotulados na classe positiva é igual a um.

Tabela 5.3: Ranking médio (*Average ranking* (*AR*)) e Ranking geral (*General ranking* (*GR*)) para algoritmos selecionados nas múltiplas quantidades de documentos rotulados.

	1 doc rotulado		5 doc rotulados		10 doc rotulados		20 doc rotulados		30 doc rotulados	
Alg.	AR	GR	AR	GR	AR	GR	AR	GR	AR	GR
K-Means	3.882	4th	3.824	4th	3.912	4th	4.441	5th	4.529	4th
PU-LP	4.676	5th	4.382	5th	4.353	5th	4.294	4th	4.794	5th
RCSVM	6.000	6th	6.000	6th	5.824	6th	5.765	6th	5.353	6th
LP-PUL	3.147	3rd	2.735	3rd	2.735	3rd	2.412	3rd	2.324	2nd
NMFPUL	1.529	1st	2.176	2nd	2.294	2nd	2.206	2nd	2.353	3rd
Deep NMF	1.765	2nd	1.882	1st	1.882	1st	1.882	1st	1.647	1st

Após o Teste de *Friedman* atestar que existe uma diferença estatisticamente significativa entre os algoritmos LP-PUL, NMFPUL e o Deep NMF e o restante dos algoritmos, o teste post-hoc de *Nemenyi* foi realizado, teste esse que traz a diferença significativa entre os algoritmos, ou seja, uma comparação do ranking médio, para verificar aqueles que diferem significativamente entre si através da diferença crítica.

Em um diagrama de diferença crítica, foi verificado que algoritmos conectados através de uma barra horizontal de tamanho menor ou igual ao valor da diferença crítica não são estisticamente significativa. Na Figura 5.2, o diagrama indica que o *NMFPUL* e o *Deep NMF* não possuem diferença significativa entre si, assim como o LP-PUL. Já em relação aos algoritmos RCSVM, PU-LP e K-Means, NMFPUL e Deep NMF possuem uma diferença crítica siginificativa.

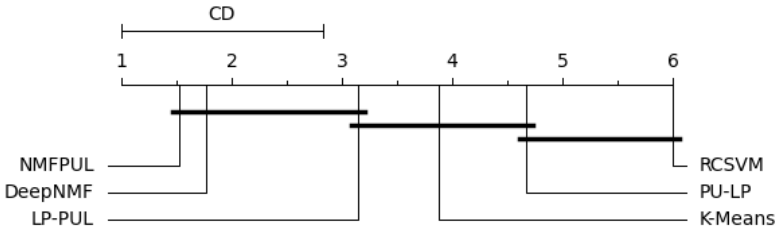


Figura 5.2: Diagrama de classificações de diferença crítica (CD) obtido com teste estatístico de post-hoc de Nemenyi para 1 documento rotulado

Nas Figuras 5.3, 5.4, 5.5 e 5.6 observa-se a classificação estatística, segundo o teste de Nemenyi, onde os algoritmos *Deep NMF*, NMFPUL e LP-PUL possuem significância estatística sobre os algoritmos RCSVM, PU-LP e *K-Means* em todas as quantidades de documentos rotulados.

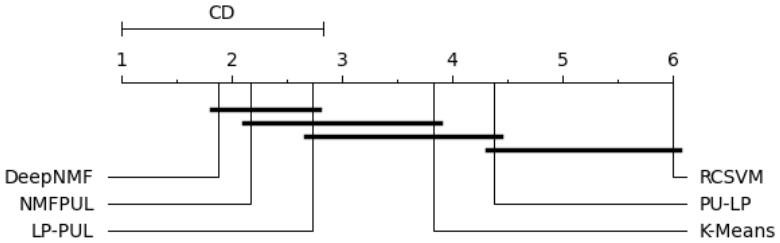


Figura 5.3: Diagrama de classificações de diferença crítica (CD) obtido com teste estatístico de post-hoc de Nemenyi para 5 documentos rotulados

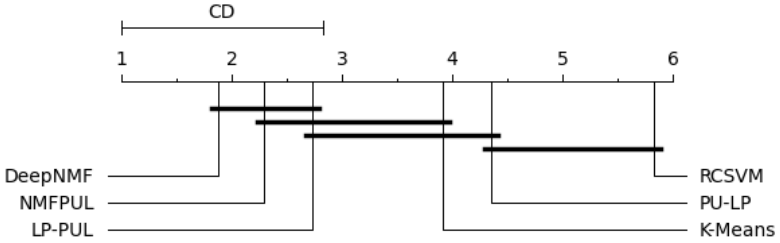


Figura 5.4: Diagrama de classificações de diferença crítica (CD) obtido com teste estatístico de post-hoc de Nemenyi para 10 documentos rotulados

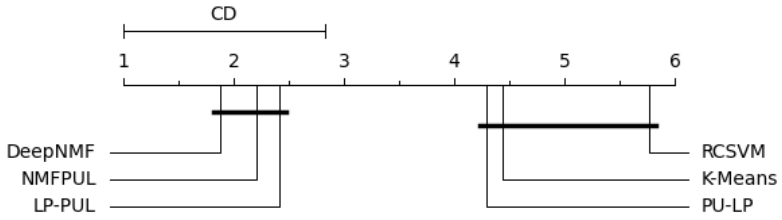


Figura 5.5: Diagrama de classificações de diferença crítica (CD) obtido com teste estatístico de post-hoc de Nemenyi para 20 documentos rotulados

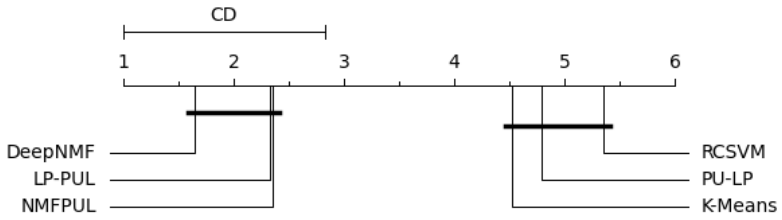


Figura 5.6: Diagrama de classificações de diferença crítica (CD) obtido com teste estatístico de post-hoc de Nemenyi para 30 documentos rotulados

Ademais, com a utilização de rede neural para incrementar a implementação do NMF, foi avaliada a possibilidade de que o processo de se utilizar as matrizes W e H de saída do treinamento da rede neural Deep NMF como entrada para a função de Atualização Multiplicativa no NMFPUL, conforme explicado na Seção 2.2.2, resulte em melhoria na convergência do algoritmo, ou seja, em menor uso de recursos computacionais no processo de classificação dos documentos. Nesse caso, foi realizado o cálculo do erro e da diferença entre erros consecutivos de iterações para verificar se esse valor atingia um patamar menor que um valor de tolerância definido, no caso definido em $[tol = 1e - 3]$. Caso esse valor atingisse um valor menor que a tolerância, o algoritmo poderia cessar as iterações para aquela execução.

Considerando a estrutura do experimento e processo de avaliação, conforme detalhado na Seção 4.5, ao realizar a operacionalização do *NMFPUL* e do *Deep NMF*, para a totalidade dos datasets selecionados para esse trabalho, serão realizadas, ao todo, 2550 execuções completas de cada algoritmo, para chegar aos resultados já ilustrados aqui.

Para o *NMFPUL*, em apenas 72 execuções o algoritmo realizou uma parada antes de atingir o número máximo de iterações definido. Já para o *Deep NMF*, em 729 execuções o algoritmo convergiu antes de atingir o número máximo de iterações, e em muitos desses casos, a convergência foi atingida logo após a primeira iteração. A distribuição de frequência da quantidade de iterações, excetuando as execuções que atingiram o número máximo de iterações, pode ser visualizada na **Figura 5.7**.

Como pode ser visto na **Figura 5.8**, o algoritmo *Deep NMF*, permite que as execuções

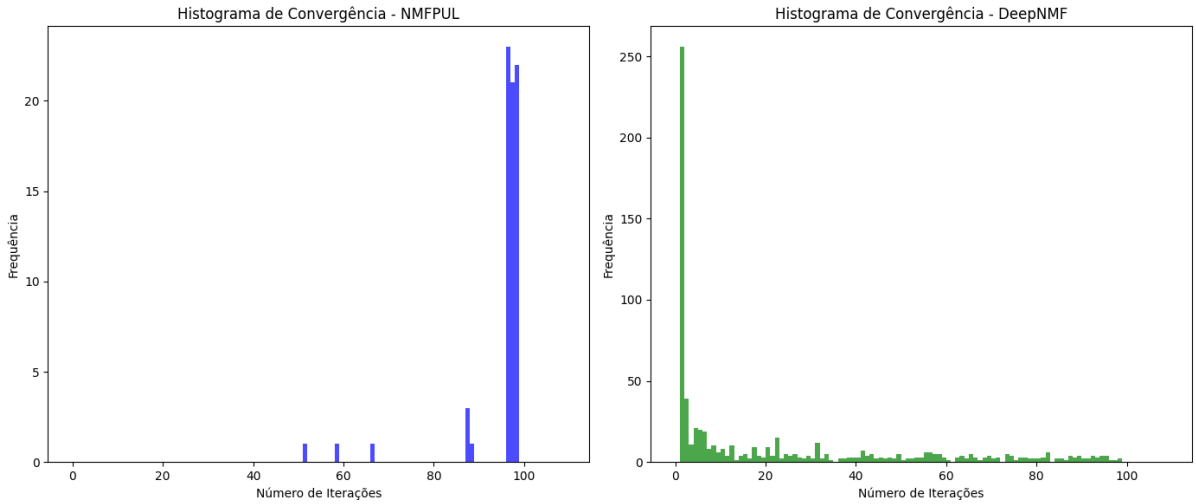


Figura 5.7: Histograma da quantidade de iterações para convergência para os algoritmos NMFPUL e Deep NMF - desconsiderando a frequência para a quantidade máxima de iterações

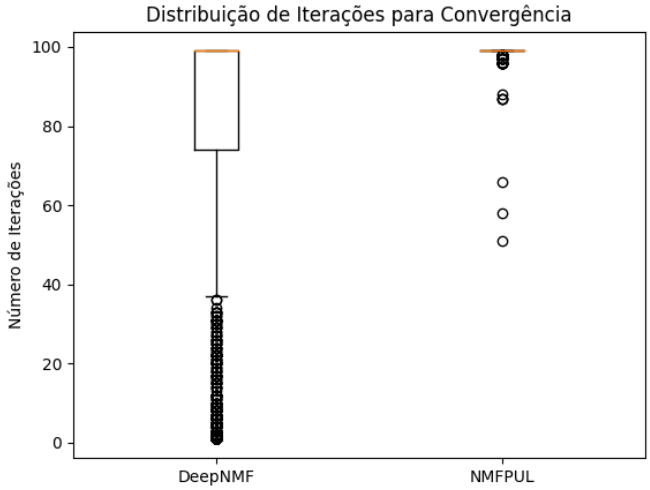


Figura 5.8: Boxplot da quantidade de iterações para convergência para os algoritmos NMFPUL e Deep NMF

sejam mais breves, pois atingem a convergência mais rapidamente, sendo consequência da utilização de matrizes W e H mais otimizadas para iniciar o processo de Atualização Multiplicativa. Por isso, pode ser notado que existe uma boa distribuição de execuções do *Deep NMF* onde o número de iterações necessárias para atingir o critério de parada é menor do que 40, diferentemente do NMFPUL, onde nenhuma das execuções conseguiu convergir com menos de 53 iterações.

Capítulo 6

Conclusão

Neste trabalho foram realizadas análises da aplicação de *positive unlabeled learning* em dados textuais e avaliada o desempenho no que se diz respeito a métricas de qualidade em classificações de texto. A hipótese avaliada, até aqui, nesse trabalho, é de que realizar modificações no método de redução de dimensionalidade , especificamente o NMF, para resolver problemas de ***Positive Unlabeled Learning*** é uma estratégia para superar o estado da arte na aplicação de PUL para classificação de textos. Existe na literatura abordagens em PUL que aplicam o NMF, mas apenas com o objetivo de reduzir a dimensionalidade do problema (Kaur et al., 2021) e para exploração de tópico/clusters, como em (Li et al., 2016). Assim, para validar a hipótese levantada, as questões de pesquisa a seguir orientam esse estudo:

- A aplicação do algoritmo *Non-Negative Matrix Factorization* introduz melhorias relevantes na classificação de texto?
- Existe uma forma de adaptação do NMF que direcionam esse algoritmo para realizar a classificação de dados do tipo texto, obtendo níveis satisfatórios de classificação dos dados?
- A aplicação do algoritmo de Fatoração de Matrizes Não-Negativas através de Redes Profundas *Deep Non-Negative Matrix Factorization* introduz melhorias relevantes na classificação de texto?

Para que a hipótese seja validada, algumas etapas foram definidas, tendo como foco responder às questões de pesquisa. Assim, uma revisão de literatura voltada para verificar os estudos realizados em *Positive Unlabeled Learning* aplicado a textos foi realizada, identificando aqueles que utilizavam técnicas de redução de dimensionalidade ou fatoração de matrizes, técnicas que poderiam ser base para o desenvolvimento da metodologia do presente trabalho, principalmente aqueles trabalhos que aplicavam o conceito de *Non-Negative Matrix Factorization*. A partir da revisão de literatura realizada, a proposta

foi definida em torno de um algoritmo, que modifica a estrutura do NMF para melhor classificar documentos de texto, lidando com o problema de não ter a maioria dos dados rotulados, o que é um problema recorrente em dados gerados pelas principais aplicações. A estrutura do nosso modelo de NMF adaptado fornece uma ponte para obter inferências matemáticas sobre a classificação de dados PU.

O NMFPUL foi aplicado em um conjunto de *datasets* possuindo diversificadas quantidades de documentos, classes e palavras, variando a quantidade de documentos rotulados do conjunto de dados, visando avaliar o desempenho de classificação dentro dessas diferentes situações. Realizamos um experimento substancial comparando o NMFPUL com os resultados de outros algoritmos e mostramos que nossa proposta pode superar alguns métodos de ponta para classificação de texto em dados PU. Em alguns casos, principalmente naqueles onde a quantidade de classes ou de documentos era maior, houve uma melhora de até 30% em relação ao algoritmo que possui o segundo melhor resultado. Já nos *datasets* com menor volume de classes ou documentos, o resultado foi similar a o resultados dos melhores algoritmos. Também observa-se que o NMFPUL performa acima da média dos outros algoritmos do estado da arte comparados quando a quantidade de documentos rotulados é menor, o que é positivo visto que nesses casos, teoricamente é mais díficil classificar acertadamente os dados quando a proporção de dados rotulados é menor.

Além disso, também foi contribuição desse trabalho, o desenvolvimento do algoritmo Deep NMF voltado para classificação de texto em uma situação de dados PU. Ao aproveitar as capacidades do Aprendizado Profundo, antecipamos o potencial de melhorar o desempenho e o aprendizado de representações em cenários de dados positivos e não rotulados. Para essa aplicação, foi detectada melhora na classificação de dados do tipo PU no contexto de dados do tipo textual, para a muitas das coleções de documentos avaliadas. O algoritmo forneceu resultados de classificação dos documentos em classes satisfatórios, considerando todas as instâncias de quantidade de dados rotulados, e também considerando a variação na combinação dos parâmetros da quantidade de componentes do processo de classificação e dos parâmetros de regularização da rede neural. Em alguns casos, houve desempenho de classificação próximo ao que o NMFPUL forneceu como resultado de classificação, mas em outros casos o Deep NMF propiciou melhorias no desempenho em algumas coleções de documentos de até 15%, em relação ao NMFPUL, principalmente para quantidades maiores de dados rotulados.

Também é importante concluir que o desenvolvimento de um algoritmo baseado em redes neurais aplicadas ao contexto de *Positive Unlabeled Learning* trouxe melhorias na velocidade de convergência do algoritmo, o que pode conferir, de forma prática, menor gasto de recursos computacionais e de tempo de execução. Como levantado na Seção 5, o

algoritmo NMFPUL, desenvolvido nesse trabalho, entregou bons resultados, comparado com os algoritmos baseline, em termos de desempenho de classificação dos documentos. Entretanto, foi verificado que, no processo de Atualização Multiplicativa, o algoritmo era executado até o número máximo de iterações, sendo raras as execuções onde um menor número de iterações era necessário para a convergência do algoritmo, seja por alcançar um erro muito baixo ou pela convergência lenta, alcançando uma estagnação. Dessa forma, o NMFPUL demandava quase na totalidade das execuções, de mais recurso computacional, visando atingir o passo final de metrificação do desempenho de classificação, considerando ainda que a formulação do NMFPUL realiza a execução do algoritmo por dez vezes para obter uma média de classificação mais justa, visto que alguns parâmetros são iniciados ou selecionados de forma randômica.

Após o desenvolvimento do algoritmo Deep NMF, verificou-se que, além de proporcionar um melhora no desempenho de classificação para algumas das coleções selecionadas para esse estudo, o algoritmo propiciou melhorias no que se refere à velocidade de convergência. Enquanto, no NMFPUL, aproximadamente **98%** das execuções atingiam a quantidade máxima de iterações definida, no Deep NMF, aproximadamente **71,1%** das execuções atingiam o limite máxima de iterações, ou seja, em 28,9% das execuções, o algoritmo necessitava de uma quantidade menor de iterações para atingir um convergência. E, também, o Deep NMF precisa, em média, de **77,93%** do limite máximo de iterações para convergir ou atingir uma estagnação. Isso denota que o fato de se utilizar matrizes W e H advindas da rede de treinamento do Deep NMF, ao invés de se utilizar valores randômicos para essas matrizes, conforme é feito no NMFPUL, traz uma melhoria no que tange à convergência do algoritmo. Esse fator tem como consequência, menor espaço de tempo para classificação dos documentos da coleção de documentos e também menor uso de recursos computacionais.

6.1 Trabalhos Futuros

No estágio atual do presente trabalho, oportunidades para estender seus resultados foram visualizadas:

Adaptação de outros algoritmos de redução de dimensionalidade: O presente estudo focou na adaptação do algoritmo NMF para classificação de dados positivos e não rotulados, entretanto outros algoritmos como PCA, LSA, CMF, podem conter caracterísiticas que também permitem a aplicação como método de classificação de dados textuais em problemas do tipo PUL.

Análise ampla da aplicação de algoritmos de redução de dimensionalidade em problemas do tipo PUL: Como detalhado na Seção 3.3.4, a maioria dos artigos

estudados não utilizavam técnicas de redução de dimensionalidade em algoritmos construídos para lidar com problemas do tipo PUL em dados do tipo texto. Logo, pode haver um espaço de estudo, quanto ao ganho de desempenho, não apenas de classificação, mas também de tempo e custo de memória, quando se utiliza alguma das diversas técnicas de redução de dimensionalidade existentes.

Estudo de técnicas variadas de aprendizado profundo em PUL: Assim, como está definida como parte dessa pesquisa, a avaliação de aprendizado profundo aplicado a problemas de classificação de dados textuais positivos e não rotulados oferece oportunidades de análise a partir de outros métodos de aplicação. Como foco futuro desse trabalho, pode-se citar a aplicação e adaptação do algoritmo Deep NMF para *datasets* do tipo PU, considerando outras modificações estruturais no algoritmo como: a inclusão de camadas lineares, a interconexão entres redes distintas considerando processos de atualização dos pesos para as matrizes W e H de forma separadas.

Referências

A. Abed, J. Yuan, L. Li, and M. K. Mesmin Junior. Research on adoption of gene-disease identification and recognition phenotype process. *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems*, 50, 09 2019. doi: 10.1109/HPCC/SmartCity/DSS.2019.00047. 46

C. Aggarwal and C. Zhai. *A survey of text classification algorithms.* Springer, 2012a. 2

C. Aggarwal and C. Zhai. *Mining Text Data.* Springer, 2012b. 2

D. Banerjee, G. Prabhat, and R. Bhowal. Imbalanced classification algorithm for semi supervised text learning (icasstle). *17th IEEE International Conference on Machine Learning and Applications*, 22, 11 2018. doi: 10.1109/ICMLA.2018.00165. 19, 43, 49

J. Bekker and J. Davis. Learning from positive and unlabeled data under the selected at random assumption. *Journal of Machine Learning Research*, 1, 08 2018. 15

J. Bekker and J. Davis. Learning from positive and unlabeled data: a survey. Springer Nature 2020, 2020. 3, 5, 11, 12, 13

J. Bekker, P. Robberechts, and J. Davis. Beyond the selected completely at random assumption for learning from positive and unlabeled data. *Journal of Machine Learning Research*, 1, 06 2019. 15, 19

P. Bian, L. Liu, and S. Penny. Detecting spam game reviews on steam with a semi-supervised approach. *Australian National University*, 06, 03 2021. doi: 10.1145/3472538.3472547. 19, 42

V. Bonnici. Kullback-leibler divergence between quantum distributions, and its upper-bound. 12 2020. 5

M. Caravanti de Souza, B. Nogueira, R. Rossi, R. Marcacini, B. Santos, and S. Rezende. A network-based positive and unlabeled learning approach for fake news detection. *Machine Learning*, 111, 11 2021. doi: 10.1007/s10994-021-06111-6. 44, 53, 54

J. C. Carnevali, R. Geraldelli Rossi, E. Milios, and A. de Andrade Lopes. A graph-based approach for positive and unlabeled learning. *Information Sciences 580 (2021)*, 580, 09 2021. doi: 10.1016/j.ins.2021.08.099. 3, 16, 53, 54, 56, 57, 61, 64

C. Carvalho, E. Silva de Moura, A. Veloso, and N. Ziviani. Website replica detection with distant supervision. *Springer Science+Business Media, LLC 2017*, 21, 07 2018. doi: 10.1007/s10791-017-9320-z. 45

H. Chen, F. Liu, Y. Wang, L. Zhao, and H. Wu. 2020. 39

J. D. Saunders and A. A. Freitas. Ga-auto-pu: A genetic algorithm-based automated machine learning system for positive-unlabeled learning. 05 2020. 40

T. de Paulo Faleiros, A. Valejo, and A. de Andrade Lopes. Unsupervised learning of textual pattern based on propagation in bipartite graph. Intelligent Data Analysis, 2020. 2

M. Du Plessis, G. Niu, and M. Sugiyama. Convex formulation for learning from positive and unlabeled data. *International conference on machine learning*, 14, 09 2015. doi: 10.1016/j.engappai.2024.108641. 53

A. Egorov, T. Sokhin, and N. Butakov. Towards a retrospective one-class oriented approach to parents detection in social media. *27TH CONFERENCE OF FRUCT ASSOCIATION*, 10, 09 2020. doi: 10.1109/ACCESS.2022.3219071. 43

E. Ferretti, L. Cagninga, V. Paiz, S. Delle Donne, R. Zacagnini, and M. Errecalde. Quality flaw prediction in spanish wikipedia: A case of study with verifiability flaws. *Information Processing and Management 54 (2018)*, 54, 08 2018. doi: 10.1016/j.ipm.2018.08.003. 45

J. Flenner and B. Hunter. A deep non-negative matrix factorization neural network. 1, 12 2017. 24, 49

S. Garg, Y. Wu, A. Smola, S. Balakrishnan, and Z. C. Lipton. Mixture proportion estimation and pu learning: A modern approach. *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, 653, 11 2024. doi: 10.48550/arXiv.2111.00980. 13

A. Gharahighehi, K. Pliakos, and C. Vens. Addressing the cold-start problem in collaborative filtering through positive-unlabeled learning and multi-target prediction. *Flemish Government (AI Research Program)*, 10, 09 2022. doi: 10.1109/ACCESS.2022.3219071. 43, 49

B. Ghasemkhani, K. Balbal, K. Birant, and D. Birant. A novel classification method: Neighborhood-based positive unlabeled learning using decision tree (npulud). *Entropy in Real-World Datasets and Its Impact on Machine Learning II*, 14, 05 2024. doi: 10.3390/e26050403. 48

C. Goutte and E. Gaussier. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. *Lecture Notes in Computer Science*, 3408, 06 2005. doi: 10.1007/978-3-540-31865-1_25. 51

M. Gôlo, M. Caravanti, R. Rossi, S. Rezende, B. Nogueira, and R. Maracin. Learning textual representations from multiple modalities to detect fake news through one-class learning. *WebMedia '21*, 10, 11 2021. doi: 10.1145/3470482.3479634. 1, 44

Z. Haj-Yahia, A. Sieg, and L. A. Deleris. Towards unsupervised text classification leveraging experts and word embeddings. Association for Computational Linguistics, 2019. 2

H. Hassani, C. Beneki, S. Unger, M. T. Mazinani, and M. R. Yeganegi. Text mining in big data analytics. Big Data Cognitive Computing, 2020. 2

D. He, M. Pan, K. Hong, Y. Cheng, S. Chan, X. Liu, and N. Guizani. Fake review detection based on pu learning and behavior density. *IEEE Network*, 92, 08 2020. doi: 10.1109/MNET.001.1900542. 3, 17, 42, 53

F. He, T. Liu, G. Webb, and D. Tao. Instance-dependent pu learning by bayesian optimal relabeling. 12, 10 2018. 19

S. Herbold. Autorank: A python package for automated ranking of classifiers. *Journal of Open Source Software*, 5, 04 2020. doi: 10.21105/joss.02173. 68

R. J. Hershey, J.R. and F. Weninger. Deep unfolding: Model-based inspiration of novel deep architectures. *Mach. Learn. 2014*, 04, 09 2014. doi: 10.48550/arXiv.1409.2574. 26

L. T. K. Hien and N. Gillis. Algorithms for nonnegative matrix factorization with the kullback-leibler divergence. *Journal of Scientific Computing*, 87, 10 2020. doi: 10.48550/arXiv.2010.01935. 6, 23, 62

W. Hu, R. Le, B. Liu, F. Ji, J. Ma, D. Zhao, and R. Yan. Predictive adversarial learning from positive and unlabeled data. *The Thirty-Fifth AAAI Conference on Artificial Intelligence*, 29, 02 2021. doi: 10.1111/coin.12329. 19, 43

A. Jacovi, G. Niu, Y. Goldberg, and M. Sugiyama. Scalable evaluation and improvement of document set expansion via neural positive-unlabeled learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, page 581–592. Association for Computational Linguistics, 2021. 46

Y. Jaemin, J. Kim, H. Yoon, G. Kim, C. Jang, and K. U. Graph-based pu learning for binary and multiclass classification without class prior. *Knowledge and Information Systems (2022)*, 10, 06 2022. doi: 10.1007/s10115-022-01702-8. 4, 53, 54

H. Jagadish, J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. Patel, R. Ramakrishnan, and C. Shahabi. Big data and technical challenges. ACM, 2014. 1

K. Jaskie and A. Spanias. Positive and unlabeled learning algorithms and applications: a survey. *SenSIP Center, School of ECEE*, 1, 12 2019. doi: 978-1-7281-4959-2. 3, 14, 15, 17

R. Jeffrey, P. Bian, F. Ji, and S. Penny. The wisdom of the gaming crowd. *Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 292, 11 2020. doi: 10.1145/3383668.3419915. 42

Z. Ji, C. Du, J. Jiang, L. Zhao, H. Zhang, and I. Ganchev. Improving non-negative positive-unlabeled learning for news headline classification. *IEEE Access*, 11, 04 2023. doi: 10.1109/ACCESS.2023.3269304. 53

C. Jiang, H.-F. Yu, C.-J. Hsieh, and K.-W. Chang. 2018. 38, 49

C. Jiang, J. Zhu, and Q. Xu. Which goods are most likely to be subject to click farming? an evidence from the taobao platform. *Electronic Commerce Research and Applications*, 50, 11 2021. doi: 10.1016/j.elerap.2021.101107. 18

Josh James. Data never sleeps 10.0, 2022. URL `https://www.domo.com/data-never-sleeps`. [Online; accessado em 24-Jan-2023]. 1

J. Jungmaier, N. Kassner, and B. Roth. 2020. 38, 49

M. Kato, T. Teshima, and J. Honda. Learning from positive and unlabeled data with a selection bias. *ICLR 2019*, 1, 09 2019. 15

R. Kaur, S. Singh, and H. Kumar. An intrinsic authorship verification technique for compromised account detection in social networks. *Soft Computing (2021)*, 10, 09 2021. doi: 10.1007/s00500-020-05445-y. 43, 49, 54, 72

J. Kim and H. Park. Fast nonnegative matrix factorization: An active-set-like method and comparisons. *SIAM J. Sci. Comput.*, 33, 04 2011. 26

B. Kitchenham, P. O. Brereton, D. Budgen, M. Turnera, J. Bailey, and S. Linkman. Systematic literature reviews in software engineering – a systematic literature review. *Journal of Information and Software Technology*, 51, 01 2009. doi: 10.1016/j.infsof.2008.09.009. 28

V. Korde. Text classification and classifiers: A survey. Iformation, 2012. 2

K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown. Text classification algorithms: A survey. Information, 2019. 2

R. Kyrio, G. Niu, M. C. du Plessis, and M. Sugiyama. Positive-unlabeled learning with non-negative risk estimator. *31st Conference on Neural Information Processing Systems - NIPS*, 1, 1 2017. 15, 53

D. Lee and H. Seung. Algorithms for non-negative matrix factorization. Neural Inf. Process. Syst., 2000. 22

M. Li, S. Pan, Y. Zhang, and X. Cai. Classifying networked text data with positive and unlabeled examples. Pattern Recognition Letters, 2016. 2, 4, 54, 72

Q. Li, H. Peng, J. Li, C. Xia, R. Yang, L. Yang, and S. P. S. Yu. A survey on text classification: From traditional to deep learning. ACM Transactions on Intelligent Systems and Technology, 2022. 1

X. Li and B. Liu. Learning to classify texts using positive and unlabeled data. volume 1, pages 587–592, 08 2003. 18, 53

G. Liang, X. Zhu, and C. Zhang. An empirical study of bagging predictors for different learning algorithms. *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 25, 11 2011. doi: 10.1609/aaai.v25i1.8026. 68

X. Lin and P. C. Boutros. Optimization and expansion of non-negative matrix factorization. BMC Bioinformatics, 2020. 23, 65

B. Liu, C. Liu, Y. Xiao, W. Li, and X. Chen. Adaboost-based transfer learning method for positive and unlabelled learning problem. *Knowledge-Based Systems*, 241, 01 2022a. doi: 10.1016/j.knosys.2022.108162. 18, 40

B. Liu, J. Liu, Y. Xiao, Q. Chen, K. Wang, R. Huang, and L. Li. A new self-paced learning method for privilege-based positive and unlabeled learning. *Information Sciences 609 (2022)*, 609, 07 2022b. doi: 10.1016/j.ins.2022.07.143. 18, 19, 44

H. Liu, P. Burnap, W. Alorainy, and M. L. Williams. Fuzzy multi-task learning for hate speech type identification. *IW3C2 (International World Wide Web Conference Committee),*, 10, 05 2019. doi: 10.1145/3308558.3313546. 43, 49

S. Ma and R. Zhang. Pu-lp: A novel approach for positive and unlabeled learning by label propagation. *2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, 01, 07 2017. doi: 10.1109/ICMEW.2017.8026296. 53

B. Mahesh. Machine learning algorithms - a review. *International Journal of Science and Research (IJSR)*, 9, 01 2020. doi: 10.21275/ART20203995. 10, 11

R. Mahmoodi, S. Seyedi, A. Abdollahpouri, and F. Tab. Enhancing link prediction through adversarial training in deep nonnegative matrix factorization. *Engineering Applications of Artificial Intelligence*, 133, 06 2024. doi: 10.1016/j.engappai.2024.108641. 50

E. G. Moayed H., Mansoori. Deep and wide nonnegative matrix factorization with embedded regularization. *Pattern Recognition*, 153, 06 2024. doi: 10.1016/j.patcog.2024.110530. 50

V. Monga, Y. Li, and Y. Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Sign. Process. Magaz*, 38, 08 2021. doi: 10.48550/arXiv.1912.10557. 26

G. Muric, A. Tregubov, J. Blythe, A. Abeliuk, D. Chouhary, and K. Lerman. Massive cross-platform simulations of online social networks. *AAMAS 2020*, 10, 09 2020. doi: 10.1109/ACCESS.2022.3219071. 43

B. Na, H. Kim, K. Song, W. Joo, Y.-Y. Kim, and I.-C. Moon. Deep generative positive-unlabeled learning under selection bias. 10 2020. 40

M. Naeem, T. Jamal, J. Diaz-Martinez, N. A. Butt, S.and Montesano, M. I. Tariq, E. De-la Hoz-Franco, and E. De-la Hoz-Valdiris. Trends and future perspective challenges in big data. 2021. 1

R. Nasser, Y. C. Eldar, and R. Sharan. Deep unfolding for non-negative matrix factorization with application to mutational signature analysis. Department of Math and Computer Science, Weizmann Institute of Science, Rehovot, Israel, 2021. 5, 23, 24, 26, 50, 59

K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using machine learning. 2000. 2

J. Niu, Z. Sun, and W. Zhang. Enhancing knowledge graph completion with positive unlabeled learning. *24th International Conference on Pattern Recognition (ICPR)*, 24, 08 2018. doi: 10.1007/s11704-020-9240-8. 41

C. H. Park. Multi-class positive and unlabeled learning for high dimensional data based on outlier detection in a low dimensional embedding space. *Center for Information and Language Processing*, 19, 05 2022. 39

D. G. Pereira, A. Afonso, and F. Medeiros. Overview of friedman's test and post-hoc analysis. *Communications in Statistics - Simulationand Computation*, 44, 11 2015. doi: 10.1080/03610918.2014.931971. 68

K. Pham, A. Santos, and J. Freire. Bootstrapping domain-specific content discovery on the web. *International World Wide Web Conference Committee*, 01, 05 2019. doi: 10.1145/3308558.3313709. 40

F. Z. Qachfar, R. M. Verma, and A. Mukherjee. Leveraging synthetic data and pu learning for phishing email detection. *Proceedings of the Twelveth ACM Conference on Data and Application Security and Privacy (CODASPY '22)*, 12, 04 2022. 5, 19

W. A. Qader, M. M. Ameen, and B. I. Ahmed. An overview of bag of words;importance, implementation, applications, and challenges. *2019 International Engineering Conference (IEC)*, 1, 06 2019. doi: 10.1109/IEC47844.2019.8950616. 20

R. G. Rossi, R. M. Marcacini, and S. O. Rezende. Benchmarking text collections for classication and clustering tasks. Technical report, 11 2013. 8, 56

S. Shalev-Schwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014. 10, 11

C. Shi, J. Ding, X. Cao, L. Hu, B. Wu, and X. Li. Entity set expansion in knowledge graph: a heterogeneous information network perspective. *2nd International Conference on Pattern Recognition and Machine Learning*, 15, 04 2021. doi: 10.1007/s11704-020-9240-8. 41

Y. Shuqin and F. Jing. Fake reviews detection based on text feature and behavior feature. *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems*, 12, 08 2019. doi: 10.1109/HPCC/SmartCity/DSS.2019.00277. 19, 41

T. Sidorowicz, P. Peres, and Y. Li. A novel approach for cross-selling insurance products using positive unlabelled learning. *2022 International Joint Conference on Neural Networks (IJCNN)*, 11, 04 2022. doi: 10.1109/IJCNN55064.2022.9892762. 45

M. M. Silva. Uma abordagem evolucionária para aprendizado semi-supervisionado em máquinas de vetores de suporte. Master's thesis, Escola de Engenharia - UFMG, Belo Horizonte, 2008. 11

M. C. Souza, M. P. S. Gôlo, A. M. G. Jorge, E. C. F. Amorim, R. N. T. Campos, R. M. Marcacini, and S. O. Rezende. Keywords attention for fake news detection using few positive labels. *Information Sciences*, 663, 03 2024. doi: 10.1016/j.ins.2024.120300. 45

P. Ta, M. Steinbach, A. Karpatne, and V. Kumar. *Anomaly Detection*. Pearson, 2019. 57

U. Tanielian and F. Vasile. Relaxed sofmax for pu learning. *Thirteenth ACM Conference on Recommender Systems (RecSys '19)*, 13, 09 2019. doi: 10.1145/3298689.3347034. 49

P. Teisseyre. Classifier chains for positive unlabelled multi-label learning. *Knowledge-Based Systems*, 213, 01 2021. doi: 10.1016/j.knosys.2020.106709. 39

J. Tollefson. China declared world's largest producer of scientific articles, 01 2018. 34

J. E. van Engelen and H. H. Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, Feb 2020. ISSN 1573-0565. doi: 10.1007/s10994-019-05855-6. URL `https://doi.org/10.1007/s10994-019-05855-6`. 2

B. P. W. Chen, Q. Zeng. A survey of deep nonnegative matrix factorization. Neurocomputing Volume 491, 28 June 2022, Pages 305-320, 2022. 5, 24, 25, 50

G. Wang, Z. Yu, Y. Xian, and Y. Zhang. Case-related news filtering via topic-enhanced positive-unlabeled learning. *Journal of Information Processing Systems*, 17, 12 2021. doi: 10.3745/JIPS.01.0081. 44

J.-Y. Wang and X.-L. Zhang. Deep nmf topic modeling. *Neurocomputing*, 515, 01 2023. doi: 10.1016/j.neucom.2022.10.002. 24, 50

S. Wang, M. Zhou, S. Mazumder, B. Liu, and Y. Chang. Disentangling aspect and opinion words in target-based sentiment analysis using lifelong learning. *Yahoo! Research*, 12, 03 2020. doi: 10.1109/HPCC/SmartCity/DSS.2019.00277. 42

Z. Wang, J. Jiang, and G. Long. Positive unlabeled learning by semi-supervised learning. *Australian Artificial Intelligence Institute,*, 213, 01 2022. doi: 10.1109/ICIP46576.2022.9897738. 39

M. Wu, S. Pan, L. Du, and X. Zhu. Learning graph neural networks with positive and unlabeled nodes. *ACM Trans. Knowl. Discov*, 101, 05 2021. doi: 10.1145/3450316. 16, 41

X. Wu, X. Zhu, G. Wu, and W. Ding. Data mining with big data. IEEE Trans. Knowl. Data Eng, 2014. 1

Z. Wu, J. Cao, Y. Wang, Y. Wang, L. Zhang, and J. Wu. hpsd: A hybrid pu-learning-based spammer detection model for product reviews. *IEEE TRANSACTIONS ON CYBERNETICS*, 50, 04 2020. doi: 10.1109/TCYB.2018.2877161. 19, 42

Y. Xiao and M. Watson. Guidance on conducting a systematic literature review. *Journal of Planning Education and Research*, 39, 08 2019. doi: 10.1177/0739456X17723971. 28

Y. Xu, L. Li, J. Huang, Y. Yin, W. Shao, Z. Mai, and L. Hei. Positive-unlabeled learning for sentiment analysis with adversarial training. *Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 292, 03 2019. doi: 10.1007/978-3-030-30146-0_25. 42

Y. Xu, L. Li, H. Gao, L. Hei, R. Li, and Y. Wang. Sentiment classification with adversarial learning and attention mechanism. *Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 292, 06 2020. doi: 10.1111/coin.12329. 5, 42

F. Yang, E. Dragut, and A. Mukherjee. Claim verification under positive unlabeled learning. *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 92, 12 2020. doi: 10.1109/ASONAM49781.2020.9381336. 43, 53

P. Yang, J. T. Ormerod, W. Liu, C. Ma, A. Y. Zomaya, and J. Y. H. Yang. Adasampling for positive-unlabeled and label noise learning with bioinformatics applications. *IEEE TRANSACTIONS ON CYBERNETICS*, 55, 03 2018. doi: 10.1109/TCYB.2018.2816984. 19, 46

J. Yoo, J. Kim, H. Yoon, G. Kim, C. Jang, and U. Kang. Accurate graph-based pu learning without class prior. *International Conference on Data Mining (ICDM)*, 24, 08 2021. doi: 10.1109/ICDM51629.2021.00094. 41

Z. Yun-tao, G. Ling, and W. Yong-cheng. An improved tf-idf approach for text classification. *Journal of Zhejiang University-SCIENCE A*, 6, 08 2005. doi: 10.1631/BF02842477. 20

D. Zhang, J. Yin, X. Zhu, and C. Zhang. Search efficient binary network embedding. *ACM Transactions on Knowledge Discovery Data*, 15, 05 2021. doi: 10.1145/3436892. 49

J. Zhang and P. S. Yu. Broad learning: An emerging area in social network analysis. *SIGKDD Explorations*, 20, 09 2018. doi: 10.1109/ACCESS.2022.3219071. 43

L. Zhang, Z. Wu, and J. Cao. Detecting spammer groups from product reviews: A partially supervised learning model. *National Key Technologies Research and Development Program of China*, 292, 02 2018. doi: 10.1109/ACCESS.2017.2784370. 5, 41

Z. S. H. W. Z. X. Zhang, S. A web semantic-based text analysis approach for enhancing named entity recognition using pulearning and negative sampling. *International Journal on Semantic Web and Information Systems*, 20, 02 2024. doi: 10.4018/IJSWIS.335113. 18

H. Zheng, H. Yu, Y. Hao, Y. Wu, and S. LI. Distantly supervised named entity recognition with spy-pu algorithm. *2nd International Conference on Pattern Recognition and Machine Learning*, 213, 01 2021. doi: 10.1109/PRML52754.2021.9520707. 41

Zoya, S. Latif, F. Shafait, and R. Latif. Analyzing lda and nmf topic models for urdu tweets via automatic labeling. *IEEE Access*, 9, 09 2021. doi: 10.1109/ACCESS.2021.3112620. 21

Apêndice I

Resultados de performance de classificacão para todos os algoritmos e datasets

Tabela I.1: Valores de F1 Score para diferentes algoritmos na Coleção de Documentos CSTR

Algoritmo	***# Documentos Rotulados - CSTR***				
	1	5	10	20	30
K-Means	0.49	0.61	0.68	0.65	0.51
PU-LP	0.55	**0.74**	**0.78**	0.7	0.69
RCSVM	0.02	0.12	0.29	0.52	0.39
LP-PUL	0.61	0.69	0.77	**0.79**	**0.8**
NMFPUL	**0.632**	0.679	0.715	0.735	0.741
Deep NMF	0.457	0.577	0.710	0.725	0.762

Tabela I.2: Valores de F1 Score para diferentes algoritmos na Coleção de Documentos Oh0

Algoritmo	***# Documentos Rotulados - Oh0***				
	1	5	10	20	30
K-Means	0.42	0.61	0.68	0.71	0.69
PU-LP	0.28	0.54	0.61	0.6	0.59
RCSVM	0.01	0.19	0.37	0.5	0.59
LP-PUL	0.51	0.67	0.7	0.73	0.71
NMFPUL	**0.707**	**0.726**	**0.744**	**0.761**	**0.771**
Deep NMF	**0.706**	0.709	0.712	0.713	0.721

Tabela I.3: Valores de F1 Score para diferentes algoritmos na Coleção de Documentos Oh5

Algoritmo	***# Documentos Rotulados - Oh5***				
	1	5	10	20	30
K-Means	0.41	0.60	0.68	0.71	0.70
PU-LP	0.26	0.52	0.62	0.55	0.54
RCSVM	0.01	0.11	0.30	0.5	0.58
LP-PUL	0.50	0.63	0.70	0.72	0.71
NMFPUL	0.710	0.722	0.739	**0.752**	0.765
Deep NMF	**0.729**	**0.739**	**0.749**	**0.755**	**0.774**

Tabela I.4: Valores de F1 Score para diferentes algoritmos na Coleção de Documentos Oh10

Algoritmo	***# Documentos Rotulados - Oh10***				
	1	5	10	20	30
K-Means	0.37	0.52	0.60	0.61	0.61
PU-LP	0.22	0.40	0.47	0.54	0.50
RCSVM	0.01	0.11	0.28	0.45	0.53
LP-PUL	0.44	0.56	0.60	0.62	0.62
NMFPUL	0.713	0.714	0.716	0.725	0.727
Deep NMF	**0.751**	**0.758**	**0.783**	**0.812**	**0.817**

Tabela I.5: Valores de F1 Score para diferentes algoritmos na Coleção de Documentos Oh15

Algoritmo	***# Documentos Rotulados - Oh15***				
	1	5	10	20	30
K-Means	0.38	0.53	0.60	0.63	0.65
PU-LP	0.25	0.51	0.52	0.57	0.51
RCSVM	0.01	0.10	0.26	0.43	0.53
LP-PUL	0.41	0.56	0.60	0.63	0.65
NMFPUL	**0.717**	0.729	0.725	0.726	0.729
Deep NMF	**0.722**	**0.739**	**0.753**	**0.747**	**0.747**

Tabela I.6: Valores de F1 Score para diferentes algoritmos na Coleção de Documentos Fbis

Algoritmo	***# Documentos Rotulados - Fbis***				
	1	5	10	20	30
K-Means	0.40	0.50	0.53	0.52	0.51
PU-LP	0.24	0.44	0.48	0.42	0.39
RCSVM	0.12	0.43	0.51	0.56	0.58
LP-PUL	0.41	0.51	0.53	0.52	0.5
NMFPUL	**0.740**	0.740	0.741	0.745	0.746
Deep NMF	0.732	**0.754**	**0.771**	**0.776**	**0.781**

Tabela I.7: Valores de F1 Score para diferentes algoritmos na Coleção de Documentos Re0

Algoritmo	***# Documentos Rotulados - Re0***				
	1	5	10	20	30
K-Means	0.40	0.53	0.59	0.39	0.41
PU-LP	0.28	0.45	0.45	0.39	0.36
RCSVM	0.12	0.37	0.52	0.39	0.41
LP-PUL	0.32	0.41	0.45	0.50	0.45
NMFPUL	**0.714**	**0.722**	**0.730**	**0.735**	**0.740**
Deep NMF	0.701	0.704	0.686	0.699	0.702

Tabela I.8: Valores de F1 Score para diferentes algoritmos na Coleção de Documentos Re1

Algoritmo	***# Documentos Rotulados - Re1***				
	1	5	10	20	30
K-Means	0.41	0.59	0.60	0.41	0.37
PU-LP	0.27	0.45	0.50	0.43	0.38
RCSVM	0.09	0.23	0.36	0.29	0.31
LP-PUL	0.39	0.54	0.60	0.70	0.62
NMFPUL	**0.755**	**0.757**	**0.762**	**0.767**	**0.768**
Deep NMF	0.724	**0.758**	**0.765**	**0.768**	**0.771**

Tabela I.9: Valores de F1 Score para diferentes algoritmos na Coleção de Documentos SyskillWebert

Algoritmo	***# Documentos Rotulados - SyskillWebert***				
	1	5	10	20	30
K-Means	0.51	0.73	0.71	0.80	0.81
PU-LP	0.58	0.68	0.82	0.73	0.70
RCSVM	0.40	0.39	0.37	0.34	0.30
LP-PUL	**0.69**	**0.80**	**0.88**	**0.90**	**0.89**
NMFPUL	0.613	0.633	0.647	0.670	0.688
Deep NMF	0.639	0.666	0.667	0.724	0.728

Tabela I.10: Valores de F1 Score para diferentes algoritmos na Coleção de Documentos Tr11

Algoritmo	***# Documentos Rotulados - Tr11***				
	1	5	10	20	30
K-Means	0.50	0.57	0.55	0.41	0.30
PU-LP	0.40	0.49	0.61	0.58	0.70
RCSVM	0.07	0.25	0.38	0.34	0.28
LP-PUL	0.51	0.65	0.70	0.71	0.79
NMFPUL	0.689	0.699	0.705	0.71	0.727
Deep NMF	**0.767**	**0.793**	**0.805**	**0.807**	**0.811**

Tabela I.11: Valores de F1 Score para diferentes algoritmos na Coleção de Documentos Tr12

Algoritmo	***# Documentos Rotulados - Tr12***				
	1	5	10	20	30
K-Means	0.54	0.65	0.54	0.53	0.31
PU-LP	0.58	0.61	0.66	0.62	0.51
RCSVM	0.05	0.28	0.34	0.46	0.30
LP-PUL	0.54	0.68	0.69	0.70	**0.81**
NMFPUL	**0.708**	**0.717**	**0.726**	0.733	0.744
Deep NMF	0.694	0.699	**0.732**	**0.752**	0.780

Tabela I.12: Valores de F1 Score para diferentes algoritmos na Coleção de Documentos Tr21

Algoritmo	***# Documentos Rotulados - Tr21***				
	1	5	10	20	30
K-Means	0.49	0.47	0.41	0.35	0.32
PU-LP	0.40	0.59	0.61	0.70	0.60
RCSVM	0.04	0.19	0.21	0.23	0.27
LP-PUL	0.53	0.61	**0.70**	**0.84**	**0.79**
NMFPUL	**0.692**	**0.693**	0.684	0.690	0.697
Deep NMF	0.563	0.612	0.647	0.652	0.657

Tabela I.13: Valores de F1 Score para diferentes algoritmos na Coleção de Documentos Tr23

Algoritmo	***# Documentos Rotulados - Tr23***				
	1	5	10	20	30
K-Means	0.53	0.61	0.43	0.31	0.28
PU-LP	0.46	0.55	0.41	0.69	0.51
RCSVM	0.09	0.30	0.30	0.27	0.31
LP-PUL	0.50	0.70	0.55	0.69	0.60
NMFPUL	0.687	0.683	0.692	0.705	0.719
Deep NMF	**0.720**	**0.728**	**0.772**	**0.775**	**0.802**

Tabela I.14: Valores de F1 Score para diferentes algoritmos na Coleção de Documentos Tr31

Algoritmo	***# Documentos Rotulados - Tr31***				
	1	5	10	20	30
K-Means	0.49	0.58	0.60	0.59	0.60
PU-LP	0.37	0.62	0.67	0.58	0.62
RCSVM	0.05	0.28	0.42	0.54	0.47
LP-PUL	0.51	**0.72**	**0.80**	**0.79**	**0.76**
NMFPUL	**0.673**	0.677	0.678	0.678	0.690
Deep NMF	0.583	0.614	0.634	0.671	0.698

Tabela I.15: Valores de F1 Score para diferentes algoritmos na Coleção de Documentos Tr41

Algoritmo	***# Documentos Rotulados - Tr41***				
	1	5	10	20	30
K-Means	0.59	0.71	0.69	0.61	0.54
PU-LP	0.51	0.61	0.59	0.68	0.62
RCSVM	0.05	0.38	0.51	0.54	0.48
LP-PUL	0.54	**0.78**	**0.83**	0.79	0.78
NMFPUL	0.689	0.708	0.713	0.722	0.725
Deep NMF	**0.703**	0.714	0.751	**0.834**	**0.857**

Tabela I.16: Valores de F1 Score para diferentes algoritmos na Coleção de Documentos Tr45

Algoritmo	***# Documentos Rotulados - Tr45***				
	1	5	10	20	30
K-Means	0.53	0.65	0.61	0.59	0.58
PU-LP	0.53	0.65	0.61	0.67	0.60
RCSVM	0.04	0.28	0.42	0.49	0.52
LP-PUL	0.60	0.71	0.73	**0.82**	**0.80**
NMFPUL	0.713	0.713	0.719	0.722	0.733
Deep NMF	**0.730**	**0.735**	**0.759**	0.752	0.755

Tabela I.17: Valores de F1 Score para diferentes algoritmos na Coleção de Documentos Wap

Algoritmo	***# Documentos Rotulados - Wap***				
	1	5	10	20	30
K-Means	0.32	0.46	0.49	0.43	0.40
PU-LP	0.18	0.39	0.41	0.43	0.38
RCSVM	0.02	0.14	0.27	0.34	0.41
LP-PUL	0.33	0.46	0.47	0.55	0.51
NMFPUL	**0.739**	0.740	**0.745**	**0.746**	**0.748**
Deep NMF	**0.737**	**0.761**	**0.749**	**0.743**	0.743