UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Detecção de notícias falsas usando poucos dados positivos rotulados

Mariana Caravanti de Souza

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP SÃO CARLOS

Mariana Caravanti de Souza

Detecção de notícias falsas usando poucos dados positivos rotulados

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutora em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Solange Oliveira Rezende
Coorientador: Prof. Dr. Alípio Mário Guedes Jorge

USP – São Carlos
Novembro de 2023

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

Caravanti de Souza, Mariana
C719d Detecção de notícias falsas usando poucos dados positivos rotulados / Mariana Caravanti de Souza; orientador Solange Oliveira Rezende; coorientador Alípio Mário Guedes Jorge. -- São Carlos, 2023.
157 p.

Tese (Doutorado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2023.

1. Detecção de Notícias Falsas. 2. Aprendizado de Uma Única Classe. 3. Aprendizado Positivo e Não Rotulado. 4. Redes Heterogêneas. 5. Aprendizado Semissupervisionado. I. Oliveira Rezende, Solange, orient. II. Mário Guedes Jorge, Alípio, coorient. III. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Mariana Caravanti de Souza

Fake news detection using few positive labels

Thesis submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP – in accordance with the requirements of the Computer and Mathematical Sciences Graduate Program, for the degree of Doctor in Science. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Solange Oliveira Rezende
Co-advisor: Prof. Dr. Alípio Mário Guedes Jorge

USP – São Carlos
November 2023

Aos meus pais dedico esta tese, que embarcaram comigo em todas as grandes mudanças que propus, me dando apoio incondicional em todo o percurso. Sem vocês, eu nada seria.

AGRADECIMENTOS

Agradeço a Deus, que sempre me deu coragem e bom ânimo para seguir em frente. Agradeço aos meus pais, Jorge e Marli, minha base sólida, por me ensinarem tantos valores e me apoiarem em todas as decisões importantes. Cada conquista ao longo desta jornada é reflexo da dedicação e carinho que sempre recebi de vocês. Agradeço ao meu irmão, Gabriel, que morou comigo parte do período de doutorado, sendo um apoio em dias difíceis. Agradeço também ao meu parceiro, Thiago, que segurou minha mão em todo o percurso, e mesmo com a distância, sempre me incentivou a cumprir meus objetivos.

Agradeço aos meus professores da Universidade Federal de Mato Grosso do Sul, Edson Cáceres, Francisco Vasconcellos e Bruno Magalhães, pelo incentivo e empurrões que me deram na direção da Universidade de São Paulo. Agradeço a minha orientadora, Solange Rezende, por ter me abraçado como aluna. Foram os quatro anos de maiores aprendizados da minha vida. Embora tenha muito a melhorar, me desenvolvi como pessoa, aprendi a ter mais foco, autoconfiança, estar presente, além de vivenciar a dedicação e o olhar que tem com cada um de seus alunos. Melhor exemplo do que é ser professora, eu não poderia ter.

Agradeço ao meu co-orientador Alípio Jorge, que tão bem me recebeu na Universidade do Porto, a qual vivi momentos inesquecíveis. Obrigada por alavancar minha pesquisa e pelo privilégio de trabalhar com a sua "equipa".

Agradeço aos bons amigos que fiz em São Carlos, que tornaram meus dias mais leves, Allan, Isadora, Anderson e Thomaz, além dos amigos de longa data, Nícolas e Roberta. Obrigada ao Brucce por me ensinar novos códigos e *scripts* em Python, e ao Marcos pelos artigos em parceria. Obrigada também a dona Amélia, que me acolheu como sua neta e por vezes me preparava café da manhã. Todos vocês foram fundamentais para eu concluir minha jornada no doutorado.

Agradeço ao Instituto de Ciências Matemáticas e de Computação e ao Laboratório de Inteligência Computacional, por me oferecerem um ambiente de estudos e trocas inigualável. Agradeço também as agências financiadoras que me apoiaram para elaboração desta tese: Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Códigos de Financiamento nº. 88887.357712/2019-00, 88887.695327/2022-00, 88887.827602/2023-00, Fundação de Amparo à Pesquisa do Estado de São Paulo - Brasil (FAPESP) - Códigos de Financiamento nº. 2019/25010-5, 2019/07665-4, e Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil (CNPq) - Código de Financiamento nº. 309575/2021-4.

“As invenções são, sobretudo,
o resultado de um trabalho de teimoso.”
(Santos Dumont)

RESUMO

SOUZA, M. C. **Detecção de notícias falsas usando poucos dados positivos rotulados**. 2023. 157 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2023.

O avanço da tecnologia tem permitido a disseminação de notícias falsas em larga escala. Criadas para manipular a perspectiva de usuários, atrair sua postura ideológica e instigá-los a compartilhar a informação, notícias falsas vêm se tornando mais suscetíveis a enganar o público alvo. Métodos de Aprendizado de Máquina têm sido utilizados como estratégia promissora para auxiliar na detecção de conteúdo falso, cujo problema geralmente é modelado com algoritmos de aprendizado binário ou multiclasse. No entanto, um dos desafios é definir um conjunto de notícias representativo e conciso para treinar os algoritmos, devido (i) ao desbalanceamento naturalmente latente entre a quantidade de notícias verdadeiras e falsas disponíveis; (ii) a dinamicidade na qual notícias falsas evoluem, cada vez mais convincentes e semelhantes a notícias verídicas; (iii) além da dificuldade em se rotular uma grande quantidade de notícias, sendo necessário a checagem de cada fato relatado no conteúdo da publicação. Considerando a dificuldade na rotulação de notícias falsas (exemplos de interesse, ou positivos) enquanto a caracterização de notícias verdadeiras é ampla (exemplos não interessantes, ou negativos), neste projeto é proposta uma abordagem para detecção de notícias falsas que caracteriza o problema por meio de Aprendizado de Uma Única Classe (OCL). Algoritmos OCL aprendem modelos de classificação considerando apenas informações da classe de interesse. Além disso, métodos de Aprendizado Positivo e Não Rotulado (PUL) utilizam informações de dados não rotulados com o intuito de aumentar o desempenho de classificação. Neste trabalho são propostas abordagens baseadas no algoritmo Positive and Unlabeled Learning by Label Propagation (PU-LP), um algoritmo PUL baseado em redes de similaridade. PU-LP identifica potenciais exemplos da classe positiva e negativa, e posteriormente um algoritmo semissupervionado realiza a classificação dos demais nós não rotulados. São avaliadas diferentes configurações de rede e algoritmos de classificação semissupervisionados em seis bases de notícias que apresentam cenários distintos quanto a linguagem, tópicos, tipo de coleta e balanceamento entre as classes. Experimentos indicam que redes compostas por notícias e termos representativos podem beneficiar o desempenho da abordagem, que é capaz de indetificar notícias falsas com até 94% de F_1 usando 10% de dados positivos rotulados.

Palavras-chave: Detecção de notícias falsas, aprendizado de uma única classe, aprendizado positivo e não rotulado, redes heterogêneas, aprendizado semissupervisionado.

ABSTRACT

SOUZA, M. C. **Fake news detection using few positive labels**. 2023. 157 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2023.

The advancement of technology has allowed the spread of fake news on a large scale. Fake news is created to manipulate users' perspectives, attract their ideological stance and instigate them to share information, and it is becoming more susceptible to misleading the target audience. Machine Learning methods have been used as a promising strategy to detect fake content, whose problem is usually modeled with binary or multiclass learning algorithms. However, one of the challenges is to define a representative and concise set of news to train the algorithms due to (i) the naturally latent imbalance between the amount of true and false news available; (ii) the dynamism in which fake news evolves, increasingly convincing and similar to accurate news; (iii) in addition to the difficulty in labeling a large amount of news, it is necessary to check each fact reported in the publication's content. Considering the difficulty in labeling fake news (examples of interest or positive) while the characterization of true news is broad (not interesting or negative examples), this project proposes an approach for detecting fake news that characterizes the problem through One-Class Learning (OCL). OCL algorithms learn classification models considering only information from the class of interest. In addition, Positive Learning and Unlabeled (PUL) methods use information from unlabeled data to increase classification performance. This work proposes approaches based on the Positive and Unlabeled Learning by Label Propagation (PU-LP) algorithm, a PUL algorithm based on similarity networks. PU-LP identifies potential examples of the positive and negative class, and subsequently, a semi-supervised algorithm calculates the remaining unlabeled nodes. Different network configurations and semi-supervised classification algorithms are evaluated in six news bases that present different scenarios regarding language, topics, type of collection, and balance between classes. Experiments indicate that networks composed of news and representative terms can improve the performance of the approach, which is capable of identifying fake news with up to a 94% F_1 score using 10% of labeled positive data.

Keywords: Fake news detection, one class learning, positive and unlabeled learning, heterogeneous networks, semi-supervised learning.

LISTA DE ILUSTRAÇÕES

LISTA DE ALGORITMOS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

k-NN	*K-Nearest Neighbors*
AM	Aprendizado de Máquina
CBOW	*Continuous Bag-of-words*
CNN	*Convolutional Neural Network*
DT	*Decision Tree*
FN	*Fake News*
FNDOCC	*Fake News Detection through One Class Classification*
GFHF	*Gaussian Fields and Harmonic Functions*
GRU	*Gated Recurrent Unit*
HTML	*HyperText Markup Language*
LIWC	*Linguistic Inquiry and Word Count*
LLGC	*Learning With Local and Global Consistency*
LOF	*Local Outlier Factor*
LPHN	*Label Propagation through Heterogeneous Network*
LR	*Logistic Regression*
LSTM	*Long Short Term Memory*
MT	Mineração de Textos
NB	*Naïve Bayes*
OCC	*One Class Classification*
PLN	Processamento de Linguagem Natural
PU	*Positive and Unlabeled*
PU-LP	*Positive-Unlabeled learning by Label Propagation*
PUL	*Positive-Unlabeled Learning*
RF	*Random Forest*
RNN	*Recorrent Neural Network*
SVM	*Support Vector Machines*
tf	*term frequency*
tf-idf	*term frequency - inverse document frequency*
URL	*Uniform Resource Locator*
XGB	*Extreme Gradient Boosting*

SUMÁRIO

CAPÍTULO

1

INTRODUÇÃO

Fake news são notícias falsas escritas com a intenção de enganar, difamar, causar pânico, manipular a perspectiva de usuários, confirmar suas crenças pessoais e instigá-los a compartilhar a informação (ROHERA *et al.*, 2022; SIVEK, 2018a). A rápida disseminação de notícias falsas tem sido facilitada pela popularização da internet e redes sociais, e vem causando sérios prejuízos pessoais, sociais e econômicos nos últimos anos (SHAHID *et al.*, 2022; AÏMEUR; AMRI; BRASSARD, 2023).

Abordagens têm sido propostas na literatura para redução do impacto das fake news, com o intuito de classificar informações textuais como reais ou falsas, ou considerando diferentes níveis de falsidade. A forma mais comum de atacar este problema é por meio de algoritmos de aprendizado supervisionados binários ou multiclasse. Neste modelo de aprendizado, algoritmos necessitam de bases de dados extensivamente rotuladas para a extração de padrões discriminativos das notícias. No entanto, rotular um alto volume de notícias é uma tarefa complexa, que além de consumir tempo, pode ser enviesada pelo rotulador (MISHRA; SHUKLA; AGARWAL, 2022). Além disso, pesquisas mostram que notícias falsas estão disponíveis em diferentes tipos e formas, assumindo características distintas para se adequarem ao público de determinada mídia social. Em geral, quando um algoritmo é treinado com notícias de um certo tópico, como política, ele não discrimina bem notícias de outros tópicos, como religião. O mesmo vale para diferentes línguas, demandando bases de dados específicas para lidar com tarefas específicas, o que torna difícil a definição de uma única solução global para lidar com este desafio (AÏMEUR; AMRI; BRASSARD, 2023; SHAHID *et al.*, 2022). Diante destas limitações e do fato de que a qualidade das bases de dados disponibilizadas atualmente ainda limitam o desempenho de soluções propostas, são necessários algoritmos que obtenham melhores respostas na classificação considerando poucos dados rotulados.

Considerando os desafios de rotular grandes conjuntos de treinamento para detecção de notícias falsas, o Aprendizado de uma Única Classe (PERERA; OZA; PATEL, 2021; WANG;

BAH; HAMMAD, 2019; KHAN; MADDEN, 2014; TAX, 2001), ou *One-Class Learning* (OCL), pode ser uma abordagem promissora (FAUSTINI; COVÕES, 2019). Além de utilizar apenas exemplos de interesse como entrada, contribuindo em cenários nos quais é complexo rotular exemplos das demais categorias (BELLINGER *et al.*, 2017), OCL deriva uma abordagem semis-supervisionada denominada Aprendizado Positivo e Não Rotulado (BEKKER; DAVIS, 2020), ou *Positive and Unlabeled Learning* (PUL), que aprende um modelo a partir de poucos dados rotulados, fazendo uso de dados não rotulados para aumentar o desempenho de classificação.

Algoritmos PUL geralmente realizam a fase de aprendizado do modelo seguindo duas etapas: primeiramente, gera-se um conjunto de documentos de classes de não interesse, também chamado de classe negativa, por meio da seleção de exemplos não rotulados que apresentem características divergentes do conjunto de treinamento. O conjunto inicialmente rotulado também pode ser aumentado de forma automática a partir da seleção de exemplos com características próximas a ele. Uma vez que há documentos da classe de interesse, negativos e não rotulados, um algoritmo de aprendizado binário semissupervisionado classifica os demais exemplos na segunda etapa (BEKKER; DAVIS, 2020).

Pressupondo a utilização de dados não rotulados no processo de aprendizado, outro fator que impacta diretamente nos resultados é a forma de representação dos textos (ENGELEN; HOOS, 2020). Neste cenário, representações baseadas em redes podem ser úteis por serem capazes de unir vários tipos de dados (nós) e relações (arestas), com rica semântica (YANG *et al.*, 2020; SHI; PHILIP, 2017). Redes permitem a extração de padrões de classes que dificilmente são capturados por modelos espaços vetoriais (BREVE *et al.*, 2012), além de demonstrarem efetividade no aprendizado semi-supervisionado (ROSSI; LOPES; REZENDE, 2016).

Com base nas informações apresentadas, e no fato de que OCL e PUL foram pouco exploradas na literatura para classificação de notícias (FAUSTINI; COVÕES, 2019; LIU; WU, 2020; SOUZA *et al.*, 2022; SOUZA *et al.*, 2021), neste trabalho é proposta uma abordagem fundamentada no algoritmo Aprendizado Positivo e Não Rotulado por Propagação de Rótulos, ou *Positive and Unlabeled Learning by Label Propagation* (PU-LP) (MA; ZHANG, 2017), um algoritmo de aprendizado PUL transdutivo baseado em redes homogêneas. PU-LP foi desenvolvido e avaliado em bases de dados de atributos numéricos, obtendo bom desempenho com apenas 10% de dados rotulados. O algoritmo identifica potenciais documentos de interesse e negativos considerando uma medida de similaridade baseada em caminhos (KATZ, 1953), chamada índice de Katz. Então, um algoritmo transdutivo de propagação é usado para rotular os demais objetos da rede. Neste trabalho, PU-LP é adaptado para classificação de textos e aplicado ao cenário de detecção de notícias falsas, considerando a classe “fake” como interesse.

Com o intuito de reduzir esforços de rotulação no cenário de Aprendizado Positivo e Não Rotulado e explorar representações baseadas em redes, são analisadas diferentes formas de estruturar notícias e seus elementos, visando enriquecer o modelo para aumentar o desempenho de classificação (DEEPAK *et al.*, 2021; YANG *et al.*, 2020; SHI; PHILIP, 2017; ROSSI; REZENDE;

LOPES, 2015). Dentre as características que podem ser incorporadas à rede, estão (i) termos, considerados como característica genérica, que podem ser extraídos de qualquer coleção de notícias. A inclusão de termos em redes de documentos já demonstrou benefícios na literatura, aumentando o desempenho de classificação no cenário de aprendizado semissupervisionado. Além disso, a análise de termos discriminativos é frequentemente empregada para detecção de conteúdo falso (YAN *et al.*, 2020; ROSSI; REZENDE; LOPES, 2015; AGGARWAL; LI, 2011; HASSAN *et al.*, 2020; AHMED; TRAORE; SAAD, 2017; PÉREZ-ROSAS *et al.*, 2017; RUBIN *et al.*, 2016; MIHALCEA; STRAPPARAVA; PULMAN, 2010); e (ii) características linguísticas, como emotividade, número médio de palavras por sentença e pausalidade, que já se mostraram relevantes (SANTOS; PARDO, 2020) na classificação de notícias.

Para a classificação de nós não rotulados na segunda etapa de PU-LP, são explorados algoritmos semissupervisionados clássicos e estado da arte da literatura (MATTOS; MARCACINI, 2021; ROSSI, 2016; JI *et al.*, 2010), baseados em regularização e propagação de rótulos. A abordagem é avaliada considerando algoritmos de aprendizado de uma única classe de diferentes paradigmas, como métodos estatísticos, baseados em distância e densidade, agrupamento, aprendizado profundo, *Graph Neural Networks* e PUL que demonstraram efetividade na classificação de dados textuais (GÔLO; MARCACINI; ROSSI, 2020; WANG *et al.*, 2021). Os experimentos são realizados em bases de dados de diferentes cenários, abordando um mesmo assunto ou vários assuntos, contendo diferentes níveis de balanceamento de notícias reais e falsas, línguas portuguesa e inglesa, além de distintas formas de coleta.

Com a abordagem de detecção de notícias falsas proposta nesta tese, espera-se que: (i) características relevantes e discriminativas, que possam ser extraídas do próprio texto da publicação por meio de técnicas de processamento de linguagem natural, sejam representadas em uma rede heterogênea; (ii) a abordagem de aprendizado de uma única classe baseada no algoritmo PU-LP seja adequada para manipular a rede heterogênea, discriminando notícias reais e falsas com boa acurácia; e por fim, (iii) esforços de rotulação sejam reduzidos, tanto pela baixa quantidade de objetos de interesse rotulados quanto pela não caracterização de notícias verdadeiras no conjunto de treinamento.

1.1 Motivação e Lacunas

Identificar conteúdo falso é uma tarefa difícil até mesmo para o ser humano, já que notícias falsas geralmente vão de encontro aos seus medos, ansiedades, curiosidades, e exploram suas capacidades cognitivas, emoções e vieses ideológicos (SIVEK, 2018b).

Apesar da existência de sites de checagem, eles são insuficientes para combater o volume e alcance de desinformação disseminada. Nas eleições presidenciais brasileiras de 2022, por exemplo, houve uma média de 311,5 mil mensagens falsas compartilhadas diariamente. Isso ocorre já que fake news estão cada vez mais sofisticadas, adotando características complexas que

diferem entre diferentes tipos de redes sociais [1].

O problema é ainda mais desafiador para modelos computacionais, que precisam projetar estratégias que atenuem o problema de notícias falsas sem restringir o acesso rápido a informação de qualidade (ZHOU *et al.*, 2019; SHARMA *et al.*, 2019).

Em um cenário no qual técnicas para criação de conteúdo falso se tornam cada vez mais sofisticadas, sistemas de detecção de notícias falsas apresentam um grande risco tanto em classificar erroneamente informações verdadeiras, quanto em não identificar a presença de conteúdo enganoso de notícias potencialmente virais. Embora na literatura sejam encontradas técnicas aplicáveis a detecção de notícias falsas e sua mitigação, a área ainda possui muitas limitações e lacunas que podem ser exploradas (BONDIELLI; MARCELLONI, 2019).

Trabalhos atualmente propostos para detecção de notícias falsas cobrem três grandes áreas: (i) identificação de notícias falsas por meio do conteúdo da publicação; (ii) métodos que classificam notícias com base em informações de contexto; e (iii) soluções baseadas em intervenção que restrinjam a disseminação da informação. Entre os trabalhos baseados em conteúdo, há métodos que analisam características linguísticas, que sejam informativas para diferenciar notícias verdadeiras de falsas. No entanto, indícios de notícias falsas diferem entre tópicos, linguagens e domínios, demandando adaptações dinâmicas em estratégias anteriormente propostas (CAPUANO *et al.*, 2023; SHARMA *et al.*, 2019; BONDIELLI; MARCELLONI, 2019; ZHOU *et al.*, 2019).

Além da necessidade de características generalizáveis de notícias falsas, que possam ser extraídas de qualquer base de dados, a rapidez na qual notícias falsas evoluem demanda o estudo de modelos de classificação que apresentem bom desempenho diante de poucos dados rotulados. Estudos recentes também apontam a necessidade de soluções semissupervisionadas, que visem diminuir o esforço manual de rotulação de notícias, dado que um dos principais desafios de sistemas supervisionados é definir um conjunto de treinamento representativo e conciso (ZHANG; GHORBANI, 2020; SILVA; FONTES; JÚNIOR, 2020; BONDIELLI; MARCELLONI, 2019; MEEL; VISHWAKARMA, 2019; SHARMA *et al.*, 2019).

1.2 Questões de Pesquisa e Objetivos

Esta tese de doutorado propõe a detecção de notícias falsas por meio de abordagens baseadas em aprendizado de uma única classe utilizando Mineração de Textos (AGGARWAL, 2018; REZENDE *et al.*, 2003). A abordagem proposta é fundamentada no algoritmo PU-LP, que por ser totalmente baseado em redes permite a inclusão de novas características na estrutura. PU-LP explora o conjunto inicialmente rotulado de notícias falsas, dados não rotulados e uma medida de similaridade para criação de conjuntos de notícias potencialmente reais e falsas.

[1] http://www.netlab.eco.ufrj.br/blog/acompanhamento-multiplataforma-da-desinformacao-durante-as-eleicoes-2022

Posteriormente, um algoritmo semissupervisionado realiza a classificação dos demais nós não rotulados considerando a informação de rótulo de nós vizinhos.

A primeira abordagem proposta consiste na classificação de notícias utilizando o algoritmo PU-LP com redes homogêneas e heterogêneas. As redes homogêneas são compostas apenas por notícias, enquanto as redes heterogêneas são compostas por notícias e características genéricas, que podem ser extraídas de qualquer base de dados textual existente na literatura. Diante deste cenário, as seguintes questões de pesquisa devem ser respondidas:

Q1 *"Qual grupo de algoritmos (OCL x PUL) se destaca na detecção de notícias falsas?"*

Q2 *"Como o modelo de representação utilizado para transformar notícias em dados estruturados pode influenciar no desempenho da abordagem de detecção de notícias falsas?"*

Q3 *"Dentre as características textuais analisadas para inclusão na rede heterogênea, quais contribuem no desempenho da abordagem proposta de detecção de notícias falsas?"*

Q4 *"O desempenho da abordagem proposta, baseada em aprendizado de uma única classe semissupervisionado, supera algoritmos OCL e PUL da literatura? E algoritmos semissupervisionados binários?"*

Q5 *"O aumento do número de notícias falsas inicialmente rotuladas aumenta significativamente o desempenho de classificação dos algoritmos PUL?"*

Diante das questões de pesquisa, foram definidos os seguintes objetivos para o desenvolvimento deste trabalho.

1. Mapear algoritmos de classificação de uma única classe e aprendizado positivo e não rotulado disponíveis na literatura, avaliando suas lacunas. Analisar o desempenho destes algoritmos (medida F_1) em diferentes bases de notícias. Esse objetivo está relacionado às questões de pesquisa **Q1**.

2. Mapear bases de dados da literatura apresentando diferentes cenários quanto a tipo de linguagem, balanceamento, tópicos e formas de coleta. Mapear e propor modelos de representação capazes de modelar o conteúdo completo da notícia e avaliar o desempenho destes modelos em algoritmos OCL e PUL. Esse objetivo está relacionado à questão de pesquisa **Q2**.

3. Mapear características discriminativas de notícias reais e falsas propostas em trabalhos da literatura que classificam notícias considerando o conteúdo textual. A partir das características relevantes mapeadas, analisar formas de incluí-las na rede heterogênea da abordagem proposta e analisar seu impacto na detecção de notícias falsas. Esse objetivo está relacionado à questão de pesquisa **Q3**.

4. Propor e desenvolver uma abordagem, que por meio de aprendizado positivo e não rotulado seja capaz de classificar notícias dispostas em uma rede heterogênea, que reúna

características que forneçam indícios de conteúdos verdadeiros ou falsos. Comparar o desempenho da abordagem a algoritmos OCL e PUL da literatura. Comparar os resultados atingidos com abordagens semissupervisionadas binárias. Analisar como o desempenho do algoritmo é afetado pelo número inicial de notícias falsas rotuladas. Esse objetivo está relacionado às questões de pesquisa **Q1**, **Q2**, **Q3**, **Q4** e **Q5**.

Considerando os resultados atingidos a partir dos objetivos previamente definidos, notou-se que a adição de termos relevantes à rede de notícias proporcionou aumento de F_1 da classe falsa e macro. Logo, foi proposta uma nova abordagem a fim de tornar os resultados mais precisos. A evolução da abordagem consiste (i) na seleção de uma ferramenta de extração de palavras-chave sólida na literatura; e (ii) da utilização de um algoritmo semissupervisionado estado da arte na etapa final do algoritmo PU-LP, baseado em redes neurais e mecanismos de atenção. Mecanismos de atenção se mostram relevantes neste contexto, pois permitem que o algoritmo aprenda informações sobre quais nós vizinhos são mais relevantes no processo de classificação de uma notícia. As questões de pesquisa respondidas foram as seguintes:

Q6 *"A inclusão de termos representativos na rede heterogênea, extraídos com ferramentas de extração de palavras-chave, pode aumentar o desempenho de classificação? Quais tipos de termos são relevantes na discriminação de conteúdo real e falso?"*

Q7 *"Estratégias de classificação semissupervisionadas baseadas em redes de atenção podem aumentar o desempenho de PU-LP? A integração de atenção no algoritmo PU-LP pode superar algoritmos estados da arte baseados em uma única classe na detecção de notícias falsas?"*

Q8 *"Quais as vantagens e limitações da abordagem proposta baseada em PU-LP na classificação de notícias?"*

Nesta etapa, foram definidos os seguintes objetivos.

1. Mapear e selecionar ferramentas de extração de palavras-chave existentes na literatura. Após a definição da ferramenta e do processo de classificação, analisar quais tipos de termos estão mais relacionadas às classes reais e falsas. Este objetivo está relacionado à questão de pesquisa **Q6**.

2. Mapear e selecionar algoritmos estado da arte para classificação de nós não rotulados baseados em uma única classe, redes neurais e grafos. Executar algoritmos e analisar resultados. Este objetivo está relacionado à questão de pesquisa **Q7**.

3. Analisar de forma aprofundada os textos de notícias para discutir sobre as vantagens e limitações da abordagem PU-LP. Este objetivo está relacionado à questão de pesquisa **Q8**.

1.3 Principais Resultados Obtidos

Os principais resultados obtidos com o desenvolvimento deste projeto são:

R1 Revisão da literatura sobre Detecção de Notícias Falsas. Foram analisados trabalhos abrangendo a literatura de Mineração de Textos que consideram a tarefa de classificação de notícias. Observaram-se lacunas sobre o desenvolvimento de abordagens de aprendizado de uma única classe e aprendizado positivo e não rotulado, na qual grande parte dos trabalhos ataca o problema com algoritmos supervisionados binários, que demandam grande quantidade de dados rotulados e bases balanceadas de notícias. Esse resultado viabilizou a resposta da questão **Q2**.

R2 Revisão da literatura sobre algoritmos OCL e PUL que pudessem ser aplicados ao domínio de textos. Seleção do algoritmo PU-LP para compor a abordagem proposta. Seleção dos algoritmos utilizados para avaliação da abordagem. Esse resultado viabilizou a resposta da questão **Q1**.

R3 Mapeamento de bases de notícias da literatura para avaliação da abordagem. Durante o desenvolvimento deste protejo, foram coletadas diferentes bases de dados que representassem cenários distintos: língua portuguesa e inglesa, bases balanceadas e extremo desbalanceamento, e bases contendo apenas um ou diversos assuntos. Esse resultado viabilizou a resposta de todas das questões **Q1**, **Q2**, **Q3**, **Q4 Q5**, **Q6**, **Q7** e **Q8**.

R3 Revisão da literatura sobre modelos que pudessem representar notícias de forma estruturada considerando o texto completo da publicação. Foram considerados os modelos *Bag-of-Words* e *Doc2Vec*. Esse resultado viabilizou a resposta da questão **Q2**.

R4 Seleção de características relevantes para compor a rede heterogênea do algoritmo PU-LP. Foi analisada a inclusão de termos, extraídos com *Bag-of-Words* e tf-idf, bem como emotividade, número médio de palavras por sentença e pausalidade. Para propagação de rótulos, foram selecionados os algoritmos LPHN e GNetMine. Foram avaliados os resultados atingidos no contexto de classificação de notícias. Como principal resultado, notou-se que a inclusão de termos contribuiu para o desempenho dos algoritmos de propagação de rótulos, em especial quando notícias falsas estavam dispersas no espaço de características, considerando informações de tópicos e veracidade. Além disso, notou-se que utilizar apenas 10% de notícias forneceu bons resultados de classificação em determinadas bases de dados. Esse resultado viabilizou a resposta das questões **Q3**, **Q4** e **Q5**.

R5 Considerando os resultados atingidos com a inclusão de termos relevantes, foi realizada uma revisão de ferramentas de extração de palavras-chave. Dentre elas, destacou-se a ferramenta Yake!, superando outras em tarefas avaliadas na literatura. Novos experimentos foram realizados, e os resultados se mostraram melhores em relação à estratégia anteriormente definida. Esse resultado viabilizou a resposta da questão **Q6**.

R6 Avaliação de algoritmos estado da arte que pudessem compor a etapa de classificação

de nós não rotulados de PU-LP, após a inferência de notícias de interesse e não interesse confiáveis. Foi selecionado o algoritmo GNEE desenvolvido para redes heterogêneas. GNEE é semissupervisionado, baseado em redes de atenção e apresentou resultados relevantes na literatura. GNEE contribuiu para a melhoria dos resultados, em especial na utilização de 10% de notícias falsas inicialmente rotuladas. Esse resultado viabilizou a resposta da questão **Q7**.

R7 Uma análise sobre as limitações da abordagem proposta apontou a presença de termos e expressões presentes nas bases de dados que poderiam contribuir no enviesamento do processo de classificação. A presença destes termos depende das fontes que foram retiradas as notícias. Esta análise mostra a importância de ir a fundo sobre o conteúdo das bases de dados para avaliar o desempenho de algoritmos de classificação. Além disso, evidencia a importância de estudos de modelos que desempenhem bem diante de poucos dados rotulados, considerando a dificuldade de se obter dados com número significativo de notícias rotuladas e com curadoria adequada. Esse resultado viabilizou a resposta da questão **Q8**.

R8 Disponibilização dos recursos e algoritmos. Considerando a continuidade desta pesquisa, a reprodução dos resultados e aplicação dos métodos desenvolvidos, os resultados deste trabalho foram disponibilizados na página do *Github*: <https://github.com/marianacaravanti>. Nesta página encontram-se todas as bases de dados e algoritmos desenvolvidos e utilizados ao longo da pesquisa, bem como o resultado de avaliações experimentais atingidos nesta tese.

1.4 Organização do Texto

Esta tese de doutorado é organizada como se segue.

- Capítulo 2 - *Mineração de Textos para Detecção de Notícias Falsas*. Neste capítulo são apresentados conceitos essenciais para o entendimento deste projeto de doutorado. Primeiramente, é apresentada uma visão geral sobre Mineração de Textos. A etapa de pré-processamento é detalhada, considerando técnicas amplamente utilizadas e aplicáveis ao problema de detecção de notícias falsas. O capítulo aborda também a definição do problema de notícias falsas, as principais coleções de notícias utilizadas na literatura para avaliar modelos de classificação, além de algoritmos e abordagens de extração de padrões propostos nos últimos anos para discriminação de conteúdo.

- Capítulo 3 - *Aprendizado de Uma Única Classe*. Neste capítulo são discutidos algoritmos de aprendizado de uma única classe, divididos em modelos estatísticos, baseados em distância e densidade, agrupamento, aprendizado profundo, *graph neural networks* e aprendizado positivo e não rotulado. O capítulo aborda também algoritmos de propagação de rótulos, geralmente utilizados na segunda etapa de abordagens PUL para classificação de nós não rotulados. Além disso, são listados algoritmos recentemente propostos de OCL aplicados a textos, bem como suas lacunas no contexto de detecção de notícias falsas.

- Capítulo 4 - *Detecção de Notícias Falsas a partir de Poucos rótulos de interesse*. Neste capítulo é apresentado a proposta deste doutorado, que consiste em detectar notícias falsas por meio de aprendizado semissupervisionado baseado em uma única classe e modelo de representação de redes heterogêneas. São descritos os experimentos realizados, os resultados atingidos, bem como evoluções implementadas na abordagem.

- Capítulo 5 - *Conclusões*. Neste capítulo é apresentada a conclusão, contribuições científicas da autora, publicações, além da discussão de trabalhos futuros que podem ser explorados por meio dos resultados atingidos nesta tese.

CAPÍTULO

2

MINERAÇÃO DE TEXTOS PARA DETECÇÃO DE NOTÍCIAS FALSAS

Ao longo dos anos, um campo de pesquisa que vem crescendo de forma rápida é a descoberta de conhecimento a partir de bases de dados textuais. Tanto na *web* quanto em dispositivos móveis, grande parte dos dados disponíveis possuem formato textual, que se explorados podem revelar tendências e padrões úteis em processos de tomadas de decisão (AGGARWAL, 2018; REZENDE *et al.*, 2003).

A extração de informação útil de coleções de textos pode ser definida como Mineração de Textos (MT) (REZENDE *et al.*, 2003). Uma vez definido o problema, duas dimensões principais devem ser consideradas em tarefas de Mineração de Textos: (i) o tipo de representação, que indica como os textos serão representados (caracteres, frases, *Bag-of-Words*, *part of speech*, estruturas de redes, etc); e (ii) a extração de padrões, na qual são definidos os algoritmos de aprendizado que manipularão as representações (GROBELNIK, 2011).

Diante deste cenário, e conforme apresentado no Capítulo 1, o interesse deste trabalho está em utilizar recursos da Mineração de Textos para detectar notícias falsas por meio de algoritmos de aprendizado semissupervisionados baseados em uma única classe e modelos de representação baseados em redes heterogêneas. Neste capítulo são abordados o problema de detecção de notícias falsas, bem como os principais meios utilizados na literatura para representação de documentos e extração de padrões. O capítulo é organizado da seguinte forma: na Seção 2.1 é apresentada uma visão geral sobre Mineração de Textos, seguido de uma revisão da literatura sobre modelos de representação na Seção 2.2. Na Seção 2.3 são fornecidos os principais fundamentos sobre detecção de notícias falsas, bem como bases de dados normalmente utilizados nesta tarefa. Na Seção 2.4 são discutidos algoritmos de detecção de padrões empregados na literatura para discriminação de conteúdo falso. Na Seção 2.5 são apresentadas as considerações finais e introduzida a proposta desta tese considerando discussões sobre as lacunas encontradas.

2.1 Mineração de Textos

Considerada uma subárea de Mineração de Dados, a Mineração de Textos (MT) trabalha sobre dados inerentemente não estruturados, transformando textos em língua natural em formatos que sejam manipuláveis por algoritmos de aprendizado. A partir destes dados, informações valiosas podem ser extraídas, como tendências e padrões que podem ser úteis no processo de tomada de decisão (AGGARWAL, 2018; WEISS *et al.*, 2005).

Técnicas de MT podem ser aplicadas à diferentes tipos de dados textuais. Exemplos são (AGGARWAL, 2018): alavancar pesquisas médicas e biológicas por meio de bibliotecas digitais; analisar dados de redes sociais, que contam com uma grande variedade de usuários, trazendo a necessidade de técnicas especiais que lidem com a presença de gírias e acrônimos; o desenvolvimento de sistemas de recomendação em portais de notícias; ou até mesmo classificar notícias como verídicas ou não com base no conteúdo da informação, sendo esse último o foco deste projeto. De maneira geral, a MT pode ser tratada como um processo composto de cinco etapas principais (REZENDE *et al.*, 2003): (i) identificação do problema, (ii) pré-processamento, (iii) extração de padrões, (iv) pós-processamento e (v) utilização do conhecimento, ilustradas na Figura 1.

Figura 1 – Processo de Mineração de Textos.

Fonte: REZENDE *et al.* 2003

A primeira fase é responsável pela identificação do domínio da aplicação. Durante a **identificação do problema**, são estabelecidos objetivos e metas a serem alcançadas no processo de MT. Para o sucesso do processo, é importante a participação e o conhecimento de especialistas que façam parte do domínio de aplicação. Antes de iniciar qualquer tarefa, é necessário um estudo que permita o conhecimento inicial do domínio. Além disso, devem ser identificadas quais coleções de textos serão utilizadas, bem como formas de se explorar possíveis resultados obtidos (REZENDE *et al.*, 2003).

Após definidas as metas que devem ser alcançadas, o próximo passo é **pré-processar** a coleção de textos para que elas sejam adequadamente utilizadas por algoritmos de extração de padrões. Esta etapa pode ser feita de algumas formas, como (AGGARWAL, 2018; AGGARWAL *et al.*, 2018; ROSSI, 2016): por meio de técnicas que se baseiam na funcionalidade dos termos ou na contabilização da ocorrência de termos nos documentos; ou utilizando redes complexas, que permitem a modelagem de diferentes entidades do texto por meio de nós e suas relações (arestas). Além disso, é importante que o modo de pré-processamento escolhido preserve os padrões existentes na base textual. Na Seção 2.2 são discutidas de forma mais ampla as formas de representação geralmente utilizadas no domínio de detecção de notícias falsas.

Após o tratamento da coleção de textos, o próximo passo é a **extração de padrões**. A partir da representação dos textos obtida, métodos para a extração de padrões são aplicados aos dados, como algoritmos de Aprendizado de Máquina (AGGARWAL, 2018; REZENDE *et al.*, 2003). As principais atividades de extração de padrões podem ser divididas em atividades preditivas, que produzem um modelo descrito pelo conjunto de dados para predizer o valor de uma ou mais variáveis de interesse; e atividades descritivas, que utilizam algoritmos de extração de padrões em dados não-rotulados, buscando por regras de associação, agrupamentos ou sumarização de documentos (NOGUEIRA *et al.*, 2008).

Os padrões obtidos na etapa de extração de padrões podem ser interpretados pela etapa de **pós-processamento**, que consiste na validação das descobertas efetuadas, bem como na visualização dos resultados encontrados. No pós-processamento, métricas de avaliação de resultados, ferramentas de visualização e o conhecimento de especialistas ajudam a consolidar os resultados (REZENDE *et al.*, 2003). Em aplicações que envolvam tarefas preditivas, os modelos podem ser avaliados com medidas relacionadas à precisão na predição de dados, como taxa de erro, precisão, revocação e cobertura (AGGARWAL, 2018; NOGUEIRA *et al.*, 2008).

Após a etapa de pós-processamento, o conhecimento pode ser aproveitado pelo usuário, sendo válido e útil se as etapas anteriores forem devidamente aplicadas. Esta etapa é conhecida como **utilização do conhecimento**. O conhecimento adquirido pode ser utilizado no processo de tomada de decisão, conforme os objetivos definidos no início do processo. Caso o conhecimento extraído não tenha cumprido o objetivo proposto, outro ciclo do processo de MT deve ser executado, de forma que mudanças podem ser realizadas desde a etapa de pré-processamento. Além disso, se novos objetivos forem estabelecidos, o processo deve retornar para a etapa de identificação do problema.

2.2 Pré-processamento de Textos

Após a coleta de dados, a informação textual pode ser representada computacionalmente por uma sequência de caracteres. Para que estes apresentem valor semântico, é necessário que eles sejam convertidos em *tokens*, que são sequências contínuas de caracteres que possuem

significado. *Tokens* são diferentes de *termos*, já que os termos de uma coleção se referem ao conjunto de palavras pertencentes à coleção, sem repetição (AGGARWAL, 2018).

Na tokenização é comum a eliminação de sinais de pontuação, caracteres alfanuméricos, padronização de caixas e a utilização de espaços em branco como separadores no texto. Entretanto, tais regras podem ser inadequadas em tarefas específicas, como, por exemplo, ao se trabalhar com coleções de textos curtos, ou quando a informação eliminada poderia ser úteis na discriminação de conteúdo (HORNE; ADALI, 2017; PÉREZ-ROSAS *et al.*, 2017; CASTILLO; MENDOZA; POBLETE, 2011). Além disso, ao separar palavras compostas como "São Paulo", perde-se o significado semântico. Para evitar esta perda, algumas estratégias podem ser adotadas, como: utilizar dicionários de palavras semanticamente coocorrentes; armazenar palavras utilizadas comumente juntas e extraí-las das sequências de caracteres; e analisar *n*-gramas mais frequentes da coleção para a identificação de termos compostos.

Em coleções de textos, palavras comuns na linguagem que não carregam informação discriminativa são chamadas *stopwords*. Exemplos de *stopwords* são: artigos, preposições, conjunções e pronomes, ou mesmo tokens que ocorrem com muita frequência em um domínio (*stopwords* de domínio), podendo ser removidas da coleção de textos dependendo do objetivo da aplicação. Se os critérios de remoção de *stopwords* forem muito rígidos pode haver perda de informação importante. Neste caso, uma alternativa é a redução do peso de palavras frequentes ao invés de removê-las da coleção (AGGARWAL, 2018).

Uma vez que palavras são convertidas em *tokens*, eles podem ser transformados em *termos* associados com suas frequências dentro da coleção. Considerando que documentos podem conter palavras de diferentes tempos verbais, assim como substantivos podem ter diferentes formas de flexão de número, em algumas aplicações é interessante consolidar estas variações em uma só palavra. Este processo tem como intuito a simplificação das palavras, fazendo uso de técnicas como substantivação, radicalização e lematização que reduzem termos aos seus substantivos, radicais e lemas, respectivamente.

Após a extração dos termos relevantes da coleção, algumas técnicas podem ser utilizadas para representar termos e documentos como vetores de características, que posteriormente servirão como conjunto de entrada a algoritmos de extração de padrões. A seguir são listadas as principais técnicas disponíveis na literatura, amplamente utilizadas para detecção de notícias falsas, e detalhadas aquelas que foram essenciais para o desenvolvimento desta tese.

2.2.1 Bag-of-Words

Um modelo de representação de documentos clássico é o modelo *Bag-of-Words* (BoW), no qual cada documento é representado por um vetor de termos da coleção (MANNING; RAGHAVAN; SCHÜTZE, 2008; FELDMAN; SANGER *et al.*, 2007).

Na Tabela 1 é apresentada a ilustração de uma representação *Bag-of-Words*. Seja

$\mathscr{D} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_m\}$ um conjunto de M documentos e $\mathscr{T} = \{t_1, t_2, \ldots, t_n\}$ um conjunto de N termos. Logo, cada um dos M vetores possui N dimensões. A união dos vetores de documentos de uma coleção é então representada pela matriz *Bag-of-Words*. O valor de uma célula w_{d_i,t_j} representa o peso da dimensão t_j em um documento d_i. A última coluna da matriz representa a classe do documento c_{d_i} (rótulo), que nem sempre é conhecida. Os valores w_{d_i,t_j} associados às dimensões são sempre não negativos. Para dimensões referentes aos termos da coleção, o valor da célula pode ser representado como (i) a frequência do termo, ou *term frequency* (tf); (ii) a frequência normalizada do termo, na qual a medida mais utilizada é a frequência do termo ponderada pelo inverso da frequência do documento, ou *term frequency - inverse document frequency* (tf-idf); ou ainda (iii) um indicador binário de valor 1. Caso o termo não esteja presente no documento, o valor da célula é preenchido com zero.

Tabela 1 – Ilustração de uma representação *Bag-of-Words* com m documentos e n dimensões.

	$\mathbf{t_1}$	$\mathbf{t_2}$	...	$\mathbf{t_n}$	**Classe**
$\mathbf{d_1}$	w_{d_1,t_1}	w_{d_1,t_2}	...	w_{d_1,t_n}	c_{d_1}
$\mathbf{d_2}$	w_{d_2,t_1}	w_{d_2,t_2}	...	w_{d_2,t_n}	c_{d_2}
...	...	...	...	...	...
$\mathbf{d_m}$	w_{d_m,t_1}	w_{d_m,t_2}	...	w_{d_m,t_n}	c_{d_m}

Este tipo de representação é indicado quando os textos são longos, nos quais a frequência de termos específicos em cada classe sejam capazes de ajudar algoritmos de classificação no processo de discriminação. No entanto, um ponto negativo da representação *Bag-of-Words* é a perda da informação de ordem entre os termos da coleção. Quando a interpretação semântica de sentenças é importante, ou quando o tamanho dos segmentos de texto é pequeno, esta abordagem torna-se inapropriada (AGGARWAL, 2018; AGGARWAL *et al.*, 2018).

2.2.2 Modelos de Linguagem Independentes de Contexto

Em aplicações mais sofisticadas, uma abordagem comum é representar textos por meio de modelos de linguagem estatísticos, que atribuem uma probabilidade P a uma sequência de palavras $w_1, w_2, \ldots, w_m$. A forma mais simples de computar $\mathrm{P}(w_1, w_2, \ldots, w_m)$ é aplicar a regra da cadeia na sequência, conforme a Equação 2.1 (AGGARWAL, 2018):

$$\begin{aligned} P(w_1, w_2, \ldots, w_m) &= P(w_1) \cdot P(w_2|w_1) \cdot P(w_3|w_1, w_2) \cdot \ldots \cdot P(w_m|w_1, w_2, \ldots, w_{m-1}) \\ &= \prod_{i=1}^{m} P(w_i|w_1, \ldots, w_{i-1}) \end{aligned} \tag{2.1}$$

A distribuição de probabilidade P é calculada usando a contagem da ocorrência de termos da frase no conjunto de treinamento (Equação 2.2).

$$P(w_i|w_1,\ldots,w_{i-1}) = \frac{P(w_1,\ldots,w_i)}{P(w_1,\ldots,w_{i-1})} = \frac{\text{Contagem}(w_1,\ldots,w_i)}{\text{Contagem}(w_1,\ldots,w_{i-1})} \tag{2.2}$$

Um problema deste modelo é que a contagem do grupo de termos é difícil de ser estimada com precisão para grupos grandes, o que pode resultar em um numerador e denominador próximos de 0. Para minimizar este problema usa-se o pressuposto Markoviano, que considera apenas os últimos $n-1$ *tokens* para estimar a probabilidade condicional de um *token*, resultando em um modelo *n-gram*. Assim, o pressuposto Markoviano pode ser estimado conforme a Equação 2.3:

$$P(w_i|w_1,\ldots,w_{i-1}) \approx P(w_i|w_{i-n+1},\ldots,w_{i-1}) = \frac{\text{Contagem}(w_{i-n+1},\ldots,w_i)}{\text{Contagem}(w_{i-n+1},\ldots,w_{i-1})} \tag{2.3}$$

Embora grandes valores de n proporcionem melhores discriminações, a quantidade de dados disponíveis geralmente não é suficiente para que estimativas confiáveis sejam obtidas. Além disso, em modelos n-gramas, pequenas variações em uma sentença podem causar grandes efeitos na estimativa do modelo.

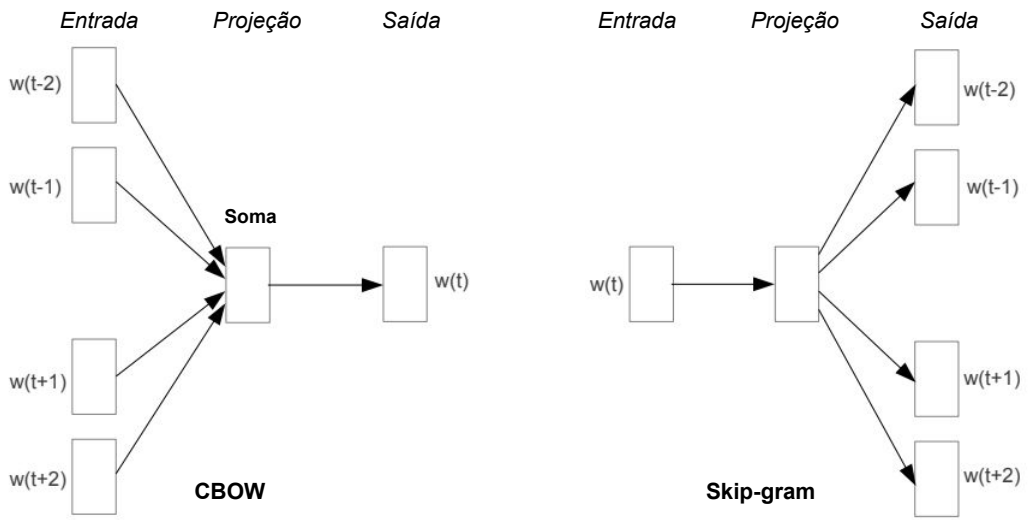
Para sanar tais lacunas, modelos de linguagem baseados em redes neurais apresentaram desempenhos superiores na estimativa de distribuição de probabilidade, representando palavras em vetores de baixa dimensão chamados *word embeddings* (BENGIO; DUCHARME; VINCENT, 2000). Esta técnica de processamento de linguagem natural mapeia palavras que ocorrem em contextos similares a representações próximas no espaço de características N-dimensional (NASEEM *et al.*, 2021).

Modelos de linguagem neurais obtém representações de palavras como produto do treinamento de modelos de linguagem. Assim como modelos de linguagem tradicionais, modelos neurais também utilizam $n-1$ palavras para predizer a n-ésima palavra da sentença. Em diferentes modelos propostos na literatura, *word embeddings* são pré-treinadas, em que o algoritmo deve predizer uma palavra com base em seu contexto. Isso permite que os modelos não percam a ordem das palavras e capturem seus significados.

Um modelo neural bem consolidado é o *Word2Vec* (MIKOLOV *et al.*, 2013), treinado em dois passos. No primeiro passo, vetores de palavras contínuas são aprendidos por meio de um modelo. No segundo passo, um modelo de linguagem neural n-gram é treinado a partir das representações distribuídas das palavras (BENGIO; DUCHARME; VINCENT, 2000).

Word2Vec realiza o treinamento de *embeddings* considerando duas variantes: *Continuous Bag-of-Words* e *Skip-gram*, e vem sendo usado como base para o desenvolvimento de novos modelos de linguagem. Uma janela deslizante de tamanho pré-definido é movido pelo corpus textual em ambos os modelos, e o treinamento é realizado com as palavras obtidas dentro desta janela (MIKOLOV; YIH; ZWEIG, 2013; NASEEM *et al.*, 2021; AGGARWAL, 2018). Tais modelos são detalhados a seguir, conforme a Figura 2.

Figura 2 – Variantes do modelo Word2Vec. Na esquerda é apresentada a arquitetura do modelo CBOW e na direita é apresentada a arquitetura do algoritmo Skip-gram. Figura adaptada de MIKOLOV; YIH; ZWEIG (2013).



O intuito do modelo *Continuous Bag-of-Words* (CBOW) é predizer a i-ésima palavra w_i de uma sentença utilizando uma janela de tamanho t ao redor da palavra. Assim, as palavras $w_{i-t}w_{i-t+1}\ldots w_{i-1}w_{i+1}\ldots w_{i+t-1}w_{i+t}$ são usadas para predizer a palavra alvo w_i.

A arquitetura CBOW é similar a uma rede neural *feed forward* (BENGIO; DUCHARME; VINCENT, 2000). A rede é composta por uma camada de entrada, uma camada de projeção, uma camada oculta e uma camada de saída (LE; MIKOLOV, 2014). Cada palavra é mapeada para um vetor único, representado como uma coluna na matriz $\mathbf{W}$. Cada coluna é indexada pela posição da palavra no vocabulário. A concatenação ou soma dos vetores é usado como características para a predição da próxima palavra na sentença. Ou seja, dado a sequência de palavras de treinamento $w_1, w_2, w_3, \ldots, w_t$, o objetivo do modelo é maximizar a média do log da probabilidade, conforme a Equação 2.4.

$$\frac{1}{T}\sum_{t=k}^{T-k}\log p(w_t|w_{t-k},\ldots,w_{t+k}) \tag{2.4}$$

A tarefa de predição é feita geralmente por um classificador multiclasse, como *Softmax* (Equação 2.5).

$$p(w_t|w_{t-k},\ldots,w_{t+k}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}} \tag{2.5}$$

Cada y_i é o log de probabilidade não normalizado de cada palavra de saída i, calculado conforme a Equação 2.6, na qual U e b são os parâmetros do classificador *Softmax* e h é construído pela concatenação ou média dos vetores de palavras extraídos de W.

$$y = b + Uh(w_{t-k}, \ldots, w_{t+k}; W) \tag{2.6}$$

A rede neural é geralmente treinada usando gradiente descendente estocástico, na qual o gradiente é obtido via retro propagação (RUMELHART; HINTON; WILLIAMS, 1986).

O modelo neural *Skip-gram* possui arquitetura similar a CBOW, no entanto, ao invés de predizer uma palavra dado um contexto, o intuito deste modelo é predizer o contexto $w_{i-t}w_{i-t+1}\ldots w_{i-1}w_{i+1}$ ao redor da i-ésima palavra da sentença w_i. Ou seja, usa-se uma palavra como entrada do classificador e são realizadas predições de palavras que ocorram antes e depois da palavra atual. Quanto maior o intervalo de palavras a ser predito, maior a complexidade computacional. Palavras que geralmente estão mais próximas à palavra de entrada recebem pesos maiores, enquanto palavras mais distantes recebem pesos menores, devido à menor amostragem nos exemplos de treinamento (MIKOLOV; YIH; ZWEIG, 2013).

Novas variações de algoritmos para a criação de *embeddings* ao nível de sentenças, parágrafos e documentos surgiram, considerando os dois modelos anteriores como base. Um algoritmo clássico na literatura é o *Doc2Vec* (D2V) (LE; MIKOLOV, 2014). *Doc2Vec* é um algoritmo não supervisionado que aprende representações de características de tamanho fixo considerando entradas textuais de tamanho variável.

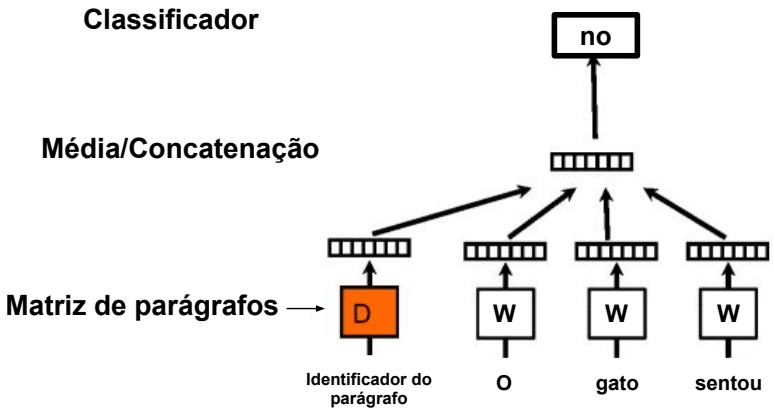
Doc2Vec representa cada texto como um vetor denso, treinado para predizer palavras de um parágrafo. Para isso, concatenam-se vetores de palavras a um vetor de parágrafo e o algoritmo deve ser capaz de predizer a próxima palavra da sentença com base no contexto. O vetor de parágrafo e os vetores de palavras são treinados com gradiente descendente estocástico e retro propagação (*backpropagation*) (RUMELHART; HINTON; WILLIAMS, 1986). Há duas variações do modelo *Doc2Vec*: um modelo de memória distribuída, baseado na arquitetura CBOW, e um modelo de *Bag-of-Words* distribuído, baseado na arquitetura *Skip-gram*.

No modelo de memória distribuída, palavras são mapeadas para vetores únicos, como uma coluna da matriz **W**. Cada parágrafo é também mapeado para um vetor único, como uma coluna da matriz **D**. O vetor de parágrafo e os vetores de palavras são concatenados ou calculam-se suas médias para a predição da próxima palavra no contexto.

A mudança deste modelo em relação ao modelo CBOW está na Equação 2.6, na qual h é constituído por **W** e **D**. O token do parágrafo pode ser pensado como outra palavra, agindo como uma memória que lembra o que está faltando no contexto atual, de acordo com Figura 3.

Os contextos são amostrados a partir de uma janela deslizante de tamanho fixo sobre o parágrafo. O vetor de parágrafo é compartilhado com todos os contextos gerados a partir do mesmo parágrafo, mas não entre parágrafos. Já a matriz de vetores de palavras **W** é compartilhada entre parágrafos. Isso quer dizer que o vetor criado para a palavra "computação" será o mesmo em todos os parágrafos.

Figura 3 – *Framework* para aprendizado de vetores de parágrafos. Um parágrafo é mapeado em um vetor da matriz **D**. A concatenação ou média deste vetor com o contexto das palavras "O", "gato" e "sentou" é usada para predizer a próxima palavra "no". O vetor de parágrafo representa a informação que falta do contexto atual e pode atuar como memória do tópico do parágrafo.



Em cada passo do cálculo de gradiente descendente estocástico, deve-se amostrar o contexto de comprimento fixo de um parágrafo aleatório, computar o erro do gradiente na rede e usar o gradiente para atualizar os parâmetros do modelo.

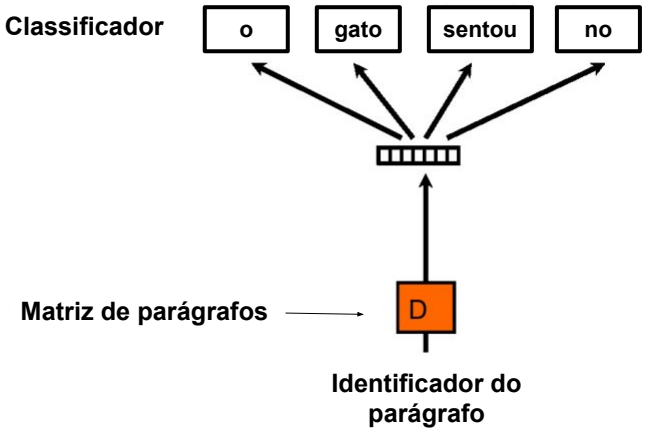
Considerando que o corpus possui N parágrafos, M palavras, que cada parágrafo é mapeado para p dimensões e cada palavra é mapeada para q dimensões, o modelo terá $N \times p + M \times q$ parâmetros (excluindo os parâmetros *Softmax*). Ou seja, *Doc2Vec* é eficiente até mesmo para valores grandes de N, já que as atualizações durante o treinamento são tipicamente esparsas.

O segundo modelo, *Bag-of-Words* distribuída, força o algoritmo a predizer palavras aleatoriamente amostradas a partir do parágrafo de entrada. Ou seja, a cada iteração do gradiente descendente estocástico amostra-se uma janela de texto, em seguida amostra-se uma palavra aleatória da janela de texto, formando uma tarefa de classificação dado o vetor de parágrafo, como apresentado na Figura 4. Além de ser simples, este modelo requer menor armazenamento de dados. É necessário apenas armazenar os pesos *Softmax*.

Em LE; MIKOLOV (2014) foi demonstrado que a melhor forma de gerar vetores de parágrafos é combinar os dois modelos anteriormente apresentados: um aprendido com memória distribuída e outro aprendido com *Bag-of-Words* distribuída. A combinação é geralmente mais consistente e fortemente recomendada.

Modelos de representação baseados em coocorrência de palavras em documentos também foram propostos, como *Global Vectors for Word Representation* (GloVe) (PENNINGTON; SOCHER; MANNING, 2014). GloVe é um modelo não supervisionado de distribuição semântica que aprende *embeddings* de palavras a partir da probabilidade de coocorrência, ou seja, considerando palavras que ocorrem no contexto de outras palavras. O modelo aproveita informações estatísticas treinando apenas elementos não nulos em uma matriz de coocorrência de palavras, ao invés de utilizar toda a matriz esparsa ou janelas de contexto individuais, o que provê vantagens

Figura 4 – Versão *Bag-of-Words* distribuída de vetores de parágrafos. O vetor de parágrafo é treinado para predizer palavras em uma pequena janela. Imagem adaptada de LE; MIKOLOV (2014).



em grandes coleções de textos. Este modelo computa o número de vezes que uma palavra j aparece no contexto de uma palavra i, de modo que palavras semanticamente próximas possuam valores de probabilidade mais altos.

2.2.3 Modelos de Linguagem Dependentes de Contexto

Apesar das abordagens anteriores sanarem parte das desvantagens do modelo *Bag-of-Words*, as *embeddings* geradas por elas são livres de contexto. Isso quer dizer que palavras como "manga" terão sempre as mesmas representações vetoriais, independente do contexto em que elas ocorrem no documento. Para lidar com estes casos, novas abordagens foram propostas na literatura, como *Embeddings from Language Models* (ELMo). ELMo (KHAN *et al.*, 2021) é um modelo contextual de representação de palavras que modela características complexas de palavras e como seus usos variam entre diferentes contextos linguísticos. As *embeddings* geradas por ELMo são aprendidas por meio de um modelo de linguagem bidirecional profundo treinado em um grande corpus textual. A rede neural é uma *Long Short Term Memory* (LSTM). A LSTM (HOCHREITER; SCHMIDHUBER, 1997) é uma herança da arquitetura de Redes Neurais Recorrentes, na qual se mudam as condições de recorrência de como os estados ocultos são propagados. A LSTM possui um vetor oculto chamado "*cell state*", que pode ser visto como um tipo de memória que retém uma parte da informação em estados ocultos anteriores, usando uma combinação de operações parciais de "esquecimento" e "incremento". A natureza da memória permite a modelagem de dependências de longo alcance ou mesmo de padrões específicos persistam por muitos *tokens* (AGGARWAL, 2018; AGGARWAL *et al.*, 2018).

As *embeddings* aprendidas por ELMo são funções aprendidas de estados internos de uma LSTM bidirecional acoplada a um modelo de linguagem pré-treinado em um grande corpus. ELMo usa L camadas de LSTM de propagação (*forward*), que aprendem informações contextuais considerando a sequência de palavras da direita para a esquerda, predizendo o próximo token t_k da sequência com base em informações históricas usando uma camada *Softmax*:

$$p(t_1, t_2, \ldots, t_N) = \prod_{k=1}^{N} p(t_k | t_1, t_2, \ldots, t_{k-1}). \tag{2.7}$$

Os resultados obtidos são combinados com a saída de L camadas de LSTM de retro propagação (*backward*), que executam sobre a sequência inversa de palavras, predizendo o token anterior dado o contexto do futuro:

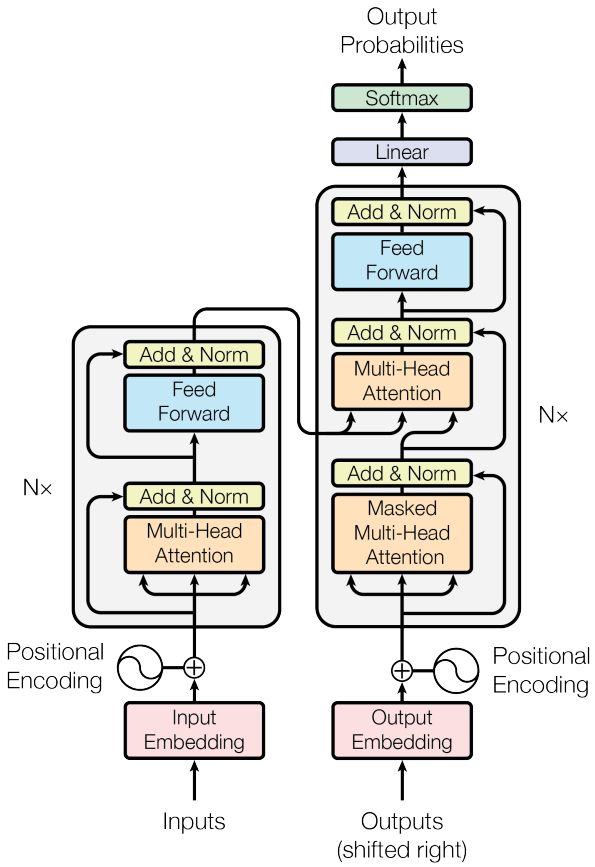
$$p(t_1, t_2, \ldots, t_N) = \prod_{k=1}^{N} p(t_k | t_{k+1}, t_{k+2}, \ldots, t_N). \tag{2.8}$$

A combinação das camadas de propagação e retro propagação é feita pela rede neural bidirecional, que maximiza o log da probabilidade de ambas as direções com o intuito de predizer a próxima palavra da sentença.

Modelos de linguagem neurais como ELMo possuem natureza sequencial, impedindo a paralelização de exemplos de treinamento, o que torna crítica a tarefa de processar sequências de tamanho longo. Para minimizar este problema, modelos de *Transformers* foram projetados com base em um mecanismo de atenção, capaz de modelar dependências globais entre a entrada e saída (VASWANI *et al.*, 2017). Mecanismos de atenção possibilitam a modelagem de sequências de palavras e suas dependências. Nele, um codificador mapeia uma sequência de símbolos de entrada para uma sequência de representações contínuas, e na sequência, o decodificador gera uma sequência de símbolos de saída, um elemento por vez. O modelo é autorregressivo, consumindo os simbolos previamente gerados como entrada adicional para gerar o próximo símbolo, e utiliza uma arquitetura que possui uma pilha de atenção (*self-attention*), com camadas totalmente conectadas entre o codificador e decodificador, mostradas na Figura 5. A seguir os componentes do Transformer são descritos, segundo (VASWANI *et al.*, 2017):

- *Input Embedding*: recebe simultaneamente as palavras da sentença e as mapeia para suas respectivas *embeddings*.
- *Positional Encoding*: armazena informações sobre a posição relativa ou absoluta de um *token* dentro da sequência de palavras, fazendo com que a rede aprenda informações sobre as posições dos *tokens* para sequências de diferentes frequências.
- *Encoder:* o codificador é composto por uma pilha de camadas idênticas. Cada camada tem duas subcamadas. A primeira é o mecanismo de *multi-head self-attention*, e a segunda é uma rede de propagação simples e totalmente conectada. Uma conexão residual é utilizada ao redor das duas sub-camadas, seguida por uma camada de normalização (*add & norm*).
- *Multi-head Self-attention:* permite que o modelo atenda um conjunto de informações de diferentes subespaços de representação em diferentes posições. As entradas e saídas da *multi-head attention* são vetores de atenção correspondentes a cada palavra da sentença.

Figura 5 – Arquitetura do Transformer. Figura retirada de VASWANI *et al.* (2017). Figura em inglês para se manter a compatibilidade com o artigo original.



Tais vetores indicam quais das palavras da sentença são mais relevantes no contexto da palavra que está sendo processada. Mecanismos de atenção permitem ainda a modelagem de dependências internas de uma sentença ou entre sentenças, distinguindo padrões importantes para a construção do modelo de linguagem.

- *Add and Normalization:* recebe valores de entrada e os vetores de saída da camada anterior, os soma e normaliza para facilitar o processo de otimização e garantir que o *positional encoding* se mantenha estável durante o processo.

- *Feed Forward:* é composta por várias redes neurais de propagação que processam palavras e sentenças de forma paralela. Cada rede neural é composta por duas camadas totalmente conectadas.

- *Decoder:* o decodificador, também composto por uma pilha de camadas idênticas, insere uma sub-camada adicional que realiza *multi-head attention* sobre a saída da pilha do codificador. Similar ao codificador, são empregadas conexões residuais ao redor de cada sub-camada, seguida pela camada de normalização. A sub-camada de *self-attention* da pilha do decodificador é modificada para impedir que as posições atendam posições subsequentes. Esta máscara, combinada com que as *embeddings* de saída são deslocados

em uma posição, garante que as predições para a posição *i* possam depender apenas de saídas conhecidas menores do que *i*.

- *Output Embedding:* recebe as palavras da sentença que foram preditas de forma simultânea.
- *Masked Multi-head Self-attention:* seu funcionamento é similar à *multi-head self-attention*, porém utiliza apenas palavras já preditas pelo mecanismo de atenção.
- *Linear:* a saída do decodificador é um vetor numérico, e para que ele seja convertido em uma palavra, são necessários dois processos adicionais. O primeiro deles é o processamento da camada linear. Trata-se de uma rede neural totalmente conectada que recebe o vetor de saída e o projeta em um vetor chamado *logits vector*. Este vetor possui o mesmo tamanho do vocabulário da coleção de documentos, de forma que cada posição deste vetor contenha uma pontuação atribuída a cada palavra.
- *Softmax:* esta camada recebe o *logits vector* e transforma a pontuação de cada palavra em probabilidade. A palavra que possua maior probabilidade associada é selecionada.

A arquitetura do transformers vem sendo amplamente utilizada na criação de diversos modelos de representação baseados em contexto, como *Bidirectional Encoder Representations from Transformers* (BERT). BERT (DEVLIN *et al.*, 2018) foi proposto para pré-treinar representações bidirecionais profundas a partir de dados não rotulados. Para isso, BERT analisa o contexto das palavras da direita para a esquerda e da esquerda para a direita em todas as camadas de sua arquitetura.

BERT primeiramente realiza a etapa de pré-treinamento, na qual coleções de textos da Wikipedia são utilizados como base para o treinar o modelo de linguagem. Em um segundo passo, BERT realiza um refinamento (*fine-tuning*), no qual se ajusta o modelo para a coleção de textos que será utilizada em uma tarefa específica (DEVLIN *et al.*, 2018).

A arquitetura utilizada para pré-treinar e refinar o modelo é a mesma, exceto pela camada de saída, em que todos os pesos são refinados. A etapa de treinamento utiliza um modelo de linguagem de propagação, na qual informação de coocorrência das palavras são inferidas e combinadas a um modelo de linguagem mascarado que realiza a tarefa de predizer a próxima sentença. Além disso, BERT gera *embeddings* para sub-palavras, o que faz o número de *embeddings* menor, reduzindo milhões de palavras para cerca de 30.000 palavras.

2.2.4 Modelos de Representação Baseados em Redes

Os métodos apresentados anteriormente geram representações espaço-vetoriais. Outra forma de representação, que possibilita a modelagem de relações entre distintas entidades de um problema de maneira eficiente, com rica semântica e utilizando um formalismo matematicamente tratável, é por meio de redes complexas (YANG *et al.*, 2020; SHI; PHILIP, 2017).

Redes permitem a extração de padrões de classe que dificilmente são capturados por modelos espaço-vetoriais (BREVE *et al.*, 2012). Documentos textuais podem ser modelados por redes de documentos ou redes documento-termo, na qual a primeira une documentos com conteúdos similares e a segunda modela documentos e termos como nós da rede, na qual arestas ligam os termos presentes em cada documento (ROSSI, 2016).

Uma rede, ou grafo, pode ser definida como uma tripla $G = \langle V, E, W \rangle$, na qual V indica um conjunto de n vértices, E representa o conjunto de m arestas e W representa os pesos destas relações. Dado dois pares de vértices $v_i, v_j \in V$, a relação entre eles é representada por $e_{i,j} = (v_i, v_j)$. Redes podem ser direcionadas, nas quais o sentido da relação é importante; e não direcionadas, nas quais o sentido da relação não importa (ou a relação de um vértice v_i para o objeto v_j tem a mesma importância da relação reversa). Além disso, um vértice pode se relacionar com ele mesmo por meio de autoarestas (ROSSI, 2016).

O peso de uma relação $e_{i,j}$ é dado por $w_{v_i,v_j} \forall v_i, v_j \in V$. As relações existentes entre os objetos de uma rede podem possuir pesos iguais, isto é, $\forall v_i, v_j \in V$, $w_{v_i,v_j} = 1$ se $\exists\, e_{i,j} \in E$. No entanto, em outros tipos de aplicações pode ser relevante considerar relações com pesos distintos, cujas redes são chamadas redes ponderadas. Neste caso, se $\exists e_{i,j} \in E$, pesos w_{v_i,v_j} podem ser equivalentes a qualquer valor real. Em geral, valores positivos são atribuídos como pesos de arestas. Quando o conjunto V é composto por vértices de um mesmo tipo, por exemplo uma rede de notícias, a rede é denominada homogênea. Caso o conjunto V seja composto por h diferentes tipos de objetos, a rede é denominada heterogênea.

Os nós de uma rede podem conter atributos $\mathbf{X} \in \mathbb{R}^{n \times d}$, no qual $\mathbf{x}_v \in \mathbb{R}^d$ representa o vetor de características do nó v. Além disso, as arestas também podem conter atributos $\mathbf{X}^e \in \mathbb{R}^{m \times c}$, no qual $\mathbf{x}^e_{v,u} \in \mathbb{R}^c$ representa o vetor de características de uma aresta (v, u).

Redes podem ser utilizadas como entrada em algoritmos de classificação indutivos ou transdutivos, que realizam a atribuição de rótulos a cada elemento da rede (ROSSI, 2016). Além de servirem de entrada a algoritmos de classificação, as relações complexas existentes na estrutura e os atributos de entrada da rede podem ser explorados e encapsulados para gerar modelos de representação espaço-vetoriais de alto nível, por meio de Graph Neural Networks (GNN) (WU *et al.*, 2020b; ZHOU *et al.*, 2020). Exemplos de GNNs amplamente empregados na literatura que apresentam resultados significativos são *Graph Convolutional Networks* (KIPF; WELLING, 2016), *Graph Attention Networks* (VELICKOVIC *et al.*, 2017) e *GraphSAGE* (HAMILTON; YING; LESKOVEC, 2017), que geram novas representações de baixa dimensão ao nível de nós, arestas ou do próprio grafo.

Representações ao nível de nós podem ser geradas por GNNs por meio da propagação de informações sobre a estrutura do grafo, posteriormente utilizadas para tarefas como classificação ou regressão. Representações ao nível de arestas podem ser úteis em tarefas como predição de links ou peso de conexão entre nós. Já representações ao nível de grafo podem ser utilizadas para classificação ou obtenção de representações compactas.

Segundo ZHOU *et al.* (2020), diferentes tipos de GNNs podem ser construídas a partir da combinação de um conjunto geral de módulos computacionais, como (i) módulo de propagação, responsável por propagar a informação entre os nós, capturando características dos atributos e topologia da rede. Usa geralmente um operador de recorrência ou convolução para agregar a informação de nós vizinhos, enquanto uma operação *skip connection* coleta informações de representações históricas dos nós e mitiga o problema de suavização excessiva (*over-smoothing*); (ii) modulo de amostragem, responsável pela condução da propagação no grafo, necessários quando grafos são muito grandes; e (iii) módulo de *pooling*, que extrai informações dos nós para gerar as representações de alto nível (WU *et al.*, 2020b).

Entre os módulos de propagação, operadores de convolução são mais utilizados para modelos de GNNs. A ideia principal dos operadores de convolução é generalizar a operação de convolução de redes neurais para o domínio do grafo. Avanços desta área podem ser divididos em dois grupos, abordagens espectrais e espaciais (ZHOU *et al.*, 2020).

Abordagens espectrais definem convoluções em grafos não direcionados, introduzindo filtros de processamento de sinais, no qual a operação de convolução é interpretada como a remoção de ruídos a partir de sinais do grafo. Um sinal $\mathbf{x}$ do grafo é primeiramente transformado no domínio espectral pela transformação de Fourier $\mathscr{F}$, e então uma operação de convolução é realizada. Após esta operação, o sinal resultante é transformado novamente por meio da inversa da transformação de Fourier do grafo $\mathscr{F}^{-1}$. Ambas as transformações são definidas na Equação 2.9, na qual $\mathbf{U}$ é a matriz de *eigenvectors* do grafo de Laplace normalizado $\mathbf{L} = \mathbf{I}_N - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$, sendo $\mathbf{D}$ a matriz de graus e $\mathbf{A}$ a matriz de adjacência do grafo. O grafo de Laplace normalizado é real simétrico positivo, logo a matriz normalizada de Laplace pode ser fatorada como $\mathbf{L} = \mathbf{U}\Lambda\mathbf{U}^T$, na qual Λ é a matriz diagonal dos *eigenvalues*.

$$\begin{aligned} \mathscr{F}(x) &= \mathbf{U}^T\mathbf{x} \\ \mathscr{F}^{-1}(x) &= \mathbf{U}\mathbf{x} \end{aligned} \tag{2.9}$$

A operação de convolução pode ser definida conforme a Equação 2.10, com base no teorema da convolução (MALLAT, 1999), na qual $\mathbf{U}^T\mathbf{g}$ é o filtro do domínio espectral.

$$\begin{aligned} \mathbf{g} \star \mathbf{x} &= \mathscr{F}^{-1}(\mathscr{F}(\mathbf{g}) \odot \mathscr{F}(\mathbf{x})) \\ &= \mathbf{U}(\mathbf{U}^T\mathbf{g} \odot \mathbf{U}^T\mathbf{x}) \end{aligned} \tag{2.10}$$

O filtro pode ser simplificado por meio do uso de uma matriz diagonal aprendível $\mathbf{g}_w$, fornecendo assim a função básica dos métodos espectrais $\mathbf{g}_w \star \mathbf{x} = \mathbf{U}\mathbf{g}_w\mathbf{U}^T\mathbf{x}$. Métodos espectrais foram propostos na literatura considerando diferentes filtros $\mathbf{g}_w$ (BRUNA *et al.*, 2013; HAMMOND; VANDERGHEYNST; GRIBONVAL, 2011; DEFFERRARD; BRESSON; VANDERGHEYNST, 2016; KIPF; WELLING, 2016).

Entre os métodos espectrais, *Graph Convolutional Network* (GCN) foi introduzida em 2016 (KIPF; WELLING, 2016) simplificando a operação de convolução e reduzindo o problema de *overfitting*. A Equação 2.11 define de forma compacta a GCN, na qual $\mathbf{X} \in \mathbb{R}^{N \times F'}$ é a matriz de entrada, $\mathbf{W} \in \mathbb{R}^{F \times F'}$ é a matriz de pesos e $\mathbf{H} \in \mathbb{R}^{N \times F'}$ é a matriz convolucionada. F e F' são as dimensões da entrada e da saída, respectivamente.

$$\mathbf{H} = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{X} \mathbf{W} \tag{2.11}$$

Abordagens espaciais tem como ideia principal generalizar a representação de um nó v agregando as características do próprio nó $\mathbf{x}_v$ com as características de nós vizinhos $\mathbf{x}_u$, $u \in N(v)$, definindo convoluções diretamente no grafo com base na sua topologia. No geral, empilha-se um número fixo de camadas convolucionais no grafo, com diferentes pesos em cada uma para tratar dependências da arquitetura. Tais abordagens se desenvolveram rapidamente devido a sua alta eficiência, flexibilidade e generalidade (WU *et al.*, 2020b), tendo como principal desafio definir a operação de convolução considerando diferentes tamanhos de vizinhança dos nós da rede e manter a invariância local das CNNs (ZHOU *et al.*, 2020).

Um framework indutivo que gera *embeddings* amostrando e agregando características de vizinhança local dos nós é o GraphSage (HAMILTON; YING; LESKOVEC, 2017), definido na Equação 2.12, na qual $\mathbf{W}^{t+1}$ é a matriz de pesos dos nós na camada $t+1$ e N_v representa a vizinhança do nó v.

$$\begin{aligned} \mathbf{h}_{N_v}^{t+1} &= \mathrm{AGG}_{t+1}(\{\mathbf{h}_u^t, \forall u \in N_v\}), \\ \mathbf{h}_v^{t+1} &= \sigma(\mathbf{W}^{t+1} \cdot [\mathbf{h}_v^t || \mathbf{h}_{N_v}^{t+1}]). \end{aligned} \tag{2.12}$$

Ao invés de usar o conjunto completo de vizinhos de um nó, GraphSage amostra uniformemente um conjunto fixo de vizinhos para agregar suas informações na representação de um nó alvo. Na Equação 2.12, AGG_{t+1} corresponde a função de agregação, na qual três funções principais podem ser utilizadas: média, LSTM e *pooling*.

Ainda entre as abordagens espaciais, Graph Attention Networks (GAT) (VELICKOVIC *et al.*, 2017) atingiu resultados promissores por incorporar mecanismos de atenção nos passos de propagação, atribuindo diferentes pesos a nós vizinhos, aprendendo de forma implícita suas importâncias para calcular representações ao nível de nós e consequentemente reduzindo o ruído.

Para computar os estados ocultos de um nó v, GAT segue uma estratégia de *self attention*, conforme a Equação 2.13, na qual $\mathbf{W}$ é a matriz de pesos associada com a transformação linear aplicada a cada nó e $\mathbf{a}$ é o vetor de pesos de uma *Multi Layer Perceptron* de camada única.

$$\begin{aligned} \mathbf{h}_v^{t+1} &= \rho\Big(\sum_{u \in N_v} \alpha_{vu} \mathbf{W}\mathbf{h}_u^t\Big), \\ \alpha_{vu} &= \frac{\exp(\text{LeakyReLU}(\mathbf{a}^T[\mathbf{W}\mathbf{h}_v || \mathbf{W}\mathbf{h}_u]))}{\sum_{k \in N_v} \exp(\text{LeakyReLU}(\mathbf{a}^T[\mathbf{W}\mathbf{h}_v || \mathbf{W}\mathbf{h}_k]))} \end{aligned} \tag{2.13}$$

GAT utiliza *multi-head attention* (VASWANI *et al.*, 2017) para estabilizar o processo de aprendizado, aplicando K matrizes de atenção independentes para computar os estados ocultos, e então concatena suas características, ou calcula a média, resultando nas representações de saída da Equação 2.14. Na equação, α_{ij}^k é o coeficiente de atenção normalizado calculado para o k-ésimo mecanismo de atenção. O cálculo de pares nó-vizinho é paralelizável, tornando a operação eficiente, e pode ser aplicável a grafos com nós de diferentes graus, especificando arbitrariamente pesos para diferentes vizinhos.

$$\begin{aligned} \mathbf{h}^{t+1} &= ||_{k=1}^{K} \sigma\Big(\sum_{u \in N_v} \alpha_{vu}^k \mathbf{W}_k \mathbf{h}_u^t\Big), \\ \mathbf{h}^{t+1} &= \sigma\Big(\frac{1}{K} \sum_{k=1}^{K} \sum_{u \in N_v} \alpha_{vu}^k \mathbf{W}_k \mathbf{h}_u^t\Big). \end{aligned} \tag{2.14}$$

Os modelos de representação discutidos nesta seção são amplamente empregados para transformação de textos e grafos em formato estruturado. A decisão tomada nesta etapa sobre qual modelo utilizar influencia diretamente a qualidade de algoritmos de aprendizado para discriminação de conteúdo verídico e falso. A seguir, o problema de detecção de notícias falsas é definido, e são discutidas as principais abordagens propostas na literatura para minimizar o problema, bem como suas lacunas.

2.3 Detecção de Notícias Falsas

Visando identificar e mitigar a propagação de notícias falsas, diversas abordagens automáticas foram propostos na literatura nos últimos anos (BONDIELLI; MARCELLONI, 2019; SHARMA *et al.*, 2019), que podem ser divididas em três grupos distintos: (i) identificação de notícias falsas pelo conteúdo da publicação; (ii) métodos de classificação de notícias com base em contextos; e (iii) soluções dedicadas a restringir o alcance da informação falsa, baseadas em intervenção.

Trabalhos relacionados ao grupo (i) geralmente analisam aspectos léxicos, sintáticos e semânticos presentes no texto da notícia. Abordagens sintáticas envolvem contagens de palavras (sujeitos, verbos, adjetivos), presença e frequência de padrões ou expressões específicas e análises de legibilidade do texto (número de caracteres, palavras complexas, palavras longas, número de sílabas, número de parágrafos) (KSIENIEWICZ *et al.*, 2023; SANTOS; PARDO, 2020; REIS

et al., 2019; VOLKOVA; JANG, 2018; PÉREZ-ROSAS *et al.*, 2017). Outros indicadores de conteúdo falso são pausalidade, incerteza, expressividade, redundância e especificidade (ZHOU *et al.*, 2004). Por meio da análise de características sintáticas, em HORNE; ADALI; PÉREZ-ROSAS *et al.* (2017) foi constatado que notícias falsas são mais similares a sátiras, cuja persuasão está na conformidade das informações contidas no título com as crenças do leitor. Enquanto títulos, geralmente longos, tratam de pessoas ou entidades específicas, o conteúdo das notícias falsas tende a ser curto, repetitivo e pouco informativo. Além disso, palavras sociais (palavras que sugerem interação humana, como *eles*, *companheiro*, *conversando*, *compartilhando*) tendem a ser utilizadas com maior frequência. Estudos também sugerem que notícias falsas possuem maior foco nos tempos verbais presente e futuro, com menor objetividade e com grande presença de advérbios, verbos e caracteres de pontuação. Já no domínio de celebridades, notou-se a presença de palavras perceptivas (como *ouvir*, *ver*, *sentir*) e alto uso de pronomes pessoais.

Abordagens léxicas analisam palavras empregadas no conteúdo textual por meio de *n-gramas*, que são sequências de *n* palavras existentes numa sentença. *N*-gramas podem apontar a presença de humor, abreviações, expressões temporais, ou ainda não cultas, informativas para determinar a veracidade da informação (KSIENIEWICZ *et al.*, 2023; HASSAN *et al.*, 2020; AHMED; TRAORE; SAAD, 2017; RUBIN *et al.*, 2016; MIHALCEA; STRAPPARAVA; PULMAN, 2010). Em RASHKIN *et al.* (2017) foi realizado um estudo para estimar os tipos de léxicos mais frequentes em notícias políticas, comparando a linguagem de notícias reais, enganosas, sátiras e propagandas. Foram analisados léxicos do *Linguistic Inquiry and Word Count* (LIWC) (PENNEBAKER *et al.*, 2015), amplamente utilizado em estudos de ciências sociais; um léxico de palavras subjetivas, que podem ser utilizadas para dramatizar ou sensacionalizar notícias (WILSON; WIEBE; HOFFMANN, 2005); um léxico de palavras vagas e obscuras (TRACY, 2015); além de um léxico com palavras intensificadas (comparativos, superlativos, advérbios de ação e modais), retiradas de *Wiktionary*[1]. Os resultados mostraram que pronomes em primeira e segunda pessoa, assim como palavras exageradas, são utilizados em maior quantidade em notícias enganosas. Notou-se também que escritores de notícias confiáveis buscam ser imparciais, utilizar palavras assertivas e oferecer números concretos, com menor ocorrência de palavras vagas.

Abordagens semânticas relacionam-se a extração de significado presente no texto. Para isso, técnicas avançadas de Processamento de Linguagem Natural, baseadas em aprendizado profundo, passaram a ser amplamente utilizadas para representar palavras e documentos como *embeddings* (KANG; HWANG; YU, 2020; LI *et al.*, 2019; WANG *et al.*, 2018).

Trabalhos relacionados ao grupo (ii) buscam extrair características relevantes considerando todo o ambiente que envolve a notícia, como local de publicação e, no caso de redes sociais, informações relacionadas a postagem, usuários e à fonte (SAIKIA *et al.*, 2022; INAN, 2022; BONDIELLI; MARCELLONI, 2019). Em CASTILLO; MENDOZA; POBLETE (2011)

[1] https://www.wiktionary.org/

foram analisadas características de contexto em postagens do *Twitter* para avaliar a credibilidade da informação divulgada, das quais presença de URL, porcentagem de sentimentos negativos, informações de credibilidade do usuário e profundidade da árvore de propagação se mostraram mais discriminativas.

Trabalhos relacionados ao grupo (iii) comparam padrões de propagação de notícias, sugerindo formas de mitigar o problema. Em SHU; BERNARD; LIU (2019) foram definidas três dimensões que compõem o ecossistema de difusão de notícias: a dimensão do conteúdo, que descreve a correlação entre elementos como notícias, postagens em mídias sociais e comentários; a dimensão social, que envolve relações entre os publicadores, disseminadores e consumidores; e dimensão temporal, que ilustra a evolução das publicações dos usuários e comportamentos da postagem ao longo do tempo. Estas informações podem ser expressas por meio de estruturas de redes homogêneas (redes de relacionamentos, redes de difusão temporal e redes de credibilidade) e heterogêneas (representando relações entre diferentes tipos de entidades, como usuários, postagens e notícias). Estudos deste grupo sugerem que notícias reais se propagam diferente de notícias falsas (ZHAO *et al.*, 2020), e estruturas de redes podem ser exploradas para determinar estratégias como: identificar usuários difusores de conteúdo falso, possibilitando que este usuário seja bloqueado pela rede social; identificar usuários influentes/líderes que possam retificar a informação; ou até mesmo, desenvolver campanhas de prevenção conforme uma necessidade específica. Além disso, propriedades de nós na rede podem ser analisadas como: centralidade; graus de entrada e saída que sugerem se o nó é transmissor ou receptor; e proximidade, na qual revela os nós de maior grau mais próximos de um conjunto que sejam os prováveis transmissores de informação falsa (SHU; BERNARD; LIU, 2019; BARBIER *et al.*, 2013).

O trabalho desenvolvido nesta tese está inserido no primeiro grupo, que busca classificar notícias falsas considerando apenas o conteúdo textual da publicação. Esta característica torna a abordagem aplicável a qualquer tipo de base de dados disponível na literatura, não sendo dependente de características externas à notícia para realizar a classificação. Na próxima seção, são apresentadas bases de dados disponíveis na literatura. Em seguida, são discutidas diferentes algoritmos de extração de padrões usados para mitigar a disseminação de conteúdo falso, bem como as lacunas existentes que dão origem a proposta desta tese.

2.3.1 Bases de Dados para Detecção de Notícias Falsas

A eficácia de classificadores de notícias está intimamente relacionada à qualidade da base de dados utilizada. Embora uma coleção *benchmark* totalmente aceita ainda não tenha sido produzida (BONDIELLI; MARCELLONI, 2019; SHU *et al.*, 2017), coleções têm sido propostas envolvendo objetivos diversos nos últimos anos. Há coleções que abrangem diferentes domínios, isto é, política, religião, famosos, etc, como as descritas por SANTOS; PARDO (2020), e coleções que abrangem domínios específicos. Além disso, bases de dados podem apresentar notícias de sites de checagem, ou até mesmo notícias curtas coletadas em redes sociais (VLACHOS;

RIEDEL, 2014; RIBEIRO, 2019; FAUSTINI; COVÕES, 2019). Há ainda aquelas que fornecem informações adicionais de contexto (SHU *et al.*, 2020), como *links*, imagens e comentários de usuários. A seguir, uma visão geral sobre as coleções de textos comumente utilizadas em tarefas de detecção de notícias falsas é apresentada (SHARMA *et al.*, 2019).

- Bases coletadas de *sites* de notícias:
  - *BuzzFace:* a base foi criada considerando uma coleção de notícias em inglês rotuladas pela agência *BuzzFeed* (SILVERMAN *et al.*, 2016)[2] em quatro graus de falsidade. As notícias foram postadas no *Facebook* por nove veículos de comunicação em setembro de 2016, cujas reações de usuários em comentários foram integradas à base (SANTIA; WILLIAMS, 2018).
  - *BuzzfeedPolitical:* contém notícias reais e sátiras rotuladas pela agência *BuzzFeed*. As notícias são em inglês, relacionadas as eleições presidenciais de 2016 nos Estados Unidos (HORNE; ADALI, 2017; SILVERMAN, 2016).
  - *Fact-checked News:* a base possui notícias em português, e foi resultado de notícias coletadas de agências de checagem de fatos - AosFatos[3], Agência Lupa[4], Fato ou Fake[5], UOL Confere[6] e G1 - Política[7]. A coleção contém 2.168 notícias, das quais 1.124 são reais e 1.044 são falsas (RIBEIRO, 2019). Por ser retirada de sites de checagem, a base de dados apresenta termos como "é falso que", "verificação", etc.
  - *Fake.Br:* a coleção[8] é o primeiro corpus de referência na área de detecção de notícias falsas da língua portuguesa. Suas notícias foram manualmente coletadas e rotuladas. Para cada notícia presente no corpus, são disponibilizadas algumas avaliações de padrões linguísticos correspondentes a ela. Todas possuem formato textual, sendo disponibilizadas em seus tamanhos originais, e truncadas. A vantagem da base truncada é que as notícias possuem número aproximado de palavras, evitando viés no aprendizado. O corpus é composto por 7.200 notícias, distribuídas em 6 categorias - política, TV e celebridades, sociedade e cotidiano, ciência e tecnologia, economia e religião, nas quais 3.600 são reais e 3.600 são falsas (SANTOS; PARDO, 2020).
  - *FakeNews2014:* a base possui sentenças em inglês de dois sites de checagem de fatos: *Channel4*[9] e *Truth-O-Meter* de *Politifact*. Ambos os sites proveem detalhes dos vereditos com rótulos de granularidade fina, como *majoritariamente falso* ou

[2] https://www.buzzfeed.com/
[3] https://aosfatos.org/noticias/
[4] https://piaui.folha.uol.com.br/lupa/
[5] https://g1.globo.com/fato-ou-fake/
[6] https://noticias.uol.com.br/confere/
[7] https://g1.globo.com/politica/
[8] https://github.com/roneysco/Fake.br-Corpus
[9] https://www.channel4.com/news/factcheck/

relativamente verdadeiro. A base armazena data, escritor, rótulo e URL (VLACHOS; RIEDEL, 2014).

- *FakeNewsNet:* o repositório FakeNewsNet[10] contém notícias em inglês de famosos checadas pela agência *GossipCop*[11] e notícias políticas checadas pela agência *PolitiFact*[12]. O repositório disponibiliza o conteúdo das notícias, interações de usuários relacionadas às notícias, localização dos usuários e horários das interações (SHU *et al.*, 2020).
- *FakevsSatire:* a base possui um conjunto de notícias falsas e satíricas coletadas manualmente que abordam diferentes temas relacionados a política americana. Para evitar o viés de classificadores, a coleção não possui mais do que cinco notícias de uma única fonte. As notícias coletadas são posteriores a janeiro de 2016 e possuem um *link* que contesta a informação falsa (GOLBECK *et al.*, 2018).
- *KaggleFN: Kaggle* contém duas coleções de notícias falsas em inglês: A primeira, *Getting Real about Fake News*[13], contém textos e metadados de 244 *websites* considerados não confiáveis, com 12.999 postagens realizadas no período de 30 dias para auxiliar cientistas em estudos de detecção de notícias falsas. A segunda coleção, Fake News[14], contém notícias, títulos e autores de mais de 20 mil notícias em inglês.
- *PoliticalDataset:* a base reúne 75 histórias de notícias reais, fakes e satíricas em inglês. As notícias selecionadas são sobre fatos aleatórios políticos e foram coletadas considerando 24 fontes distintas (HORNE; ADALI, 2017).
- *UELdataset:* a base[15] é composta por notícias balanceadas, divididas em reais, falsas (retiradas de SANTOS; PARDO (2020)) e satíricas, considerando três idiomas: português, inglês e espanhol.
- *George McIntire Dataset:* a base[16] inclui 10.558 notícias em inglês, na qual metade são falsas e metade são reais. Ela contém três colunas, correspondentes ao título, texto e categoria.
- *FakeNewsCorpus:* a base[17] é composta por milhões de notícias coletadas de 1001 domínios. Também estão presentes na base notícias do NYTimes e artigos em inglês do WebHose.

- Bases coletadas em redes sociais:

[10] https://github.com/KaiDMML/FakeNewsNet
[11] https://www.gossipcop.com/
[12] https://www.politifact.com/
[13] https://www.kaggle.com/mrisdal/fake-news
[14] https://www.kaggle.com/c/fake-news/data
[15] http://www.uel.br/grupo-pesquisa/remid/?page_id=145
[16] https://github.com/GeorgeMcIntire/fake_real_news_dataset
[17] https://github.com/several27/FakeNewsCorpus

  - *FacebookHoax:* a coleção consiste de 15.500 postagens coletadas de 18 páginas científicas e 14 páginas conspiratórias do *Facebook* durante o segundo semestre de 2016. Postagens científicas são consideradas reais e postagens conspiratórias são consideradas falsas (TACCHINI *et al.*, 2017).
  - *TwitterDataset:* a base possui 4.392 *tweets* relacionados a notícias falsas, recuperados com palavras-chave presentes em notícias de sites de checagem brasileiros. Para compor o conjunto de notícias reais, foram coletados 4.589 *tweets* postados por sites de notícias de prestígio (FAUSTINI; COVÕES, 2019).
  - *TwitterRumorDetection:* a base contém rumores ocorridos em 2015 e 2016 coletados nos sites de checagem *Emergent* e *Snopes*. Também são disponibilizados *tweets* e interações de usuários dos sites *Twitter*[18] e *Weibo*[19] referentes aos rumores, recuperados por meio de buscas por palavras-chave (MA; GAO; WONG, 2018; MA *et al.*, 2016; LIU *et al.*, 2015).
  - *WhatsAppDataset:* a base é composta por notícias falsas coletadas pelo site *Boatos*[20], que além de checar a informação, posta também o conteúdo original da mensagem. Para compor o conjunto de notícias reais, são coletados primeiros parágrafos de notícias verídicas, que contenham um resumo do que aconteceu, onde aconteceu, quem está envolvido, como aconteceu e o porquê. A base contém notícias brasileiras, na qual 165 notícias são falsas e 12 notícias são reais (FAUSTINI; COVÕES, 2019).

- Bases de sentenças curtas, retiradas de redes sociais, para checagem de fatos:
  - *FEVER: Fact Extraction and VERification* consiste em 185.445 sentenças manualmente verificadas do *Wikipedia*. As sentenças são classificadas como suportada, refutada ou inconclusiva (THORNE *et al.*, 2018).
  - *LIAR:* uma base de dados pública de 12.800 sentenças curtas manualmente rotuladas para detecção de notícias falsas. As notícias foram coletadas do site *PolitiFact*, no qual provê análises detalhadas e *links* para os documentos fontes (WANG, 2017).

Nos experimentos realizados nesta tese, descritos no Capítulo 4, foram utilizadas as bases *FakeNewsNet*, *Fake.Br*, *Fact-checked News* e *Fake News Corpus*. A base Fake.BR foi escolhida por ser o primeiro corpus de referência da língua portuguesa, e conter diferentes tópicos com boa curadoria. A base *Fact-checked News* foi escolhida por apresentar um cenário ideal, contendo notícias de apenas um tópico (política) com diferença clara entre as notícias retiradas de sites jornalísticos e de sites de checagem. A base *FakeNewsNet* foi escolhida por ser muito utilizada na literatura, abranger um domínio novo sobre celebridades e ser desbalanceada. A base *Fake*

[18] https://twitter.com/
[19] https://weibo.com/
[20] https://www.boatos.org

News Corpus foi escolhida por conter milhares de notícias, possibilitando a formação de bases de notícias menores a partir dela, contendo diferentes assuntos e estilos.

2.4 Extração de Padrões para Detecção de Notícias Falsas

Diversos algoritmos de extração de padrões foram propostos nos últimos anos para a tarefa de detecção de notícias falsas. As abordagens abrangem algoritmos supervisionados, semissupervisionados e não supervisionados, cujas notícias são representadas de forma estruturada considerando modelos de representação espaço-vetoriais ou estruturas de redes (CAPUANO *et al.*, 2023; AÏMEUR; AMRI; BRASSARD, 2023; SHAHID *et al.*, 2022; SILVA; FONTES; JÚNIOR, 2020; ZHANG; GHORBANI, 2020; BONDIELLI; MARCELLONI, 2019; MEEL; VISHWAKARMA, 2019; SHARMA *et al.*, 2019).

As notícias podem ser classificadas em duas classes, verdadeiras ou falsas, ou ainda em diferentes tipos de granularidade, considerando que notícias podem conter fragmentos falsos em meio a histórias reais. As notícias podem ser longas, retiradas de sites jornalísticos e mídias não confiáveis de divulgação, ou curtas, retiradas de redes sociais como *Twitter*. Além disso, alguns trabalhos realizam a análise da credibilidade do texto ao nível de sentença. A seguir são discutidos os principais métodos propostos para detecção de notícias falsas.

2.4.1 Algoritmos de Aprendizado Baseados no Modelo Espaço-vetorial

Em RUBIN *et al.*; PODDAR; UMADEVI *et al.* (2016, 2019) foi utilizado *Support Vector Machine* (SVM) (VAPNIK, 2013; CARVALHO *et al.*, 2011) na detecção de conteúdo falso. Em RUBIN *et al.* (2016), o SVM foi enriquecido com características de absurdo, humor, gramática, afeto negativo e pontuação e usado na classificação de 360 notícias satíricas, legítimas e falsas. O autor atingiu 87% de F_1 com 75% de dados rotulados. Em PODDAR; UMADEVI *et al.* (2019) foi atingido 92% de acurácia considerando a base *KaggleFN*, *Bag-of-Words* com esquema de pesos *tf-idf* como modelo de representação e 90% de dados rotulados.

Em ABONIZIO *et al.* (2020) foi utilizado *Random Forest* (RF) (BREIMAN, 2001). Foram coletadas notícias falsas, legítimas e satíricas de três idiomas (inglês, português e espanhol). Os autores representaram notícias com um vetor de 21 características linguísticas, atingindo acurácia de 94% com 80% de notícias rotuladas.

Algoritmos como *K-Nearest Neighbors* (k-NN) (COVER; HART, 1967), *Naïve Bayes* (NB) (LANGLEY *et al.*, 1992), e *Extreme Gradient Boosting* (XGB) (CHEN; GUESTRIN, 2016), também foram avaliados na detecção de notícias falsas (FAUSTINI; COVÕES, 2020; KALIYAR *et al.*, 2020; REDDY *et al.*, 2020; KHANDELWAL; KUMAR, 2020; SINGH; GHOSH; SONAGARA, 2020; YAVARY; SAJEDI; ABADEH, 2020; ASGHAR *et al.*, 2019; BHUTANI *et al.*, 2019; DONG *et al.*, 2019; HAN; MEHTA, 2019; PODDAR; UMADEVI *et al.*, 2019; RASOOL *et al.*, 2019; SOUZA *et al.*, 2020; ZHANG; DONG; PHILIP, 2019). Em

especial, o trabalho desenvolvido por BHUTANI *et al.* (2019), que utiliza NB para predizer e incorporar o sentimento de notícias como atributos na *Bag-of-Words*, obteve 84% de acurácia na classificação de notícias da base *George McIntire* usando 67% das notícias como treinamento. Em REDDY *et al.* (2020) foi empregado XGB, combinando características de estilo de escrita e do texto. Textos foram representados como *Bag-of-Words*, CBOW, *skip-gram* e características linguísticas relevantes. Os autores atingiram acurácia de até 95% nas bases de notícias políticas *FakeNewsNet* e *McIntire Dataset* com 80% dos dados rotulados.

Abordagens que propuseram novos algoritmos também se destacaram na literatura. Em SANTOS; PARDO (2020) foi criado um modelo que reproduz os passos de um usuário realizando uma busca no *Google* por uma notícia. O algoritmo toma uma decisão com base nos primeiros resultados obtidos, procurando por palavras-chave que ajudem a distinguir o conteúdo como falso. A abordagem apresentou 68% de acurácia em notícias da base Fake.Br. Em KANG; HWANG; YU (2020) foi proposto um *framework* que define um vetor latente para cada notícia como a soma de vetores latentes de seus componentes (texto, imagem e evento). Quanto maior a magnitude dos vetores, mais confiável e consistente é uma notícia. Foram utilizadas aproximadamente 9 mil notícias de *Twitter* e *BuzzFeed*. O método atingiu 96% de acurácia usando 80% de notícias rotuladas.

Modelos de aprendizado baseados em redes neurais também foram amplamente propostos, usando técnicas como *Multi-Layer Perceptron* (MLP), *Deep Neural Networks*, *Recurrent Neural Networks* (RNN), *Convolutional Neural Network* (CNN), *Gated Recurrent Unit* (GRU), *Long Short Term Memory* (LSTM) e *Transformers* para detecção de notícias falsas (CAPUANO *et al.*, 2023; KALIYAR *et al.*, 2020; KANG; HWANG; YU, 2020; QAZI; KHAN; ALI, 2020; WANG *et al.*, 2020; WU *et al.*, 2020a; YAVARY; SAJEDI; ABADEH, 2020; BONDIELLI; MARCELLONI, 2019; MEEL; VISHWAKARMA, 2019; SHARMA *et al.*, 2019; DONG *et al.*, 2019; MONTI *et al.*, 2019; ZHANG; DONG; PHILIP, 2019; LI *et al.*, 2019).

Em LI *et al.* (2019) foi proposto uma CNN multinível que extrai características convolucionais locais e semânticas globais de notícias. Os autores atingiram 92% de acurácia nas bases *LIAR* e *Kaggle FN*, usando 80% das notícias como treinamento.

Em UPPAL; SACHDEVA; SHARMA (2020) analisou-se a estrutura de segmentos de discurso. Sentenças foram representadas por uma RNN bidirecional e uma matriz armazenou relações ao nível de discurso. Representações de documentos foram obtidas pelo cálculo de propriedades das sentenças com uma função de ativação. O modelo atingiu 74% de acurácia com 97% de dados rotulados, aplicado a aproximadamente 6.500 notícias de *BuzzFeed* e *PolitiFact*.

Em KALIYAR *et al.* (2020) foi proposto uma rede neural que aprende características discriminativas em camadas ocultas de uma CNN, atingindo 98% de acurácia com 90% de notícias rotuladas. Em WU *et al.* (2020a) foi proposto um modelo baseado em redes neurais adversárias, que permite a redução de características irrelevantes textuais, extraídas para a avaliação de credibilidade. O modelo foi testado nas bases *LIAR*, *Weibo* e *TwitterRumorDetection*,

atingindo 42% de acurácia considerando diferentes graus de falsidade e 90% de notícias rotuladas.

Algoritmos híbridos (*ensembles*), também foram propostos para detectar conteúdo enganoso. DONG *et al.* (2019) propôs um *ensemble* baseado em *attentive RF* e redes neurais profundas. *Attentive* RF seleciona pistas discriminativas de indivíduos, textos e correlação de interações sociais de forma adaptativa, enquanto RNNs capturam relações ocultas complexas em informações textuais. O modelo fornece distribuição de probabilidade como predição de notícias de PolitiFact e Facebook, com aproximadamente 81% de acurácia e 80% de dados rotuladas.

Dispensando recorrências e convoluções, QAZI; KHAN; ALI (2020) utilizou *Transformers* para classificação da base *LIAR*, considerando 4 rótulos distintos. Enquanto CNN híbrido atingiu 27% de acurácia, o modelo proposto atingiu aproximadamente 43% usando 90% de dados rotulados. Abordagens recentes propuseram ainda o uso de algoritmos de aprendizado profundo pré-treinados, como ELMo (KHAN *et al.*, 2021) e BERT (WANI *et al.*, 2021; KALIYAR; GOSWAMI; NARANG, 2021), além de diversas variações envolvendo aprendizado profundo, atingindo resultado estado da arte (DHIMAN *et al.*, 2023; CAPUANO *et al.*, 2023).

As abordagens discutidas nesta seção usam algoritmos de aprendizado que consideram como entrada representações de notícias baseadas no modelo espaço-vetorial. Na próxima seção são discutidas abordagens baseadas em redes, que permitem a modelagem de diferentes entidades de um problema, além da extração de diferentes padrões a partir das conexões existentes.

2.4.2 Algoritmos de Aprendizado Baseados em Redes

Em GUACHO *et al.* (2018) foi utilizada propagação de rótulos para classificação de notícias e usuários maliciosos. Os autores representaram coleções de notícias como *embeddings*, utilizando-as para criar uma rede. O algoritmo utilizado foi *Fast Belief Propagation*, que propaga rótulos de notícias conhecidos para notícias não rotuladas de acordo com suas relações de proximidade na rede. Os autores atingiram 75.43% de acurácia com 30% de notícias rotuladas.

Em SHU; BERNARD; LIU (2019) foram modeladas relações entre notícias, usuários e publicadores em uma rede heterogênea chamada TriFN. Os autores consideraram que o viés existente entre o publicador, a veracidade do conteúdo e os engajamentos de usuários podem auxiliar no processo de classificação, e exploram estes relacionamentos para o aprendizado de representação de características, usando um classificador linear semi-supervisionado que aprende uma função para predizer nós não rotulados. Os autores atingiram aproximadamente 88% de F_1 nas bases de dados BuzzFeed e *Politifact*, usando 80% de dados rotulados e validação cruzada.

Em ZHANG; DONG; PHILIP (2019) a detecção de notícias falsas foi tratada como um problema de inferência de credibilidade, por meio de um modelo de GNN. O modelo infere rótulos de credibilidade de notícias, criadores e assuntos com uma rede neural difusa, em um grafo composto por 14.000 notícias, 3.600 criadores e 152 assuntos de *PolitiFact*. Na classificação binária foram atingidos 80% de $F1$ usando 90% de dados rotulados.

Em YANG *et al.* (2019) o problema de detecção de notícias falsas foi tratado com aprendizado não supervisionado. O autor explorou o engajamento de usuários para identificar suas opiniões quanto à autenticidade das notícias, modelando as informações como uma rede Bayesiana que captura dependências condicionais entre a veracidade das notícias, as opiniões e credibilidades de usuários. O modelo atingiu 75% de acurácia nas bases *LIAR* e *BuzzFeed*.

Em REN *et al.* (2020) foi proposto uma GNN Heterogênea Adversária com Aprendizado Ativo para detecção de notícias falsas. Um mecanismo de atenção hierárquico foi usado para aprender representações de nós, enquanto um seletor foi responsável por consultar candidatos de alto nível para o aprendizado ativo. Foram avaliadas duas bases de dados: a primeira contendo 14.055 notícias, 3.634 criadores e 152 assuntos e a segunda contendo 182 notícias, 15.257 usuários e 9 publicadores. Com 20% de notícias reais e falsas rotuladas, o algoritmo atingiu de 57% a 70% de macro F_1.

Em SANTOS; PARDO (2020) e KHANDELWAL; KUMAR (2020) foram propostos grafos de conhecimento para checagem de fatos, que armazenam relações textuais de *sujeito-predicado-objeto*. Em SANTOS; PARDO (2020), o grafo foi construído com informações do *Wikipedia*, no qual dada uma sentença, se ela existisse dentro do grafo de conhecimento era considerada verdadeira. A abordagem atingiu aproximadamente 74% de acurácia, treinada com 231 entidades e 33.385 sentenças. Em KHANDELWAL; KUMAR (2020) foi utilizado o Wikipedia como fonte de informação não estruturada para encontrar evidências relacionadas ao conhecimento disponível no grafo, gerado a partir de informações do Wikidata. Uma base de dados contendo triplas e rótulos foi construída, na qual o algoritmo RF apresentou o melhor desempenho, variando de 60 a 97% de acurácia.

Em YU *et al.* (2020) foi proposto o IARNet, uma rede de informação, agregação e discurso que une a fonte da postagem, comentários e usuários como nós e interações entre estes elementos como arestas. Foram avaliadas duas bases de dados, *Weibo* e *Fakeddit*. Com 70% de dados rotulados, os autores atingiram 96% de acurácia. Algoritmos como *Hidden Markov Models*, *Conditional Random Fields* e *Propagation Tree Kernel* também foram propostos considerando redes complexas e detecção de rumores, permitindo a modelagem de relacionamentos temporais, de usuários e dando suporte a propagação de informações (BONDIELLI; MARCELLONI, 2019; SOUZA *et al.*, 2020; YAVARY; SAJEDI; ABADEH, 2020; ASGHAR *et al.*, 2019).

Em WANG *et al.* (2020) foi proposto um grafo multimodal convolucional orientado a conhecimento para modelar representações semânticas, unindo informação textual, de conhecimento e visual em um *framework* para detecção de notícias falsas. Palavras das notícias foram convertidas em um grafo, capaz de modelar frases não consecutivas, obtendo composições semânticas. Informações visuais e conhecimento do mundo real também foram convertidas em nós do grafo, a fim de incluir informações complementares. Foram utilizadas duas bases de notícias, compostas por textos e imagens. O autor atingiu 89% de acurácia na base *Pheme* com 70% de dados de treinamento.

Em NI; LI; KAO (2021) foi proposto uma rede de atenção *Multi-View* (MVAN), um modelo usado para detectar fake news e prover explicações para redes sociais. MVAN considera dois mecanismos de atenção, um sobre a semântica do texto e outro sobre a estrutura de propagação. Os mecanismos capturam informações e pistas no conteúdo do tweet e na estrutura de propagação da rede, que contém usuários e suas características, atingindo de 92% a 93% de F_1 nas bases de dados Twitter15 e Twitter16 com 70% de dados rotulados.

Em SAIKIA *et al.* (2022) foi proposto um método baseado em padrões de propagação e contexto de mídias digitais. O grafo integra uma rede neural com propagação de tweets sobre notícias e representações de *encoder* bidirecional do conteúdo das notícias para o aprendizado de características textuais. Os autores usam informação temporal dos tweets para estruturar o grafo, considerando características ao nível de nó e do grafo. Com 70% e 90% de dados rotulados de *Politifact* e *Gossipcop*, os autores atingiram de 91 a 93% de F_1.

Em XU *et al.* (2022) foi desenvolvido um *framework* baseado em grafos. Foram extraídos trechos de evidência e modelados textos de usuários em uma rede. Os nós indicavam palavras e as arestas relacionavam palavras que coocorriam nos textos. Foram utilizados mecanismos de atenção para criar representações, e as interações entre os trechos de evidência e os textos de usuários foram integradas para predição da veracidade dos textos de usuários. O modelo atingiu 69% e 80% de macro F_1 em *Politifact* e *Snopes* com 80% de dados de treinamento.

Em INAN (2022) foram capturados usuários com maior probabilidade de compartilharem informação enganosa, usando GNNs. Em seguida, foi gerado um grafo incluindo comentários, descrições dos perfis de potenciais usuários e suas notícias compartilhadas. Considerando o conteúdo deste grafo, foram gerados codificadores (*encoders*) e realizada a classificação de notícias usando GAT com arestas de pesos. Com *Politifact* e *GossipCop*, 90% de dados rotulados e validação cruzada, os autores atingiram de 86% a 93% de F_1.

Em CUI *et al.* (2023) foi usado a união de informações inter e intra grafos para a propagação de informações na rede com um tensor de grafo de textos de terceira ordem. A abordagem extrai informações de relações sequencial, sintática e semântica entre palavras e propaga informações dentro e entre grafos. O autor realiza *data augmentation* para balancear as bases de dados, e representações de notícias são geradas por mecanismos de atenção ao nível de nó e grafo, alimentando um classificador. Com 80% de dados rotulados, 10% de validação e 10% de teste, os autores atingem de 86% a 96% de acurácia.

Abordagens baseadas em redes, no geral, unem na estrutura elementos que são internos e externos as notícias, isto é, informações de usuários, comentários, fontes, etc., demandando bases de dados de notícias que contenham informações específicas para tratar o problema. Além disso, a maioria dos trabalhos discutidos utilizam algoritmos de aprendizado supervisionados na classificação, que demandam conjuntos de treinamento representativos e balanceados para haver boa discriminação sobre as classes do problema. No entanto, rotular um amplo conjunto de notícias é uma tarefa custosa. Para minimizar os efeitos de rotulação, abordagens que classificam

notícias considerando conjuntos de treinamento que contenham apenas dados de uma única classe de interesse vêm sendo propostas na literatura. A seguir, tais abordagens são apresentadas.

2.4.3 Algoritmos de Aprendizado de Uma Única Classe e Aprendizado Positivo Não Rotulado

O trabalho de FAUSTINI; COVÕES (2019) foi o primeiro a propor o uso de aprendizado de uma única classe para detecção de notícias falsas. Considerando apenas características de notícias falsas, os autores propuseram o algoritmo *DCDistanceOCC*, baseado em redução de dimensionalidade. O algoritmo cria um vetor classe a partir da soma de vetores linguísticos de todas as notícias. Em seguida, calcula um valor único para cada notícia, correspondente a sua distância do vetor classe. Um novo objeto é considerado falso se sua distância para o vetor classe está acima de um limiar. O desempenho do método foi comparado à outros três algoritmos de OCL: EcoOCC (inspirado no algoritmo *k-Means*) (SALMAZZO, 2016), *NB Positive Class* (DATTA, 1997) e *One-Class SVM* (PLATT *et al.*, 1999). Os algoritmos foram executados com validação cruzada, na qual 90% das notícias falsas foram usadas para treino. Os resultados variaram de 54% a 67% para as bases *Fake.Br* e *WhatsAppDataset*. Outros autores também utilizaram OCL para encontrar rumores (tratados como anomalias) em redes sociais, com base em distância euclidiana e similaridade de cosseno (CHEN *et al.*, 2016); medindo diferenças de padrões de disseminação de informações, atribuindo pontuações de anomalias aos dados não rotulados com *One Class Conditional Random Fields* (ZHAO *et al.*, 2014).

Em LIU; WU (2020) foi proposto o *Fake News Early Detection* (FNED), no qual um extrator de características sensível a respostas de multidões extrai características textuais e de usuários, e um mecanismo de atenção destaca respostas de usuários importantes. O método detecta fake news com 90% de acurácia, usando 10% de notícias falsas rotuladas (conjunto P) e aprendizado positivo e não rotulado. A partir do conjunto não rotulado, notícias pseudo-reais são selecionadas aleatoriamente, compondo o conjunto N', com $|N'| = |P|$. Estes dois conjuntos são usados para treinar um classificador neural. O processo é repetido por k vezes, e todos os classificadores treinados são unidos para formar um classificador final. O classificador final rotula o conjunto de notícias não rotulado, e as top n notícias classificadas como falsas incrementam o conjunto P. A abordagem é avaliada em duas bases de dados, com 680 e 4.664 notícias.

Em OLIVEIRA; MEDEIROS; MATTOS (2020) foi proposto uma análise computacional de estilo baseada em Processamento de Linguagem Natural usando dados do Twitter. Notícias foram representadas com BoW e tf-idf, mantendo apenas palavras essenciais para o entendimento da ideia central do texto por meio de técnicas de redução de dimensionalidade. O autor usou One-Class Support Vector Machine (OCSVM), atingindo 86% de acurácia e 65% de precisão com 90% de notícias falsas rotuladas.

Embora OCL e PUL reduzam esforços de rotulação, os trabalhos citados possuem

limitações. A abordagem proposta por FAUSTINI; COVÕES (2019) requer a calibração de limiares, além de não apresentar desempenho satisfatório mesmo com 90% de dados rotulados. Em OLIVEIRA; MEDEIROS; MATTOS (2020) foi utilizado apenas o algoritmo OCSVM para avaliação dos resultados. Em LIU; WU (2020) é empregado subamostragem nos dados não rotulados, o que não é ideal em cenários de detecção de fake news do mundo real, além de requerer informações externas a notícia, como comentários de usuários e perfis de usuários, aplicável apenas a bases de dados específicas.

Diante do cenário apresentado, a seguir são discutidas as principais limitações das abordagens, que dão origem a proposta desta tese.

2.5 Considerações Finais

Apesar de apresentarem resultados satisfatórios, as abordagens de detecção de notícias falsas possuem limitações que ainda persistem na literatura até os dias atuais (MISHRA; SHUKLA; AGARWAL, 2022; SHAHID *et al.*, 2022; AÏMEUR; AMRI; BRASSARD, 2023; ZHANG; GHORBANI, 2020; SILVA; FONTES; JÚNIOR, 2020; BONDIELLI; MARCELLONI, 2019; MEEL; VISHWAKARMA, 2019; SHARMA *et al.*, 2019). Como discutido anteriormente, a forma mais comum de modelar o problema é utilizando algoritmos de classificação supervisionados binários ou multiclasse, que requerem um conjunto significativo de dados rotulados para extração de padrões discriminativos. No entanto, rotular muitas notícias é uma tarefa custosa, que além de consumir tempo, pode ser enviesada pelo rotulador.

Outro problema que surge sobre o avanço das notícias falsas é que elas podem ser encontradas em diferentes tipos de mídias sociais. Ou seja, a escrita das notícias evolui e assume características que as tornam atrativas para atingir usuários de diferentes perfis. Além disso, tarefas específicas demandam bases de dados específicas. Ou seja, um algoritmo treinado com notícias políticas pode não discriminar bem notícias de tópicos distintos, como religião. O mesmo vale para diferentes linguagens, demandando a constante atualização de bases de dados de treinamento para que os algoritmos de detecção de conteúdo falso obtenham bom desempenho em dados do mundo real. No entanto, não há uma base de dados que abranja estas limitações, nem um consenso sobre diretrizes que devem guiar o processo de classificação de notícias.

A forma com que as bases de dados são coletadas e pré-processadas influenciam no resultado dos algoritmos. Algumas bases são construídas de forma semi-automática, na qual sites de origem duvidosa e jornalísticos são mapeados, e um *crawler* realiza a coleta do conteúdo. No entanto, se estes dados não forem devidamente tratados, podem enviesar o comportamento de algoritmos, resultado em ótimos classificadores dentro daquele domínio, porém que não são efetivos em contextos distintos. Estas limitações motivam estudos de abordagens de classificação de notícias que desempenhem bem diante de poucos dados rotulados.

Para reduzir esforços de rotulação, nesta tese assume-se que o aprendizado baseado

em uma única classe, em especial o aprendizado positivo e não rotulado (BEKKER; DAVIS, 2020), que aprende um modelo a partir de poucos dados rotulados da classe de interesse, usando informação dos dados não rotulados para aumentar o desempenho de classificação, seja uma abordagem promissora para atacar o problema de detecção de notícias falsas.

Além dos benefícios de PUL, o modelo de representação utilizado influencia diretamente a qualidade da abordagem, principalmente no uso de dados não rotulados no processo de aprendizado (ENGELEN; HOOS, 2020). Neste contexto, representações baseadas em redes podem unir informações de múltiplos tipos de dados de forma efetiva, modelando objetos complexos e suas relações com rica semântica (YANG *et al.*, 2020; SHI; PHILIP, 2017). Além disso, representações baseadas em redes permitem a extração de padrões que dificilmente são capturados por modelos espaço-vetoriais (BREVE *et al.*, 2012), sendo úteis inclusive no aprendizado semi-supervisionado (ROSSI; LOPES; REZENDE, 2016). Geralmente, o processo de aprendizado em uma rede é realizado por um algoritmo de propagação de informações, o que permite a manipulação de grandes volumes de dados (HUA; YANG; QIU, 2021; ZHU; GOLDBERG, 2009). Portanto, nesta tese serão exploradas diferentes redes homogêneas e heterogêneas, que representem notícias e elementos que possam ser extraídos do texto da publicação. Espera-se que a adição de informação relevante na rede aumente o desempenho de algoritmos de extração de padrões a partir de um conjunto pequeno inicialmente rotulado.

Considerando a discussão realizada previamente, no Capítulo 3 são apresentados fundamentos teóricos sobre algoritmos de detecção de padrões baseados em uma única classe aplicáveis a dados textuais, que podem estar representados por modelos espaço-vetoriais e redes. Também são discutidos algoritmos de propagação de informações, que possam ser aplicados a estruturas de redes para classificação de nós não rotulados. Parte dos algoritmos serão utilizados tanto para compor a abordagem, quanto para avaliar seu desempenho.

CAPÍTULO

3

APRENDIZADO DE UMA ÚNICA CLASSE

No Capítulo 2 foram apresentados conceitos sobre Mineração de Textos, técnicas de pré-processamento, discutidos o problema de detecção de notícias falsas, lacunas encontradas na literatura e definido o escopo desta tese, que consiste na detecção de notícias falsas utilizando aprendizado semissupervisionado baseado em uma única classe, cujas notícias são modeladas em uma rede heterogênea. Diante deste cenário, neste capítulo são apresentados algoritmos de extração de padrões de uma única classe aplicáveis a dados textuais.

Métodos de classificação supervisionados binários, ou multiclasse, visam classificar um objeto desconhecido considerando duas ou mais categorias previamente definidas. Quando um objeto não pertence a nenhuma destas categorias, surge um problema. Considere um domínio no qual notícias são classificadas como *economia* ou *educação*. Caso surjam notícias pertencentes a esses domínios, um classificador binário pode resolver este problema. Porém, se surgir uma nova notícia sobre *esportes*, ela será classificada incorretamente como *economia* ou *educação*. Além disso, algoritmos supervisionados multiclasse necessitam de conjuntos de treinamento balanceados e representativos de cada categoria existente, não desempenhando bem quando uma das classes não é suficientemente amostrada (KHAN; MADDEN, 2014; TAX, 2001).

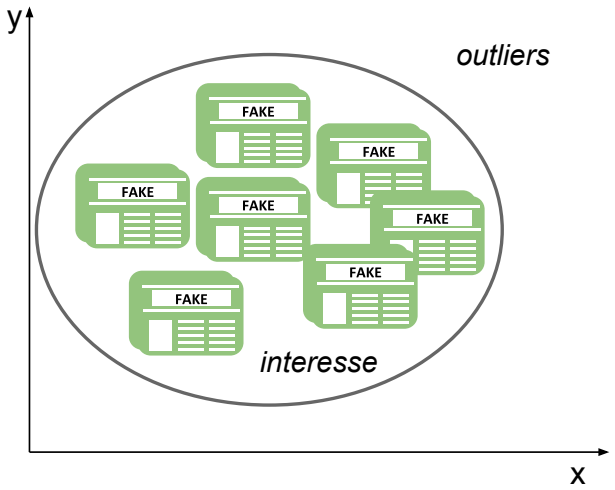
No contexto descrito anteriormente, o Aprendizado de uma Única Classe, ou *One-class Learning* (OCL), também conhecido como detecção de anomalias (*outlier detection* ou *anomaly detection*), ou ainda detecção de novidades (*novelty detection*), pode ser adequado para classificar documentos como pertencentes ou não pertencentes a uma determinada classe de interesse (WANG; BAH; HAMMAD, 2019; KHAN; MADDEN, 2014; TAX, 2001). Neste tipo de aprendizado, uma das classes é bem caracterizada por instâncias nos dados de treinamento, chamada classe de interesse. Exemplos pertencentes a outras classes, considerados de não interesse, ou não são caracterizados, ou a quantidade de instâncias existentes não possui representatividade estatística dentro do conjunto inicialmente rotulado.

Algoritmos de OCL que consistem em encontrar os objetos não caracterizados no

conjunto de treinamento, ou seja, aqueles que diferem dos demais como se fossem suspeitos de serem gerados por outro mecanismo (HAWKINS *et al.*, 2002), são conhecidos como detectores de anomalias. Anomalias podem ser interpretadas de várias formas, dependendo do domínio de aplicação. Exemplos de anomalias considerando ambientes textuais são (RUFF *et al.*, 2019; KANNAN *et al.*, 2017): gerenciamento de páginas *web*, de forma que sejam monitorados eventos não usuais, como postagens mal intencionadas, descrições ou revisões de produtos enganosos; monitoramento de transações incomuns que apontem comportamentos fraudulentos; detecção de *spam*; e monitoramento de notícias, procurando por notícias que possuam informações novas.

Um exemplo de detecção de anomalia pode ser visualizado na Figura 6. Em um cenário de notícias, considere que a classe de interesse é composta por notícias falsas, rotuladas com $c_d = +1$. A linha sólida representa um possível classificador OCL, que aprende a caracterizar a classe alvo considerando diversos atributos que identifiquem a falsidade de uma notícia. Na Figura 6, exemplos de não interesse, que seriam notícias verdadeiras, não estão disponíveis na fase de treinamento. Desta forma, caso um exemplo de notícia verdadeira venha a ser classificado, provavelmente receberá uma baixa pontuação, sendo considerada anômala, cuja classe corresponde a $c_d = -1$.

Figura 6 – Classificador OCL que distingue notícias falsas de outros tipos de objetos. Figura elaborada pela autora, inspirada em TAX (2001).



Dada a presença de anomalias, é importante fazer uma análise de características que melhor discriminem objetos de interesse de anômalos. Esta análise não é uma tarefa trivial. Considere, por exemplo, objetos representados como vetores de atributos multidimensionais em um espaço de características. Um objeto pode apresentar valores anômalos em alguns atributos e valores considerados normais em outros. Além disso, um objeto pode ser anômalo até mesmo quando nenhum de seus atributos individuais são anômalos. Considere que objetos sejam pessoas, e seja comum pessoas pesarem 15 kg (crianças) e medirem 1,70 metros (adultos). Um objeto anômalo poderia apresentar tais medidas juntas. Um objeto também pode parecer anômalo com relação a outros, porém semelhante a objetos locais. Tais situações trazem a necessidade de que uma definição geral de anomalia seja especificada, uma análise de como valores de múltiplos

atributos podem ser utilizados para determinar se um objeto é anômalo, bem como quais testes podem ser aplicados para identificar o bom funcionamento do sistema (TAN *et al.*, 2019; KHAN; MADDEN, 2014; TAX, 2001).

Algumas observações adicionais sobre abordagens OCL podem ser feitas (TAN *et al.*, 2019): (i) alguns algoritmos realizam a classificação de um objeto considerando a forma binária, isto é, retornam se o objeto é ou não anômalo. Porém, esta técnica não reflete se alguns objetos são mais anômalos que outros, tornando desejável avaliações quantitativas que informam o quanto o objeto é anômalo por meio de pontuações; (ii) enquanto algumas técnicas removem anomalias uma por vez, nas quais o objeto mais anômalo é identificado e removido de forma iterativa, em outras um conjunto de anomalias é identificada de uma vez; (iii) se a classe de anômalos é muito menor do que a classe de interesse, medidas de avaliação de desempenho como precisão, revocação e taxa de falsos positivos serão mais apropriadas do que acurácia.

No geral, algoritmos de OCL podem ser divididos em três abordagens: não supervisionados, supervisionados ou semissupervisionados. Algoritmos supervisionados requerem um conjunto de treinamento com objetos de interesse. O conjunto também pode ter objetos anômalos, porém em baixa quantidade. Após aprenderem um modelo de classificação, tais algoritmos são responsáveis por classificar um novo objeto $\mathbf{d}_i$ de acordo com uma pontuação atribuída ($f(\mathbf{d}_i)$), como na Equação 3.1. Se a pontuação está acima de um limiar, o objeto é classificado como de interesse, caso contrário, como anômalo.

$$class = \begin{cases} f(\mathbf{d}_i) \geq limiar \rightarrow interesse \\ f(\mathbf{d}_i) < limiar \rightarrow anômalo \end{cases} \tag{3.1}$$

Diferente dos algoritmos supervisionados, algoritmos semissupervisionados possuem como objetivo encontrar um rótulo ou pontuação para um dado objeto utilizando informações apenas de objetos rotulados da classe de interesse. Além disso, informações de objetos não rotulados podem ser úteis para aprimorar tais modelos de classificação, como é o caso de algoritmos conhecidos como Aprendizado Positivo e não Rotulado, ou do inglês *Positive and Unlabeled Learning* (PUL). Uma vantagem de algoritmos semissupervisionados é a facilidade em se encontrar dados não rotulados. Uma limitação desta abordagem é que em muitas situações práticas, pode ser difícil encontrar um pequeno conjunto representativo de objetos de interesse (BEKKER; DAVIS, 2020).

Algoritmos não supervisionados não consideram informações de rótulos durante o processo de aprendizado. Desta forma, o objetivo é atribuir uma pontuação para cada instância que reflita seu grau de anomalia. Um problema do aprendizado não supervisionado é que se muitas anomalias similares estiverem disponíveis nos dados, elas podem ser classificadas erroneamente como exemplos da classe de interesse (TAN *et al.*, 2019).

Nos últimos anos, diferentes algoritmos de aprendizado de uma única classe foram propostos, baseados em distância, agrupamento, densidade, além de métodos baseados em gra-

fos (ELTANBOULY *et al.*, 2020; AKOGLU; TONG; KOUTRA, 2015), Aprendizado Ativo (PIMENTEL *et al.*, 2018) e Aprendizado Profundo (CHALAPATHY; CHAWLA, 2019; MANEVITZ; YOUSEF, 2007). No entanto, limitações ainda são encontradas, como (WANG; BAH; HAMMAD, 2019): (i) a falta de estudos que caracterizem e relacionem completamente métodos de detecção de anomalias a dados reais, principalmente se tratando de dados multidimensionais, como representações de textos; (ii) em geral, métodos de detecção procuram por anomalias locais. Porém, determinar correlações locais de exemplos em espaços de alta dimensão é desafiador, sendo um problema de pesquisa em aberto; (iii) examinar a influência de certas características que divergem nos exemplos de treinamento para selecionar aquelas apropriadas para detecção de anomalias é um problema, principalmente tratando-se de dados multidimensionais aos quais técnicas já existentes não apresentam eficácia; e a (iv) necessidade de algoritmos escaláveis, que apresentem bom desempenho em bases muito grandes a um tempo mínimo de execução.

Nas próximas seções são descritas as principais categorias de algoritmos de OCL (WANG; BAH; HAMMAD, 2019; TAN *et al.*, 2019). Na Seção 3.1 são discutidos métodos estatísticos. Na Seção 3.2 são discutidos métodos baseados em densidade e distância. Na Seção 3.3 são apresentados métodos baseados em agrupamento. Na Seção 3.4 são apresentados métodos de aprendizado profundo. Na Seção 3.5 são abordados métodos baseados em GNNs. Na Seção 3.6 são discutidos algoritmos de Aprendizado Positivo e não Rotulado, que recebem como entrada dados estruturados como vetores e redes. Também são apresentados algoritmos semissupervisionados consolidados na literatura que podem ser aplicados na segunda etapa de abordagens PUL, para classificação de nós não rotulados. Por fim, na Seção 3.7 são apresentadas lacunas dos algoritmos baseados em uma única classe quando aplicados ao contexto de detecção de notícias falsas, e apresentado o algoritmo base utilizado nesta tese.

Como este projeto tem interesse por algoritmos de classificação de notícias, os algoritmos investigados serão adaptados para classificação de dados textuais. Desta forma, um conjunto de documentos será denotado como $\mathscr{D} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n\}$, um conjunto de atributos será denotado por $\mathscr{T} = \{t_1, t_2, \ldots, t_m\}$, e o peso de um atributo t_j em um documento d_i será denotado por w_{d_i,t_j}.

3.1 Métodos Estatísticos

Métodos estatísticos criam um modelo de distribuição a partir dos dados para estimar parâmetros de distribuição de probabilidade. Logo, um objeto pode ser tratado como anomalia se ele não se adequa ao modelo de distribuição dos dados (TAN *et al.*, 2019).

Muitas coleções de dados podem ser modeladas por distribuições estatísticas, como a distribuição normal, na qual a probabilidade de um exemplo diminui conforme sua distância do centro da distribuição aumenta. Esta distribuição considera que a maioria dos exemplos estão próximos ao centro e a probabilidade de um exemplo diferir significativamente da maioria é pequena. Outras distribuições bem conhecidas (SCHEAFFER; YOUNG, 2009) são a distribuição

de Poisson e multinomial. Porém, aplicar uma distribuição sobre os dados exige um conhecimento do domínio de aplicação. Muitos conjuntos de dados possuem distribuições não padrões, e se o modelo errado for escolhido, objetos de interesse podem ser classificados erroneamente.

Um classificador probabilístico OCL muito utilizado na literatura é o *Naïve Bayes Positive Class* (NBPC) (DATTA, 1997), que modifica o algoritmo *Naïve Bayes* tradicional, cujo objetivo é encontrar a probabilidade de um objeto não rotulado assumir uma classe com base no seu conjunto de atributos. Quando apenas dados de uma classe estão disponíveis, assumindo que os atributos (termos) do documento são independentes, aplica-se o teorema de Bayes, que calcula uma distribuição multinomial considerando a probabilidade dos termos do documento ocorrerem na classe de interesse. Logo, a função de pontuação para um novo documento é calculada conforme a Equação 3.2, na qual a probabilidade de um atributo t_i é dada pela Equação 3.3.

$$f(\mathbf{d}_i) = \frac{\prod_{t_j \in \mathscr{T}} p(t_j) * w_{d_i,t_j}}{\prod_{t_j \in \mathscr{T}} w_{d_i,t_j}}, \tag{3.2}$$

$$p(t_i) = \frac{1 + \sum_{d_j \in \mathscr{D}} w_{d_j,t_i}}{|\mathscr{T}| + \sum_{d_j \in \mathscr{D}} \sum_{t_k \in \mathscr{T}} w_{d_j,t_k}}. \tag{3.3}$$

Outro algoritmo estatístico aplicável a texto bem conhecido é o *One-Class Support Vector Machine* (OCSVM) (MANEVITZ; YOUSEF, 2001). Este algoritmo gera pontos fictícios próximos à origem e os considera como pontos anômalos. Então, um hiperplano de margem máxima é gerado, assim como em algoritmos de SVM. A função de otimização para obter o hiperplano de margem máxima é apresentada na Equação 3.4, na qual $\mathbf{h}$ corresponde aos coeficientes do hiperplano de separação, ε_{d_j} é o erro de classificação de um documento d_j, ρ é o limiar de erro de classificação, ν é um limite superior para a proporção de anomalias, $\Phi(\mathbf{d}_i)$ mapeia o espaço original para um espaço cujos exemplos de classes distintas sejam linearmente separáveis e a função $sgn(\cdot)$ retorna 1 caso seu valor seja ≥ 0 e 0 caso contrário. A pontuação atribuída para um novo documento d_i é apresentada na Equação 3.5.

$$\min \frac{1}{2} \parallel \mathbf{h} \parallel^2 + \frac{1}{\nu \cdot |\mathscr{D}|} \sum_{d_j \in \mathscr{D}} \varepsilon_{d_j} - \rho, \tag{3.4}$$

$$f(\mathbf{d}_i) = sgn(\mathbf{h} \cdot \Phi(\mathbf{d}_i) - \rho) \tag{3.5}$$

OCSVM foi utilizado em SONBHADRA; AGARWAL; NAGABHUSHAN (2020) para extração de tópicos de 45.000 artigos de Covid-19 contendo 75 categorias, com o intuito de auxiliar cientistas da comunidade a explorar técnicas de prevenção e tratamento. Algoritmos de agrupamento, como *k*-Means, foram usados para agrupar artigos considerando assuntos similares. Cada grupo foi treinado individualmente com OCSVM, que associou os artigos mais apropriados com a informação requerida em uma *string* de busca. A abordagem atingiu mais de 89% de F_1.

Embora métodos estatísticos sejam sólidos, possuem eficácia intimamente relacionada ao conhecimento do domínio e escolha do tipo de distribuição aplicada (considerando algoritmos paramétricos). No geral, apresentam desempenho reduzido quando os exemplos considerados são multidimensionais, como é o caso de dados textuais (TAN *et al.*, 2019).

3.2 Métodos Baseados em Distância e Densidade

Quando não se sabe qual é o modelo de distribuição seguido pelos dados, uma boa alternativa é a utilização de modelos baseados em distância. Modelos baseados em distância realizam o cálculo da distância de um novo documento em relação a um grupo de documentos (ou representantes de grupos de documentos), gerando uma pontuação.

Métodos que utilizam informações de densidade consideram anomalias como exemplos que estão em regiões pouco densas. Tais métodos são intimamente relacionados a métodos de distância, nos quais a densidade é definida em termos de proximidade. Densidade pode ser considerada como a reciprocidade da distância média entre k vizinhos mais próximos, de forma que se a distância é pequena para um conjunto, a densidade é alta. Duas variações do algoritmo k-NN que consideram informações de densidade são k-NND e k-NNRD (TAN *et al.*, 2019). O algoritmo *k-Nearest Neighbor Density* (k-NND) atribui uma pontuação a um documento por meio da distância média de seus k vizinhos mais próximos, conforme a Equação 3.6. Na equação, $\mathcal{N}_{(\mathbf{d}_i,k)}$ é o conjunto dos k vizinhos mais próximos de d_i, e $dist(\mathbf{d}_i,\mathbf{d}_j)$ retorna a distância entre os documentos $\mathbf{d}_i$ e $\mathbf{d}_j$.

$$f(d_i) = \text{densidade}(\mathbf{d}_i,k) = \left(\sum_{\mathbf{d}_j \in \mathcal{N}_{(\mathbf{d}_i,k)}} dist(\mathbf{d}_i,\mathbf{d}_j)/|\mathcal{N}_{(\mathbf{d}_i,k)}| \right)^{-1}, \tag{3.6}$$

Já o algoritmo *k-Nearest Neighbor Relative Density* (k-NNRD) atribui a densidade relativa de um objeto como a taxa da densidade de um documento dividida pela densidade média de seus vizinhos mais próximos, conforme a Equação 3.7.

$$densidade\ relativa\ média(\mathbf{d}_i,k) = \frac{densidade(\mathbf{d}_i,k)}{\sum_{\mathbf{d}_j \in \mathcal{N}_{(\mathbf{d}_i,k)}} densidade(\mathbf{d}_j,k)/|\mathcal{N}_{(\mathbf{d}_i,k)}|} \tag{3.7}$$

Detecção de anomalias baseadas em densidade possuem qualidade de desempenho relacionados com a escolha do valor k. Um algoritmo que pode ser utilizado para minimizar este problema é o *Local Outlier Factor* (LOF) (SCHUBERT; ZIMEK; KRIEGEL, 2014; BREUNIG *et al.*, 2000), que utiliza densidade e analisa uma variedade de valores de k, selecionando aqueles que atribuem pontuações máximas a anomalias (TAN *et al.*, 2019). Após o surgimento do algoritmo LOF, diversas variações também foram propostas (WANG; BAH; HAMMAD, 2019), tanto para exemplos de baixa quanto de alta dimensão.

3.3 Métodos Baseados em Agrupamento

Métodos de agrupamento são algoritmos não supervisionados que consideram exemplos anômalos como aqueles que não pertencem a nenhum grupo (região que contém grande quantidade de exemplos próximos). Para detectar exemplos anômalos, métodos baseados em densidade e em conectividade (TAN *et al.*, 2019) atuam para encontrar exemplos de baixa densidade ou fracamente conectados a outros exemplos.

Uma desvantagem de métodos de agrupamento para detecção de anomalias é que os resultados fornecidos podem não ser confiáveis, já que objetos anômalos podem afetar os grupos formados. Para minimizar tais problemas, há variações propostas que (TAN *et al.*, 2019): (i) realizam agrupamentos, removem os exemplos anômalos encontrados, e voltam a agrupar os objetos restantes até que nenhum exemplo anômalo permaneça; (ii) propõem a formação de um grupo especial de exemplos que não se encaixam bem a nenhum outro grupo, representando os potenciais exemplos anômalos.

Um algoritmo de agrupamento de uma única classe consolidado na literatura é uma variação do *spherical* k-Means. A pontuação atribuída pelo algoritmo k-Means a um novo documento d_i é dada por sua similaridade com o centroide do grupo mais próximo, conforme a Equação 3.8 (TAN *et al.*, 2019). Na equação, $\mathscr{D} = \mathscr{G}_1 \cup \mathscr{G}_2 \cup \cdots \cup \mathscr{G}_k$, $\mathscr{G}_i \subset \mathscr{D}$ é um grupo de textos e $\mathbf{g}_i$ é o centroide do grupo $\mathscr{G}_i$.

$$f(\mathbf{d}_i) = \max_{\mathscr{G}_j \subset \mathscr{D}} cos(\mathbf{d}_i, \mathbf{g}_j), \tag{3.8}$$

Técnicas de agrupamento também foram propostas na literatura para identificação de textos anômalos. Em LAZHAR (2019) foi usado o agrupamento *Fuzzy*, assumindo-se que documentos atribuídos a diferentes grupos com porcentagens muito próximas eram candidatos a serem anomalias. Após a remoção de documentos anômalos, classificadores como NB e SVM apresentaram singela melhoria de classificação.

Em GÔLO; MARCACINI; ROSSI (2020) foi realizada uma extensa avaliação de algoritmos de OCL na classficação de textos. Os três algoritmos que apresentaram maior efetividade diante das diversas bases de dados avaliadas foram o algoritmo baseado em *k-Means* (TAN *et al.*, 2019), *k-Nearest Neighbors Density-based* (k-NND) (TAN *et al.*, 2019) e *One-Class Support Vector Machine* (OCSVM) (MANEVITZ; YOUSEF, 2001), que serão considerados nesta tese para avaliação de desempenho da abordagem proposta. Os resultados também demonstraram que *tf-idf* foi o esquema de pesos que apresentou melhores resultados e a aplicação de técnicas de redução de dimensionalidade não necessariamente impacta o desempenho positivamente.

3.4 Métodos Baseados em Aprendizado Profundo

Modelos neurais treinados para filtrar documentos da classe de interesse também foram propostos na literatura. Em MANEVITZ; YOUSEF (2007) foi usado um *autoencoder* para classificação de textos, atingindo resultados superiores em relação aos algoritmos anteriormente citados, baseados em estatística, distância e densidade. O *autoencoder* é uma rede neural cuja meta é produzir uma saída que seja similar aos dados de entrada. Isto é, o objetivo é minimizar a função de regularização descrita na Equação 3.9.

$$J(\Theta) = \frac{1}{n} \sum_{\mathbf{d}_i \in \mathscr{D}} \left|\left| \mathbf{w}_{\mathbf{d}_i} - \mathbf{y}_{\mathbf{d}_i} \right|\right|_2^2, \tag{3.9}$$

Na Equação 3.9, $\mathbf{y}_{\mathbf{d}_i}$ é a informação de classe da saída da rede neural considerando como entrada o documento $\mathbf{d}_i$ e Θ são os parâmetros da rede neural. Dado um novo exemplo, se a similaridade entre a entrada e a saída ultrapassa um limiar, o documento é classificado como pertencente a classe de interesse. A abordagem apresentada em MANEVITZ; YOUSEF (2007) é uma rede neural de propagação (*feed-forward*), composta de m entradas, h neurônios na camada oculta e m saídas. Esta rede neural também é chamada de *Dense Autoencoder* (DAE), na qual todas as camadas são densas e todas as funções de ativação dos neurônios são *sigmoids*. Uma desvantagem do DAE é que o número de parâmetros a serem aprendidos pode ser muito alto.

Em RUFF *et al.* (2019) foi proposta uma abordagem de detecção de anomalias baseada em *embeddings* de palavras para aprendizado de representações de sentenças por meio de mecanismos de atenção, chamado *Context Vector Data Description* (CVDD). O método usou conjuntos de treinamento contendo aproximadamente 100 exemplos de cada classe da base de dados *Reuters*, na qual uma classe representa a classe de interesse e o restante é considerado como anomalia. Embora tenha atingido bons resultados, o método não atingiu ganho significativo em relação ao algoritmo OCSVM.

3.5 Métodos Baseados em Graph Neural Networks

Com o avanço de GNNs, em WANG *et al.* (2021) foram propostas abordagens baseadas em *One-class Graph Neural Networks* (OCGNN). Tais abordagens são supervisionadas e transdutivas, visando a classificação de nós estruturados em uma rede. Seguindo os paradigmas do aprendizado de uma única classe, o conjunto inicialmente rotulado é formado apenas por nós pertencentes a classe de interesse. Com base no paradigma de hiperesferas (TAX; DUIN, 2004), o objetivo das OCGNNs é aprender representações DE nós do grafo considerando a topologia da rede. Usando tais representações, nós de interesse encapsulados numa hiperesfera são considerados positivos, enquanto nós externos são considerados negativos. As GNNs consideram a representação estruturada de cada nó do grafo e uma matriz de adjacência $\mathbf{A}$ como entrada para o aprendizado de representações de alto nível. A representação inicial do documento

d_i é $\mathbf{d}_i \in \mathbf{D}$. Abordagens OCGNN usam $g(\mathbf{D},\mathbf{A};\mathscr{W})$ para representar uma GNN com pesos aprendíveis $\mathscr{W} = \{\mathbf{W}^{(1)},\cdots,\mathbf{W}^{(L)}\}$ em L camadas ocultas. Para a l-ésima camada, a propagação da GNN pode ser definida conforme a Equação 3.10.

$$\mathbf{H}^{l+1} = g(\mathbf{H}^{(l)},\mathbf{A};\mathbf{W}^{(l)}) \tag{3.10}$$

Na Equação 3.10, $\mathbf{H}^{(l)}$ é a entrada para a l-ésima camada ($\mathbf{H}^{l+1}$) é a camada de saída. As representações $\mathbf{D}$ são dados de entrada para a primeira camada, isto é, $\mathbf{H}^{(0)}$. As *embeddings* aprendidas para cada nó são denotadas por $\mathbf{H}^{(L)}$. Formalmente, a OCGCN minimiza a equação:

$$\mathscr{L}(r,\mathscr{W}) = \frac{1}{\nu\ |D|}\sum_{i=1}^{|D|}[\|g(\mathbf{D},\mathbf{A};\mathscr{W})_i - c\|^2 - r^2]^+ + r^2 + \frac{\lambda}{2}\sum_{l=0}^{L}\|\mathbf{W}^l\|^2. \tag{3.11}$$

Na Equação 3.11, r é o raio da hiperesfera aprendido com os pesos da rede neural $\mathscr{W}$, ν é o mesmo parâmetro de OCSVM proposto em MANEVITZ; YOUSEF (2001), c é o centro da hiperesfera definido pela média das *embeddings* do nó de interesse por uma propagação inicial, λ é o decaimento de peso da OCGNN e $[.]^+ = \max(0,\cdot)$ é um operador não negativo. Essa abordagem pode ser aplicada a diferentes tipos de GNNs, como *Graph Convolutional Networks* (KIPF; WELLING, 2016), *Graph Attention Networks* (VELICKOVIC *et al.*, 2017) e *GraphSAGE* (HAMILTON; YING; LESKOVEC, 2017), abordados na Subseção 2.2.4, modificando o termo de saída $g(\mathbf{D},\mathbf{A};\mathscr{W})$ na Equação 3.11. O método atingiu resultados considerados estados da arte em OCL, e, portanto, foi escolhido para avaliar a proposta desta tese.

3.6 Aprendizado Positivo e Não Rotulado

Algoritmos PUL aprendem modelos considerando um conjunto de documentos rotulados da classe de interesse e não rotulados para treinar um classificador (aprendizado semissupervisionado indutivo) ou classificar documentos conhecidos não rotulados (aprendizado semissupervisionado transdutivo) (JASKIE; SPANIAS, 2019). Assim, o conjunto de documentos de treinamento é dado por $\mathscr{D} = \mathscr{D}^+ \cup \mathscr{D}^U$, no qual $\mathscr{D}^+$ é o conjunto de documentos da classe de interesse, $\mathscr{D}^U$ é o conjunto de documentos não rotulados, e geralmente $|\mathscr{D}^U| \gg |\mathscr{D}^+|$.

O propósito de utilizar documentos não rotulados durante o aprendizado é melhorar o desempenho de classificação, assim como em algoritmos semissupervisionados multiclasses (ZHU; GOLDBERG, 2009). Como documentos não rotulados são fáceis de coletar, e devido à facilidade de se rotular poucos documentos de interesse, PUL tem ganhado atenção nos últimos anos (JASKIE; SPANIAS, 2019; ZHANG *et al.*, 2019; MA; ZHANG, 2017; LI *et al.*, 2014).

Apesar dos benefícios de PUL, ainda existem questões em aberto na literatura que podem ser exploradas (BEKKER; DAVIS, 2020), como: (i) assegurar que pressupostos e atribuições considerados dentro de PUL estejam alinhados a tarefas do mundo real; (ii) uma comparação

empírica de abordagens PUL, mostrando quais pressupostos são razoáveis para obter bons desempenhos na prática; (iii) *Benchmarks* do mundo real para testar abordagens PUL; e a (iv) exploração de aprendizado *Positive and Unlabeled* (PU) em domínios relacionais.

Abordagens mais comuns de algoritmos PUL realizam a fase de aprendizado seguindo duas etapas: na primeira etapa, um conjunto de documentos de não interesse, também chamados documentos negativos, é gerado por meio da seleção de exemplos que possuam características distintas em relação ao conjunto inicialmente rotulado. Da mesma forma, o conjunto D^+ pode ser incrementado com a seleção de documentos considerados confiáveis, ou seja, exemplos que possuam características similares a ele. Uma vez que há documentos positivos, negativos e não rotulados, um algoritmo transdutivo infere o rótulo dos documentos restantes na segunda etapa (JASKIE; SPANIAS, 2019), conforme apresentado na Subseção 3.6.1.

Um dos algoritmos PUL mais intuitivos usados para classificação de textos é o RC-SVM (LI; LIU, 2003). RC-SVM trata o conjunto de documentos não rotulados $\mathscr{D}^U$ como pertencentes a classe de não interesse. O conjunto de interesse $\mathscr{D}^+$ e os documentos não rotulados $\mathscr{D}^U$ são usados como dados de treinamento para a construção do classificador Rocchio, utilizado para classificar documentos no conjunto $\mathscr{D}^U$. Os documentos classificados como negativos são considerados como dados de não interesse, compondo o conjunto *RN*. Os passos do algoritmo Rocchio são descritos no Algoritmo 1.

Algoritmo 1 – *Rocchio Support Vector Machine (LI; LIU, 2003)*

Entrada:
$\mathscr{D}^+$, um conjunto de documentos positivos rotulados
$\mathscr{D}^U$, um conjunto de documentos não rotulados
Saída:
RN, um conjunto extraído de documentos negativos confiáveis

1: Atribua a classe negativa ao conjunto não rotulado $\mathscr{D}^U$ e a classe positiva ao conjunto positivo rotulado $\mathscr{D}^+$
2: Seja $\mathbf{c}^+ = \alpha \frac{1}{|\mathscr{D}^+|} \sum_{\mathbf{d}_i \in \mathscr{D}^+} \frac{\mathbf{d}_i}{||\mathbf{d}_i||} - \beta \frac{1}{\mathscr{D}^U} \sum_{\mathbf{d}_i \in \mathscr{D}^U} \frac{\mathbf{d}_i}{||\mathbf{d}_i||}$
3: Seja $\mathbf{c}^- = \alpha \frac{1}{|\mathscr{D}^U|} \sum_{\mathbf{d}_i \in \mathscr{D}^U} \frac{\mathbf{d}_i}{||\mathbf{d}_i||} - \beta \frac{1}{\mathscr{D}^+} \sum_{\mathbf{d}_i \in \mathscr{D}^+} \frac{\mathbf{d}_i}{||\mathbf{d}_i||}$
4: **para todo** documento $\mathbf{d}_i$ in $\mathscr{D}^U$ **faça**
5: **se** $sim(\mathbf{c}^+, \mathbf{d}_i) \leq sim(\mathbf{c}^-, \mathbf{d}_i)$ **então**
6: $RN \leftarrow RN \cup \{\mathbf{d}_i\}$
7: **fim se**
8: **fim para**
9: **retorna** *RN*

Um classificador é construído usando vetores de protótipos positivos e negativos $\mathbf{c}^+$ and $\mathbf{c}^-$. Os parâmetros α e β ajustam o impacto relativo dos dados de treinamento positivos e negativos (LI; LIU, 2003) e $sim(.)$ corresponde a similaridade entre o documento e o protótipo.

A segunda etapa de RC-SVM consiste na construção do classificador final usando SVM iterativamente com os conjuntos D^+ e *RN*, descrito no Algoritmo 2. Um classificador SVM é

usado a cada iteração para extrair documentos negativos de interesse de Q, $Q = \mathscr{D}^U - RN$. As iterações terminam quando não há mais documentos negativos a serem extraídos. Se muitos documentos positivos são incluídos no conjunto RN, o classificador final terá desempenho extremamente inferior. Logo, o primeiro classificador S_1 será escolhido (LI; LIU, 2003).

Algoritmo 2 – *Construção do Classificador SVM*

Entrada:
$\mathscr{D}^+$, um conjunto de documentos positivos rotulados
$\mathscr{D}^U$, um conjunto de documentos não rotulados
RN, um conjunto de documentos negativos confiáveis extraído

A todo documento em $\mathscr{D}^+$ é atribuído o o rótulo *positivo*;
A todo documento em RN é atribuído o o rótulo *negativo*;
$Q = \mathscr{D}^U - RN$;
para $i \leftarrow 1$ inicialmente e $i \leftarrow i + 1$ **faça**
    Use $\mathscr{D}^+$ e RN para treinar um classificador SVM S_i;
    Classifique Q usando S_i. Seja W os documentos em Q classificados como negativos;
    se $W = \{\}$ **então** pare;
    senão $Q \leftarrow Q - W$
        $RN \leftarrow RN \cup W$
    fim se
fim para
Use o último classificador SVM S_{ultimo} para classificar D^+;
se mais de 5% de documentos positivos forem classificados como negativos **então**
    use S_1 como o classificador final;
senão use S_{ultimo} como classificador final;
fim se

Processos similares são feitos por outros algoritmos PUL para classificar textos, como *Spy Expectation-Maximization* (S-EM) (JASKIE; SPANIAS, 2019). No algoritmo S-EM (LIU *et al.*, 2002), alguns documentos da classe de interesse (espiões) são adicionados ao conjunto de documentos não rotulados. Um modelo de classificação baseado em NB é construído e as probabilidades atribuídas aos documentos espiões são usadas para atribuir limiares a fim de inferir documentos negativos confiáveis. Logo, o algoritmo EM é usado para classificar os documentos não rotulados restantes.

Rocchio também foi usado para extração de revisões positivas e potencialmente negativas de exemplos não rotulados em XU *et al.* (2019). O algoritmo PUL usa treinamento adversário e Long Short-term Memory (LSTM) para análise de sentimentos, adicionando uma perturbação aleatória às representações de documentos para minimizar o risco de rotulação incorreta. Foram utilizados de 20 a 40% de revisões positivas no conjunto D^+. O algoritmo atingiu de 65% a 83% de macro F_1 considerando duas bases de dados.

Os algoritmos PUL apresentados anteriormente são baseados em representações espaço-vetoriais, como o treinamento realizado em RC-SVM ou o *Expectation Maximization* aplicado em S-EM. S-EM apenas apresenta bom desempenho quando possui um número pequeno de

exemplos positivos no conjunto não rotulado. Além disso, NB assume que todos os exemplos não rotulados são anômalos, já que ele tolera pouco ruído no conjunto de treinamento. Em XU *et al.* (2019) a proposta não é comparada a outros algoritmos PUL. Além disso, o sucesso do algoritmo depende altamente da escolha das propriedades da perturbação aleatória.

Para minimizar estas limitações de abordagens baseadas no modelo espaço-vetorial, representações baseadas em redes têm demonstrado serem boas alternativas (YANG *et al.*, 2020; SHI; PHILIP, 2017; ROSSI; LOPES; REZENDE, 2016; ROSSI; REZENDE; LOPES, 2015). Mesmo assim, há poucos algoritmos PUL baseados em redes, como *Positive documents Enlarging PU Classifier* (PE-PUC) (YU; LI, 2007) e *Positive and Unlabeled Learning by Label Propagation* (PU-LP) (MA; ZHANG, 2017).

PE-PUC usa Naïve Bayes para extração de exemplos negativos confiáveis. Exemplos não rotulados são todos considerados anômalos, enquanto o conjunto inicialmente rotulado representa a classe de interesse. O modelo aprendido por NB classifica exemplos não rotulados e aqueles classificados como negativos formam um conjunto de negativos confiáveis. PE-PUC usa a representação da rede contendo nós e arestas apenas para aumentar o conjunto de documentos de interesse com documentos positivos confiáveis. No entanto, a rede não é usada na etapa de classificação. Além disso, se o conjunto de documentos de interesse inicialmente rotulado for muito pequeno, o conjunto não será suficiente para representar a distribuição da classe de interesse. Assim, a maioria dos dados não rotulados será classificada como documentos negativos confiáveis, o que limita o desempenho do algoritmo. Por outro lado, PU-LP é um algoritmo semissupervisionado, totalmente baseado em redes, que apresentou resultados satisfatórios em dados numéricos e pode ser explorado e adaptado para dados textuais (MA; ZHANG, 2017).

PU-LP segue uma estratégia de dois passos: (i) primeiramente, constrói-se uma rede de nós e arestas conforme a similaridade de exemplos pertencentes a uma base de dados. Considerando a rede e uma medida de similaridade baseada em caminhos (índice de Katz), criam-se dois novos conjuntos de exemplos: o conjunto de interesse confiável e conjunto de não interesse confiável; e (ii) utilizam-se algoritmos de propagação de rótulos para rotular os demais nós da rede. As definições fornecidas a seguir consideram objetos da rede como documentos de texto, conforme o Algoritmo 3.

Inicialmente, PU-LP constrói uma matriz de adjacência com o conjunto completo de exemplos $\mathscr{D}$. Nesta matriz, documentos com conteúdos similares possuem uma baixa distância entre si. A matriz de adjacência é usada como base para construção de uma matriz k-NN, chamada $\mathbf{A}$, de forma que $\mathbf{A}_{i,j} = 1$ se o documento $\mathbf{d}_j$ é um dos k vizinhos mais próximas do documento $\mathbf{d}_i$, e $\mathbf{A}_{i,j} = 0$ caso contrário. Por meio do algoritmo k-NN, uma rede também é criada, na qual vértices representam documentos e arestas conectam vértices similares.

Algumas considerações podem ser feitas a partir da rede k-NN (LÜ; JIN; ZHOU, 2009; KATZ, 1953): se dois documentos estão diretamente conectados, eles em geral serão considerados da mesma classe. Além disso, rótulos podem se propagar ao longo de caminhos no grafo, de forma

Algoritmo 3 – *Positive and Unlabeled Learning by Label Propagation*

Entrada:
$\mathscr{D}^+$, um conjunto de documentos rotulados positivos (de interesse)
$\mathscr{D}^U$, um conjunto de documentos não rotulados
m, número de iterações
λ, controla o tamanho do conjunto RI, $\lambda \in (0,1)$
W, matriz de similaridade, calculada com índice de Katz
Saída:
RI, um conjunto de documentos positivos confiáveis extraídos
RN, um conjunto de documentos negativos confiáveis extraídos

1: $RI \leftarrow \emptyset, RN \leftarrow \emptyset$
2: **para** $k \leftarrow 1 : m$ **faça**
3: Com base em W, calcule $S_{\mathbf{d}_i}, \mathbf{d}_i \in \mathscr{D}^U, S_{\mathbf{d}_i} = \frac{\Sigma_{j=1}^{|\mathscr{D}^+|} W_{i,j}}{|\mathscr{D}^+|}$, $S_{\mathbf{d}_i}$ é a média da similaridade entre um documento não rotulado $\mathbf{d}_i$ e todos os documentos positivos rotulados;
4: ranquear cada documento $\mathbf{d}_i$ de acordo com $S_{\mathbf{d}_i}$, $\mathbf{d}_i \in \mathscr{D}^U$;
5: $RI' \leftarrow$ os primeiros $\frac{\lambda}{m} \times |\mathscr{D}^+|$ exemplos ranqueados em $\mathscr{D}^U$;
6: $RI \leftarrow RI \cup RI', \mathscr{D}^U \leftarrow \mathscr{D}^U - RI'$
7: **fim para**
8: Com base em W, calcule $S_{\mathbf{d}_i}, \mathbf{d}_i \in \mathscr{D}^U - RI, S_{\mathbf{d}_i} = \frac{\Sigma_{j=1}^{|\mathscr{D}^+ \cup RI|} W_{i,j}}{|\mathscr{D}^+ \cup RI|}$;
9: ranquear cada documento $\mathbf{d}_i$ de acordo com $S_{\mathbf{d}_i}$, $\mathbf{d}_i \in \mathscr{D}^U - RI$;
10: $RN \leftarrow$ os últimos $|\mathscr{D}^+ \cup RI|$ exemplos ranqueados em $\mathscr{D}^U - RI$;
11: **retorna** RI, RN

que se dois vértices possuam muitos vizinhos em comum, provavelmente serão da mesma classe. Logo, em MA; ZHANG (2017) é proposta a utilização do índice de Katz (LÜ; JIN; ZHOU, 2009; KATZ, 1953), uma medida global que calcula a similaridade entre pares de vértices considerando todos os caminhos possíveis da rede que os conectam, conforme a Equação 3.12, na qual α é um parâmetro livre que controla a influência dos caminhos, de forma que caminhos mais longos possuam menor contribuição no cálculo (MA; ZHANG, 2017).

$$sim(d_i, d_j) = \sum_{h=1}^{\infty} \alpha^h \cdot |path_{\mathbf{d}_i,\mathbf{d}_j}^{<h>}| = \alpha \mathbf{A}_{i,j} + \alpha^2 (\mathbf{A}^2)_{i,j} + \alpha^3 (\mathbf{A}^3)_{i,j} + \ldots \tag{3.12}$$

Para o cálculo do índice de Katz, quando $\alpha < 1/\varepsilon$, sendo ε o maior autovalor para a matriz $\mathbf{A}$, a Equação 3.12 converge e pode ser calculada conforme a Equação 3.13, na qual $\mathbf{I}$ denota a matriz identidade e $\mathbf{W} = (|\mathscr{D}| \times |\mathscr{D}|)$. Assim, $\mathbf{W}_{i,j} \in \mathbb{R}$ denota a similaridade entre os nós d_i e d_j conforme o índice de Katz.

$$\mathbf{W} = (\mathbf{I} - \alpha \mathbf{A})^{-1} - \mathbf{I}, \tag{3.13}$$

A partir da matriz de similaridade $\mathbf{W}$, obtêm-se um conjunto de exemplos de interesse confiáveis RI, composto de exemplos do conjunto $\mathscr{D}^U$ que sejam mais similares aos exemplos do conjunto $\mathscr{D}^+$. Para isso, um método iterativo é aplicado: a tarefa de extrair o conjunto RI é

dividida em m passos. Como uma pequena quantidade de exemplos rotulados é provido, é difícil extrair muitos exemplos confiáveis de uma única vez. O número total de exemplos positivos rotulados que será extraído é $(\lambda/m) \times |\mathscr{D}^+|$, na qual λ controla o tamanho do conjunto. Em cada passo iterativo, exemplos em $\mathscr{D}^U$ são ordenados conforme a média de sua similaridade com todos os exemplos em $\mathscr{D}^+$, com base na matriz W. Os $(\lambda/m)|\mathscr{D}^+|$ exemplos mais similares são retirados do conjunto $\mathscr{D}^U$, formando o conjunto RI'. Após o fim de cada um dos m passos, o conjunto de exemplos de interesse ($\mathscr{D}^+$) e o conjunto de exemplos de interesse confiáveis (RI) são incrementados com os elementos do conjunto RI'. Assim, $\mathscr{D}^+ \leftarrow (\mathscr{D}^+ \cup RI')$ e $RP \leftarrow (RP \cup RI')$ (MA; ZHANG, 2017).

Após aumentar $\mathscr{D}^+$ iterativamente, exemplos negativos confiáveis, que possuam maior dissimilaridade com o conjunto $\mathscr{D}^+ \cup RI$, são extraídos do conjunto de exemplos não rotulados ($\mathscr{D}^U$ - RI). Com base em W, exemplos do conjunto não rotulado são ordenados conforme a média de sua similaridade com todos os exemplos em $\mathscr{D}^+ \cup RI$. PU-LP extrai os $|\mathscr{D}^+ \cup RI|$ exemplos mais dissimilares, formando o conjunto RN. Após aumentar $\mathscr{D}^+$ e obter o conjunto RN, os conjuntos $\mathscr{D}^+ \cup RI$, RN e $\mathscr{D}^U \leftarrow (\mathscr{D}^U$ - RI - $RN)$ são usados como entrada por algoritmos semissupervisionados binários. Dessa forma, a partir dos vértices rotulados, fixos, algoritmos semissupervisionados rotulam gradualmente o conjunto de vértices não rotulados.

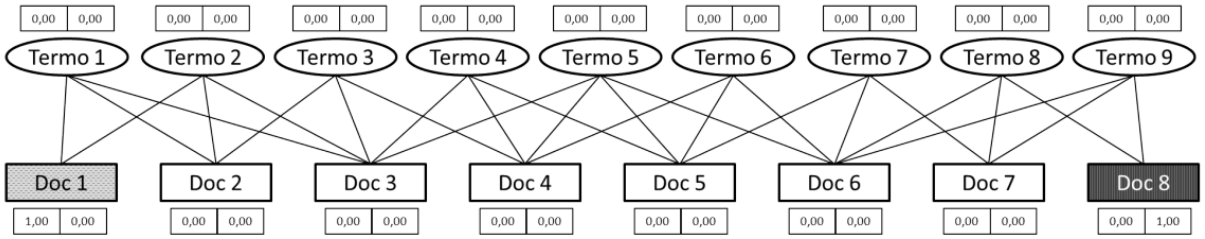
PU-LP atingiu resultados satisfatórios na literatura com um conjunto inicialmente rotulado contendo 10% de exemplos da classe de interesse, o que pode ser relevante no contexto de detecção de notícias falsas. Por ser totalmente baseado em redes, o algoritmo permite a inclusão de novas características na rede, que possam acrescentar padrões relevantes para discriminação de conteúdo, auxiliando algoritmos de propagação de rótulos na classificação. A seguir, são apresentados algoritmos clássicos da literatura que realizam a tarefa de propagar rótulos.

3.6.1 Algoritmos de Propagação de Rótulos

Estruturas de redes e um conjunto de nós rotulados podem ser utilizados como entrada em algoritmos de classificação semissupervisionados, que realizam a atribuição de rótulos a cada elemento da rede. Tais algoritmos utilizam um vetor de classes que armazena valores de pertinência de um elemento da rede a um conjunto de classes possíveis. Tais algoritmos podem ser **indutivos**, nos quais dada uma coleção de textos rotulados, um modelo é induzido para a classificação posterior de novos documentos ainda não vistos; e **transdutivos**, que classificam objetos não rotulados sem a necessidade de induzir um modelo. Em algoritmos transdutivos, documentos rotulados e não rotulados são observados e suas características são exploradas, fazendo uso das informações para classificar todos os objetos da rede, como representado na Figura 7.

Na Figura 7, nós retangulares representam documentos da coleção, relacionados em uma rede bipartida com nós circulares, que representam termos presentes em cada documento. Cada elemento da rede possui um vetor associado contendo duas dimensões. A primeira dimensão

Figura 7 – Rede heterogênea contendo relações entre documentos e termos de uma coleção. Arestas representam pesos correspondentes à frequência de um termo em um documento. Figura retirada de ROSSI (2016).



representa quanto o elemento pertence à classe c_1. Da mesma forma, a segunda dimensão representa quanto o elemento pertence à classe c_2. Os documentos 1 e 8 são rotulados, de forma que o primeiro é da classe c_1 (preenchido com o valor 1 na primeira dimensão do vetor) e o segundo da classe c_2. Após algumas iterações do algoritmo de propagação de rótulos, espera-se que todos os elementos não rotulados sejam atribuídos a uma classe, considerando as relações existentes na rede.

Seja $\mathcal{N} = \langle \mathcal{O}, \mathcal{R}, \mathcal{W} \rangle$ uma rede, na qual $\mathcal{O}$ indica um conjunto de objetos, $\mathcal{R}$ representa o conjunto de relações entre objetos e $\mathcal{W}$ representa os pesos destas relações. Dado dois pares de objetos $o_i, o_j \in \mathcal{O}$, a relação entre eles é representada por r_{o_i,o_j}. O peso de uma relação r_{o_i,o_j} é dado por $w_{o_i,o_j} \forall o_i, o_j \in \mathcal{O}$. As relações existentes entre os objetos de uma rede podem possuir pesos iguais, isto é, $o_i, o_j \in \mathcal{O}$, $w_{o_i,o_j} = 1$ se $\exists r_{o_i,o_j} \in \mathbb{R}$, ou pesos distintos (redes ponderadas), nas quais se $\exists r_{o_i,o_j} \in \mathbb{R}$, pesos w_{o_i,o_j} podem ser equivalentes a qualquer valor real. Em geral, valores positivos são atribuídos como pesos de arestas (ROSSI, 2016).

Em algoritmos de classificação transdutivos (GAMMERMAN; VOVK; VAPNIK, 1998), a classificação dos elementos não rotulados pode ocorrer por meio de regularização. Algoritmos de regularização buscam minimizar uma função de custo que satisfaça duas premissas (ROSSI, 2016): (i) as informações de classe de objetos vizinhos devem ser semelhantes; e (ii) as informações de classe dos objetos rotulados atribuídas durante o processo de classificação devem ser semelhantes às informações de classe reais. Tais premissas podem ser expressas em um *framework* de regularização, no qual o primeiro termo é a função de regularização e o segundo a função de custo (DELALLEAU; BENGIO; ROUX, 2005):

$$Q(\mathbf{F}) = \frac{1}{2} \sum_{o_i, o_j \in \mathcal{O}} w_{o_i,o_j} \Omega(\mathbf{f}_{o_i}, \mathbf{f}_{o_j}) + \mu \sum_{o_i \in \mathcal{O}^L} \Omega'(\mathbf{f}_{o_i}, \mathbf{y}_{o_i}). \tag{3.14}$$

Na Equação 3.14, o vetor $\mathbf{f}_{o_i} = \{f_1, f_2, \ldots, f_{|\mathscr{C}|}\}$ armazena o peso de um objeto o_i para cada classe da coleção de objetos, em que $\mathscr{C} = \{c_1, c_2, \ldots, c_l\}$ é o conjunto de classes da coleção. O peso será usado posteriormente para atribuir o objeto a uma classe. O termo $\mathcal{O}^L$ se refere ao conjunto de objetos rotulados. O termo "informação de classe" também pode ser utilizado para denotar o vetor de pesos dos objetos para as classes. O vetor $\mathbf{y}_{o_i}$ possui as mesmas dimensões

do vetor $\mathbf{f}$, porém armazena o rótulo original do objeto o_i, de forma que a posição do vetor correspondente ao seu rótulo seja preenchida com o valor 1. Além disso, μ é o parâmetro de regularização que define a importância que será dada a cada uma das premissas, e $\Omega(\ldots)$ e $\Omega'(\ldots)$ são funções de distância.

A função $\Omega(\ldots)$ calcula a distância entre os vetores de informação de classe e cada par de objetos relacionados na rede. A função $\Omega'(\ldots)$ calcula a proximidade entre a informação de classe de objetos rotulados e suas informações de classe reais. Tal equação pode ser resolvida por meio de soluções iterativas, chamadas "propagação de rótulos", na qual os objetos propagam sua informação de classe para objetos vizinhos de maneira gradual. A propagação é feita até que não haja mais mudanças nas informações de classe dos nós da rede, ou definindo-se um número máximo de iterações para o algoritmo.

3.6.2 Propagação de Rótulos em Redes Homogêneas

Dois algoritmos iterativos para propagação de rótulos que apresentam bons desempenhos de classificação em redes homogêneas (ROSSI, 2016) são *Gaussian Fields and Harmonic Functions* (GFHF) (ZHU; GHAHRAMANI; LAFFERTY, 2003) e *Learning With Local and Global Consistency* (LLGC) (ZHOU *et al.*, 2004). O algoritmo GFHF utiliza uma função harmônica que determina a informação de classe de um objeto com base na média das informações de classe de objetos vizinhos, ponderada pelos pesos das conexões:

$$\mathbf{f}_{o_i} = \frac{\sum_{o_j \in \mathcal{O}} w_{o_i,o_j} \mathbf{f}_{o_j}}{\sum_{o_j \in \mathcal{O}} w_{o_i,o_j}}. \tag{3.15}$$

Esta função é aplicada somente a objetos não rotulados. A função de regularização a ser minimizada é a seguinte:

$$Q(\mathbf{F}) = \frac{1}{2} \sum_{o_i,o_j \in \mathcal{O}} w_{o_i,o_j} (\mathbf{f}_{o_i} - \mathbf{f}_{o_j})^2 + \lim_{\mu \to \infty} \sum_{o_i \in \mathcal{O}^L} (\mathbf{f}_{o_i} - \mathbf{y}_{o_i})^2. \tag{3.16}$$

GFHF não permite que a informação de classe de objetos previamente rotulados sejam alteradas durante o processo de propagação de rótulos. Esta premissa é garantida pelo valor $\lim_{\mu \to \infty}$.

Ao contrário do primeiro algoritmo, LLGC permite que a informação de classe de objetos vizinhos mude durante o processo de classificação, uma vez que objetos podem ser erroneamente rotulados, deteriorando o desempenho de classificação. Além disso, no processo de propagação de rótulos, para calcular a informação de classe de um objeto, o algoritmo considera tanto informações relacionadas ao grau do objeto de origem quanto ao grau do objeto de destino. Essa característica é expressa na normalização realizada no primeiro termo do regularizador:

$$Q(\mathbf{F}) = \frac{1}{2} \sum_{\mathcal{O}_i,\mathcal{O}_j \in \mathcal{O}} w_{o_i,o_j} \left\| \frac{\mathbf{f}_{o_i}}{\sqrt{\sum_{o_k \in \mathcal{O}} w_{o_i,o_k}}} - \frac{\mathbf{f}_{o_j}}{\sqrt{\sum_{o_k \in \mathcal{O}} w_{o_j,o_k}}} \right\|^2 + \mu \sum_{\mathcal{O}_i \in \mathcal{O}^L} ||\mathbf{f}_{o_i} - \mathbf{y}_{o_i}||^2 \tag{3.17}$$

O parâmetro μ presente no segundo termo na função de regularização define o grau da importância de objetos inicialmente rotulados. O algoritmo LLGC pode ser resolvido pela aplicação da Equação 3.18 de forma iterativa até a convergência dos valores em F:

$$\mathbf{F} = \alpha\mathbf{SF} + (1-\alpha)\mathbf{Y} \tag{3.18}$$

na qual

$$\mathbf{S} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{1/2} \tag{3.19}$$

Na Equação 3.19, $\mathbf{W}$ é a matriz de pesos, $\mathbf{Y}$ é a informação de classe real dos objetos rotulados e $\mathbf{D}$ é uma matriz diagonal, $|\mathscr{O}| \times |\mathscr{O}|$, na qual cada célula da diagonal contém o grau do nó, equivalente a $d_{o_i,o_j} = \sum_{o_j \in O_{w_{o_i,o_j}}}$. O parâmetro α em geral possui valor $0 < \alpha < 1$.

3.6.3 Propagação de Rótulos em Redes Heterogêneas

Quando o conjunto $\mathscr{O}$ é composto por h diferentes tipos de objetos, isto é, $\mathscr{O} = \mathscr{O}_1 \cup \mathscr{O}_2 \cup \ldots \cup \mathscr{O}_h$, com $h \geq 2$, a rede é denominada heterogênea. Para lidar com este tipo de rede, extensões dos algoritmos GFHF e LLGC foram propostas na literatura, correspondendo aos algoritmos *Label Propagation through Heterogeneous Network* (LPHN) (ROSSI, 2016) e GNetMine (JI *et al.*, 2010), que realizam a propagação de rótulos considerando os diversos tipos de elementos e relações existentes na rede.

O algoritmo LPHN é uma extensão do algoritmo GFHF (abordado na Subseção 3.6.2) para classificação transdutiva em redes heterogêneas. Neste algoritmo, objetos previamente rotulados não devem ter seus rótulos alterados ao final da etapa de classificação. A função de regularização do LHPN é análoga à função do algoritmo GFHF (Equação 3.16), porém considera tipos de relações distintas no primeiro termo do regularizador. O algoritmo seleciona dois conjuntos de objetos propagando os rótulos de um conjunto $\mathscr{O}_i$ para um conjunto $\mathscr{O}_j$, conforme a Equação 3.20. A equação pode ainda ser resolvida por meio de métodos iterativos, como propagação de rótulos, assim como no Algoritmo 4.

$$Q(\mathbf{F}) = \sum_{\mathscr{O}_i,\mathscr{O}_j \in \mathscr{O}} \frac{1}{2} \sum_{o_i \in \mathscr{O}_i} \sum_{o_j \in \mathscr{O}_j} w_{o_i,o_j} (\mathbf{f}_{o_i} - \mathbf{f}_{o_j})^2 + \lim_{\mu \to \infty} \sum_{o_i \in \mathscr{O}^L} (\mathbf{f}_{o_i} - \mathbf{y}_{o_i})^2. \tag{3.20}$$

Já o algoritmo GNetMine (JI *et al.*, 2010) é uma extensão do algoritmo LLGC (Subseção 3.6.2). Nele, além de relações entre objetos distintos apresentarem diferentes importâncias, o algoritmo permite que a confiança dos rótulos dos objetos de treinamento seja reduzida. Desta forma, o rótulo de um objeto pode ser alterado durante o processo de classificação, caso informações de objetos vizinhos apontem divergência em relação à classe do objeto inicialmente rotulado. Na Equação 3.21 é descrita a função de regularização do GNetMine. O termo λ define a importância da relação entre objetos do tipo $\mathscr{O}_i$ e $\mathscr{O}_j$, o qual varia entre 0 e 1. O termo $\alpha_{o_j} \in \mathscr{O}^L$

Algoritmo 4 – *Label Propagation through Heterogeneus Networks*

Entrada:
$\mathscr{O}$, conjunto de objetos da rede
Y, informações de classes reais
W, pesos de conexões entre objetos
D, grau dos objetos
Saída:
$\mathbf{F}(\mathscr{O}^U)$, informação de casse dos objetos não rotulados

1: $\mathbf{P} \leftarrow (\mathbf{D}^{-1}) \cdot \mathbf{W}$;
2: **enquanto** não houver convergência ou número máximo de iterações não ser atingido **faça**
3: **para todo** $\mathscr{O}_i, \mathscr{O}_j \subset \mathscr{O}$ (subconjunto de objetos do tipo i e j) **faça**
4: $\mathbf{F}(\mathscr{O}_i, \mathscr{O}_j) \leftarrow \mathbf{P}(\mathscr{O}_i, \mathscr{O}_j) \cdot \mathbf{F}(\mathscr{O}_i, \mathscr{O}_j)$;
5: $\mathbf{F}(\mathscr{O}_i^L) \leftarrow \mathbf{Y}(\mathscr{O}_i^L)$;
6: $\mathbf{F}(\mathscr{O}_j^L) \leftarrow \mathbf{Y}(\mathscr{O}_j^L)$;
7: **fim para**
8: **fim enquanto**
9: **retorna** $\mathbf{F}(\mathscr{O}^U)$

define a importância do objeto inicialmente rotulado o_j, também variando entre 0 e 1.

$$Q(\mathbf{F}) = \sum_{\mathscr{O}_i, \mathscr{O}_j \subset \mathscr{O}} \lambda_{\mathscr{O}_i, \mathscr{O}_j} \sum_{o_k \in \mathscr{O}_i} \sum_{o_l \in \mathscr{O} j} w_{o_k, o_l} \left\| \frac{\mathbf{f}_{o_k}(\mathscr{O}_i)}{\sqrt{\sum_{o_m \in \mathscr{O}_j} w_{o_k, o_m}}} - \frac{\mathbf{f}_{o_l}(\mathscr{O}_j)}{\sqrt{\sum_{o_m \in \mathscr{O}_i} w_{o_l, o_m}}} \right\|^2 + \sum_{o_j \in \mathscr{O}^L} \alpha_{o_j} (\mathbf{f}_{o_j} - \mathbf{y}_{o_j}) \quad (3.21)$$

A Equação 3.21 pode ser resolvida por meio de métodos iterativos, como propagação de rótulos, assim como no Algoritmo 5. Na linha 2, um valor de confiança da conexão entre dois tipos de objetos é definido. Nas linhas 7-9 é realizada a propagação de rótulos, na qual informações de classe de objetos são atualizadas com base em informações de objetos vizinhos, bem como com base na importância de cada relação e a confiança da informação rotulada.

Representações baseadas em redes unidas a algoritmos de propagação de rótulos foram pouco explorados na literatura para classificar notícias (GUACHO *et al.*, 2018), e serão utilizados nesta tese por serem modelos interpretáveis e relacionais.

3.7 Considerações Finais

Neste capítulo foram abordados os principais aspectos relacionados a OCL, bem como questões em aberto na literatura e trabalhos recentemente propostos para classificação de textos. Na Tabela 2 é sumarizada a limitação dos algoritmos discutidos considerando o problema de detecção de notícias falsas. É possível observar que (i) há necessidade de abordagens PUL que

Algoritmo 5 – *GNetMine*

Entrada:
$\mathscr{O}$, conjunto de objetos da rede
Y, informações de classes reais
W, pesos de conexões entre objetos
D, grau dos objetos
α, confiança dos objetos rotulados
$\lambda[]$, importâncias das relações entre objetos distintos da rede
Saída:
$\mathbf{F}(\mathscr{O}^U)$, informação de casse dos objetos não rotulados

1: **para todo** $\mathscr{O}_i, \mathscr{O}_j \subset \mathscr{O}$ (subconjunto de objetos do tipo i e j) **faça**
2: $\quad \mathbf{S}(\mathscr{O}_i, \mathscr{O}_j) \leftarrow \mathbf{D}(\mathscr{O}_i, \mathscr{O}_j)^{-\frac{1}{2}} \cdot \mathbf{W}(\mathscr{O}_i, \mathscr{O}_j) \cdot \mathbf{D}(\mathscr{O}_i, \mathscr{O}_j)^{-\frac{1}{2}}$;
3: **fim para**
4: **enquanto** não houver convergência ou número máximo de iterações não ser atingido **faça**
5: $\quad$ **para todo** $\mathscr{O}_i \subset \mathscr{O}$ **faça**
6: $\quad\quad$ **para todo** $o_j \subset \mathscr{O}_i$ **faça**
7: $\quad\quad\quad \mathbf{f}_{o_j} \leftarrow \sum_{\mathscr{O}_k \subset \mathscr{O}, o_j \notin \mathscr{O}_k} \lambda_{\mathscr{O}_i, \mathscr{O}_k} \cdot \sum_{o_m \in \mathscr{O}_k} \cdot \mathbf{S}(\mathscr{O}_i, \mathscr{O}_k)_{o_j, o_m} \cdot \mathbf{f}_{o_m}$;
8: $\quad\quad\quad \mathbf{f}_{o_j} \leftarrow \mathbf{f}_{o_j} + \left(2 \cdot \lambda_{\mathscr{O}_i, \mathscr{O}_i} \cdot \sum_{o_m \in \mathscr{O}_i} \mathbf{S}(\mathscr{O}_i, \mathscr{O}_i) o_j, o_m \cdot \mathbf{f}_{o_m}\right) + \alpha_{o_j} \cdot \mathbf{y}_{o_j}$;
9: $\quad\quad\quad \mathbf{f}o_j \leftarrow \frac{\mathbf{f}o_j}{\sum_{\mathscr{O}_k \subset \mathscr{O}, o_j \notin \mathscr{O}_k} \lambda_{\mathscr{O}_i, \mathscr{O}_k} + 2 \cdot \lambda_{\mathscr{O}_i, \mathscr{O}_i} + \alpha_{o_j}}$;
10: $\quad\quad$ **fim para**
11: $\quad$ **fim para**
12: **fim enquanto**
13: **retorna** $\mathbf{F}(\mathscr{O}^U)$

comprovem desempenho em várias bases de dados; (ii) necessidade de examinar características para a seleção das mais apropriadas na detecção de anomalias, realizando diferentes avaliações experimentais; (iii) a maioria dos algoritmos demanda a calibração de limiares para classificar exemplos, o que é difícil em cenários do mundo real. Além disso (BEKKER; DAVIS, 2020; JASKIE; SPANIAS, 2019), (iv) algoritmos PUL não desempenham bem quando o conjunto não rotulado possui um número grande de exemplos da classe de interesse, ou quando o conjunto de interesse rotulado tem um número limitado de dados rotulados; e por fim, (v) há falta de algoritmos relacionais propostos para PUL que sejam adequados no contexto de classificação de notícias, representadas como dados multidimensionais.

O aprendizado semissupervisionado baseado em redes tem demonstrado efetividade no uso de dados não rotulados multidimensionais na melhoria de desempenho de classificação, permitindo a modelagem de diferentes objetos e relações com rica semântica. Além disso, o aprendizado semissupervisionado atinge desempenho de classificação satisfatório com poucos dados rotulados, e não requer a calibração de limiares para classificar exemplos como anômalos (SHI; PHILIP, 2017; ROSSI; LOPES; REZENDE, 2016). Portanto, nesta tese é proposta uma abordagem baseada no algoritmo PU-LP aplicada a detecção de notícias falsas, descrita no Capítulo 4.

Tabela 2 – Limitações de abordagens OCL e PUL para detecção de notícias falsas.

Citação	**Tipo**	**Modelo**	**Língua da base de dados**	**Limitações da Abordagem**
DCDistanceOCC (FAUSTINI; COVÕES, 2019)	OCL	espaço-vetorial	Português	Calibração de limiar.
k-Means (TAN *et al.*, 2019; GÔLO; MARCACINI; ROSSI, 2020)	OCL	espaço-vetorial	Inglês	Resultados são fortemente dependentes do número de grupos, do formato dos grupos e dos limiares usados para atribuir um exemplo a um grupo.
k-NND (TAN *et al.*, 2019; GÔLO; MARCACINI; ROSSI, 2020)	OCL	espaço-vetorial	Inglês	Definição do valor ótimo de k e calibração de limiar.
OCSVM (MANEVITZ; YOUSEF, 2001)	OCL	espaço-vetorial	Inglês	Resultados provaram ser muito sensitivos aos parâmetros, especialmente ao kernel escolhido e a dimensão do modelo de representação.
Dense Autoencoder (MANEVITZ; YOUSEF, 2007)	OCL	espaço-vetorial	Inglês	Definição de um limiar adequado para a classe de interesse, número de parâmetros a serem treinados e o desempenho do algoritmo depende que os documentos não rotulados da classe de interesse sejam muito similares ao conjunto inicialmente rotulado.
Context Vector Data Description (RUFF *et al.*, 2019)	OCL	espaço-vetorial	Inglês	Duas bases de dados são analisadas e o resultado é similar ao OCSVM.
Fuzzy clustering (LAZHAR, 2019)	OCL	espaço-vetorial	Inglês	O método é usado apenas para encontrar anomalias em bases de dados textuais para melhorar o desempenho de algoritmos, não aplicável a identificação de elementos da classe de interesse.
Agrupamento com OCSVM (SONBHADRA; AGARWAL; NAGABHUSHAN, 2020)	OCL	espaço-vetorial	Inglês	Abordagem avaliada considerando diferentes algoritmos de agrupamento, não sendo comparada a outros algoritmos OCL.
RC-SVM (LI; LIU, 2003)	PUL	espaço-vetorial	Inglês	A aplicação iterativa de SVM (self-training) aumenta o custo computacional e pode diminuir o desempenho de classificação.
FNED (LIU; WU, 2020)	PUL	espaço-vetorial	Inglês e Chinês	Embora o modelo desempenhe bem nas bases de dados avaliadas, o modelo não é interpretável e explicável, sendo uma limitação para detecção de notícias falsas, o treinamento iterativo da rede neural gera alto custo computacional. Também demanda informações adicionais de usuários que compartilharam notícia.
S-EM (LIU *et al.*, 2002)	PUL	espaço-vetorial	Inglês	S-EM apenas desempenha bem quando o conjunto não rotulado possui poucos exemplos de interesse. Como Naïve Bayes tolera pouco ruído no conjunto de treinamento, ele tende a classificar a maioria dos documentos não rotulados como negativos.
PE-PUC (YU; LI, 2007)	PUL	espaço-vetorial e redes	Inglês	Se o conjunto de documentos de interesse inicialmente rotulado é muito pequeno, ele não será suficiente para representar a distribuição da classe de interesse e a maioria dos exemplos não rotulados serão classificados como negativos.
PUL-SAAT (XU *et al.*, 2019)	PUL	espaço-vetorial	Inglês	A abordagem é comparada apenas com algoritmos supervisionados, e seu desempenho é altamente dependente do valor da perturbação aleatória.

CAPÍTULO

4

DETECÇÃO DE NOTÍCIAS FALSAS A PARTIR DE POUCOS RÓTULOS DE INTERESSE

Considerando os benefícios de representações de redes, tanto para modelar diferentes tipos de relações de uma coleção de textos quanto no cenário de aprendizado semissupervisionado, a proposta desta tese consiste na detecção de notícias falsas por meio de uma abordagem baseada em redes heterogêneas e Aprendizado Positivo e Não Rotulado.

A proposta é fundamentada no algoritmo PU-LP, descrito na Seção 3.6. PU-LP é um algoritmo de redes homogêneas e propagação de rótulos, que identifica potenciais nós de interesse e não interesse presentes em dados não rotulados. PU-LP foi avaliado em bases de dados contendo atributos numéricos com baixa dimensão, considerando redes de similaridade. Neste trabalho, o algoritmo foi adaptado para classificação de textos e aplicado no cenário de detecção de notícias falsas, no qual a classe "fake" é a classe de interesse. A seguir, um algoritmo de propagação de rótulos classifica os demais documentos não rotulados.

Considerando a complexidade do problema de detecção de notícias falsas e o fato de que o enriquecimento do modelo de representação com diferentes tipos de relações pode aumentar o desempenho de classificadores de textos (DEEPAK *et al.*, 2021; YANG *et al.*, 2020; SHI; PHILIP, 2017; ROSSI; REZENDE; LOPES, 2015), diferentes representações de redes foram propostas em relação às originalmente utilizadas por MA; ZHANG (2017).

Primeiramente, foram geradas redes baseadas em similaridade, considerando notícias como nós. Em seguida, foram adicionados termos representativos, e conectados às notícias, tornando a rede heterogênea. Termos foram extraídos do conteúdo da publicação considerando técnicas como extração de unigramas, bigramas e palavras-chave. Esta característica foi escolhida para compor a rede por ser considerada genérica, isto é, termos relevantes podem ser extraídos de qualquer coleção de notícias. A inclusão de termos em redes de documentos também torna o processo de rotulação mais preciso, já que a conexão entre documentos e termos pode aumentar

a pontuação de documentos para sua classe real. Além disso, termos são frequentemente usados para discriminação de conteúdo verdadeiro e falso (YAN *et al.*, 2020; ROSSI; REZENDE; LOPES, 2015; CHAKRAVARTHY *et al.*, 2014; AGGARWAL; LI, 2011; HASSAN *et al.*, 2020; AHMED; TRAORE; SAAD, 2017; PÉREZ-ROSAS *et al.*, 2017; RUBIN *et al.*, 2016; MIHALCEA; STRAPPARAVA; PULMAN, 2010). Além dos termos, características como emotividade, número médio de palavras por sentença e pausalidade, que já demonstraram efetividade na discriminação de conteúdo verídico e falso (SANTOS; PARDO, 2020), foram calculadas e incorporadas à rede.

Dentre as características analisadas, a que mais contribuiu no aumento de desempenho de classificação foi a inclusão de termos relevantes, extraídos inicialmente com uma estratégia simples de Bag-of-Words. Buscando a inclusão mais assertiva de termos e a melhoria de resultados, optou-se por utilizar estratégias mais promissoras de extração de palavras-chave, como Yake! (CAMPOS *et al.*, 2020), uma ferramenta não supervisionada que calcula estatísticas de *n*-gramas como co-ocorrência, frequência e desduplicação (termos que apresentam significados similares a outros são eliminados), e apresenta destaque em avaliações experimentais da literatura (PISKORSKI *et al.*, 2021).

Para classificação de nós não rotulados, foram considerados os algoritmos de propagação de rótulos GNetMine e *Label Propagation through Heterogeneous Network*, extensivamente utilizados e consolidados na literatura (HUA; YANG; QIU, 2021), e o algoritmo *Graph Attention Neural Event Embeddings* (GNEE). GNEE é baseado em regularização e redes de atenção e realiza o aprendizado de características importantes de um nó com base em seus nós vizinhos relevantes (MATTOS; MARCACINI, 2021).

Para a avaliação do desempenho da abordagem, foram considerados algoritmos de OCL e PUL extensivamente empregados em classificação de textos (GÔLO; MARCACINI; ROSSI, 2020): *k-Means*, *k-Nearest Neighbors Density-based* (*k*-NND), *One-Class Support Vector Machine* (OCSVM), *Rocchio Support Vector Machine* (RC-SVM), *Dense Autoencoder* (DAE) (MANEVITZ; YOUSEF, 2007), além dos algoritmos estado da arte baseados em OCGNNs, *One Class Graph Convolutional Neural Network* (OC-GCN), *One Class Graph Attention Network* (OC-GAT) e *One Class Graph SAGE* (OC-GraphSAGE), abordados no Capítulo 3.

A fim de gerar representações estruturadas, utilizadas como dados de entrada para os algoritmos OCL e PUL, foram avaliados dois modelos que consideram o texto completo da notícia: Bag-of-Words (BoW) (SALTON, 1989) e *embeddings* de documentos geradas com Doc2Vec (D2V) (LE; MIKOLOV, 2014). Tais representações também foram utilizadas para calcular similaridades e gerar relações entre documentos na rede de PU-LP. Os algoritmos foram avaliados considerando diferentes porcentagens de notícias falsas rotuladas (10%, 20% e 30%), macro F_1 e F_1 relacionada à classe de interesse (*fake* F_1 ou *interest* F_1).

Este capítulo é dividido da seguinte forma: na Seção 4.1 são apresentadas informações sobre as bases de dados e modelos de representação utilizados nos experimentos. Na Seção 4.2

é apresentada a abordagem Positive and Unlabeled Learning by Label Propagation para Detecção de Fake News (PULP-FND) (SOUZA *et al.*, 2022), que adiciona termos extraídos com Bag-of-Words na rede de notícias do algoritmo PU-LP, bem como a configuração experimental, resultados e discussões. Na Seção 4.3 é apresentada a avaliação da abordagem diante de novas características incluídas na rede heterogênea (emotividade, número médio de palavras por sentença e pausalidade) (SOUZA *et al.*, 2021). Na Seção 4.4 são apresentados novos experimentos, que consistem na adição de palavras-chave extraídas com Yake! à rede heterogênea. Na Seção 4.5 é apresentada a principal contribuição desta tese, intitulada Atenção em Palavras Chaves para Detecção de Fake News usando PU-LP (AK-PULP-FND), que considera mecanismos de atenção para classificação de nós não rotulados na rede de notícias e termos. Por fim, na Seção 4.6 é apresentada uma discussão sobre as vantagens e limitações da abordagem PU-LP.

4.1 Coleções de Notícias, Modelos de Representação e Configuração Experimental

As coleções de notícias utilizadas nos experimentos desta tese foram as bases em português Fake.BR (FBR) (MONTEIRO *et al.*, 2018) (versão truncada) e Fact-checked News (FCN) (RIBEIRO, 2019), e as bases em inglês FakeNewsNet (FNN) (SHU *et al.*, 2020) e FakeNewsCorpus (FNC)[1], detalhadas na Subseção 2.3.1. Para a coleção Fact-checked News, por serem notícias checadas e não as originalmente escritas, foram removidos alguns termos como “fato”, “fake”, “verdadeiro”, “falso”, “#fake”, “verificamos”, “montagem”, “erro” e “checagem”. Tais termos foram considerados como *stopwords* de domínio e foram extraídos com o intuito de evitar o enviesamento do algoritmo de classificação.

Quanto a base FakeNewsNet, foram utilizadas somente as notícias coletadas de *GossipCop* (SHU *et al.*, 2020). A base possui 16.095 notícias reais e 4.937 notícias falsas. Entretanto, após uma análise inicial, verificou-se um grande desbalanceamento em relação ao número de tokens, ocasionado por possíveis erros de coleta. Para minimizar este problema, notícias que possuíam entre 200 e 600 tokens foram selecionadas, restando 5.298 notícias reais e 1.705 notícias falsas na coleção.

A coleção de notícias FakeNewsCorpus, é composta por milhões de notícias coletadas de 1001 domínios. Por falta de recursos computacionais para processar essa quantidade de dados, foi realizado um pré-processamento no qual foram criadas 3 bases de dados derivadas da principal: fakeNewsCorpus 0, 1 e 2. Para cada base, foram selecionadas aleatoriamente 3.000 notícias reais e 3.000 notícias falsas de assuntos distintos. Na Tabela 3 são apresentadas estatísticas das bases de dados.

Após o pré-processamento das seis bases de dados, que envolveu a remoção de *stopwords*

[1] https://github.com/several27/FakeNewsCorpus

Base de dados	#Fake	#Real	Nº médio de termos $\bar{M}$	Mediana	Desvio padrão σ
Fake.BR	3.600	3.600	183.63	157.00	115.85
FakeNewsNet	1.705	5.298	573.93	515.00	191.75
Fact-checked News	1.044	1.124	505.41	408.00	428.56
FakeNewsCorpus 0	3.000	3.000	660.47	631.00	204.46
FakeNewsCorpus 1	3.000	3.000	635.61	595.00	198.97
FakeNewsCorpus 2	3.000	3.000	630.86	592.00	193.68

Tabela 3 – Estatísticas das coleções de notícias.

e radicalização dos termos, vetores de características para representação de notícias foram obtidos considerando duas estratégias: o modelo tradicional BoW e D2V. As *Bag-of-Words* foram geradas usando tf-idf como esquema de pesos dos termos, ignorando-se termos que apareciam em menos de dois documentos. Em D2V foi usada a união dos modelos *Distributed Memory* e *Distributed Bag-of-Words* para gerar as *embeddings* de documentos. Para o treinamento destes modelos, foram consideradas a média e concatenação dos vetores de palavras na saída da camada oculta, assim como recomendado em LE; MIKOLOV (2014). Além disso, foram considerados o intervalo de número máximo de épocas $\in \{100, 1000\}$, taxa de aprendizado inicial (*learning rate*) $\alpha = 0.025$ e decaimento da taxa de aprendizado $\alpha_{min} = 0.0001$, número de dimensões de cada modelo $= 500$, *window size* $= 8$, e minimum count $= 1$ (MARTINČIĆ-IPŠIĆ; MILIČIĆ; TODOROVSKI, 2019; PITA; PAPPA, 2018; LE; MIKOLOV, 2014). Os quatro modelos gerados foram os seguintes:

- Representação 1: Método=*average*, *Max epochs*=100;
- Representação 2: Método=*average*, *Max epochs*=1000;
- Representação 3: Método=*concatenation*, *Max epochs*=100;
- Representação 4: Método=*concatenation*, *Max epochs*=1000.

As bases de dados representam cenários distintos quanto a forma de coleta, o desbalanceamento, língua e quantidade de tópicos e foram utilizadas para avaliar o comportamento das abordagens propostas, fundamentadas no algoritmo PU-LP. As abordagens são descritas nas próximas seções.

4.2 PULP-FND: Positive and Unlabeled Learning by Label Propagation para Detecção de Fake News

Na Figura 8 é apresentada a abordagem *Positive and Unlabeled Learning by Label Propagation para Detecção de Fake News*. A figura é dividida em 9 etapas: coleção de notícias, modelo de representação, matriz k-NN, matriz de similaridade com índice de Katz, extração de conjuntos, rotulação da rede k-NN, seleção de termos relevantes, adição de termos na rede k-NN e propagação de rótulos, detalhadas a seguir.

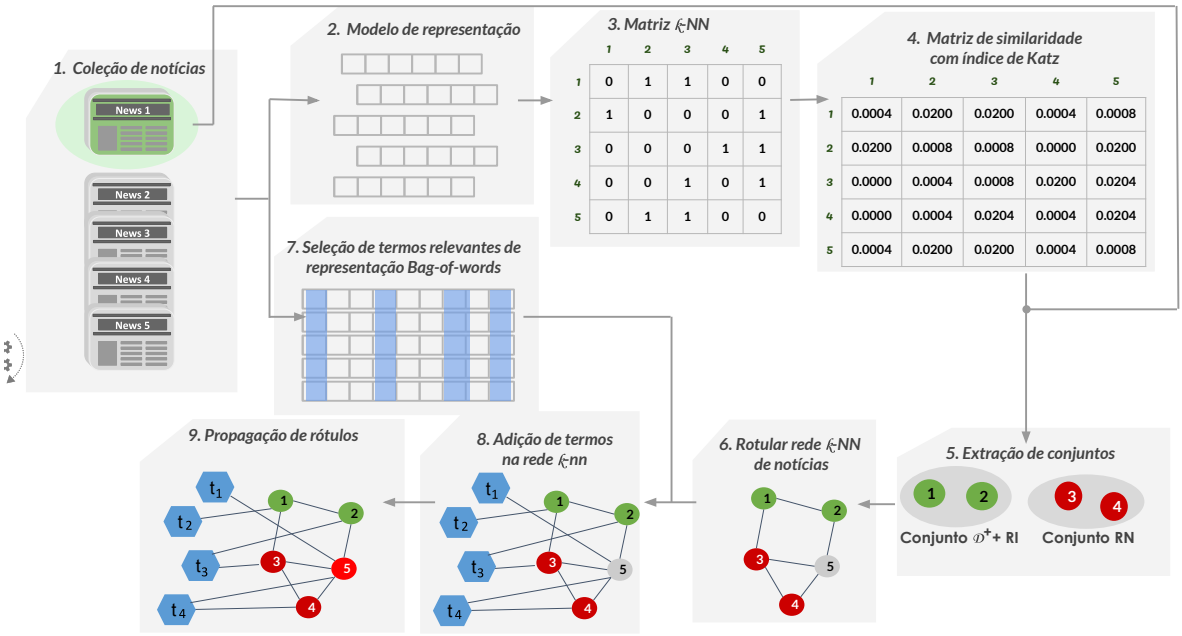


Figura 8 – Etapas do algoritmo PULP-FND para detecção de notícias falsas.

Coleção de Notícias e Modelo de Representação: seja $\mathscr{D} = \{d_1, \ldots, d_l, d_{l+1}, \ldots, d_{l+u}\}$ um conjunto de notícias e $\mathscr{C} = \{falsa, real\}$ o conjunto de rótulos possíveis. Os primeiros l elementos de $\mathscr{D}$ são notícias falsas rotuladas, que compõem o conjunto $\mathscr{D}^+$. As u notícias remanescentes são não rotuladas (falsas ou reais), compondo o conjunto $\mathscr{D}^u$, $u \gg l$. O conjunto de notícias precisa ser pré-processado, e um modelo de representação, como BoW ou *embeddings* de documentos (AGGARWAL, 2018), precisa ser adotado para transformar notícias em dados estruturados. O modelo de representação e uma métrica de distância são necessários para calcular uma matriz de adjacência considerando o conjunto completo de exemplos $\mathscr{D}$, na qual notícias com conteúdos similares possuem baixa distância entre si. A matriz de adjacência é utilizada como base para a criação de uma matriz de k-NN, detalhada a seguir.

Matriz k-NN, Matriz de Similaridade com Índice de Katz, Extração de Conjuntos e Rotulação da Rede de Notícias: a matriz de adjacência é usada como base para construção da matriz k-NN, chamada A, de forma que $A_{i,j} = 1$ se a notícia d_j é uma das k vizinhas mais próximas da notícia d_i, e $A_{i,j} = 0$ caso contrário. Uma rede $\mathscr{N}$ também é criada por meio da matriz k-NN. A rede $\mathscr{N}$ pode ser definida como uma tripla $\mathscr{N} = \langle \mathscr{O}, \mathscr{R}, \mathscr{W} \rangle$. $\mathscr{O}$ é o conjunto de objetos, no qual $\mathscr{O}_D$ é composto por notícias e $\mathscr{O} - \mathscr{O}_D$ é composto por objetos que representem características de notícias. $\mathscr{R}$ é o conjunto de relações r_{o_i,o_j} entre pares de objetos $o_i, o_j \in \mathscr{O}$, conforme a matriz k-NN, e $\mathscr{W}$ é o conjunto de pesos w_{o_i,o_j} destas relações. Os pesos são dados pela similaridade de cosseno entre as notícias correspondentes.

Por meio da similaridade de cosseno, assume-se que duas notícias serão considerados da mesma classe se elas estão diretamente conectados na rede. Além disso, para atribuir rótulos considerando uma medida de similaridade global, usa-se o índice de Katz (LÜ; JIN; ZHOU, 2009; KATZ, 1953). Assim, se dois nós possuem muitos vizinhos em comum, eles também

serão prováveis de pertencerem à mesma classe. Visando o conjunto inicialmente rotulado e W, PU-LP infere dois conjuntos de $\mathcal{O}^u$: o conjunto de notícias de interesse confiável *RI* (conjunto de potenciais notícias falsas) e conjunto de notícias de não interesse confiável *RN* (conjunto de potenciais notícias reais), os quais serão usados como entrada pelos algoritmos de propagação para rotular a rede k-NN de notícias. Mais detalhes podem se vistos no Algoritmo 3 (MA; ZHANG, 2017).

Seleção de Termos Relevantes de Representação *Bag-of-Words*, Adição na Rede k-NN e Propagação de Rótulos: após a inferência dos conjuntos *RI* e *RN*, novas relações entre notícias e termos representativos são adicionadas na rede. Foram considerados termos unigramas e bigramas gerados a partir do modelo de representação *Bag-of-words*, com tf-idf como esquema de pesos (YAN *et al.*, 2020; MANNING; RAGHAVAN; SCHÜTZE, 2008; FELDMAN; SANGER *et al.*, 2007), que pondera a frequência do termo pelo inverso do número de documentos em que o termo ocorre.

Para a seleção de termos relevantes a serem incluídos na rede, foram removidas *stopwords* e aplicado radicalização aos termos restantes. Foi criada uma nova *Bag-of-Words* considerando os parâmetros: variação de *n-gram* , que determina se a matriz será criada apenas com unigramas ou unigramas e bigramas; e *minimum df*, que ignora termos que não estejam presentes em um número mínimo de documentos informado. A partir deste modelo de representação, foram selecionados termos com tf-idf acima de um limiar ℓ para serem incluídos na rede k-NN como nós não rotulados. Foi importante limitar o vocabulário para palavras mais discriminativas para redução do tamanho da rede e complexidade computacional (DEEPAK *et al.*, 2021; YAN *et al.*, 2020; CHAKRAVARTHY *et al.*, 2014; AGGARWAL; LI, 2011). O valor do tf-idf foi usado como peso da aresta entre a notícia e o termo. Após a construção da rede de notícias e termos, a próxima etapa consistiu na execução do algoritmo de propagação de rótulos por meio de algoritmos transdutivos de redes heterogêneas.

4.2.1 Configuração Experimental e Critérios de Avaliação

Para a avaliação da abordagem PULP-FND, foram considerados quatro algoritmos tradicionais de OCL: *k-Nearest Neighbors Density-based* (k-NND), *k-Means*, *One-Class Support Vector Machine* (OCSVM) e *Dense Autoencoder*, e dois algoritmos de PUL: o tradicional PU-LP, contendo apenas redes compostas por notícias, e RC-SVM. A seguir, são apresentadas as configurações experimentais da abordagem proposta, bem como dos algoritmos de avaliação.

- **k-NND** (TAN *et al.*, 2019): $k = 1+3*p, p \in [1..7]$ e cosseno como medida de similaridade.
- **k-Means** (TAN *et al.*, 2019): $k = 1+2*p, p \in [1..9]$, 100 como o número máximo de iterações e cosseno como medida de similaridade. Foram realizados 10 experimentos e escolhido o resultado do agrupamento com mais alta coesão.

- **One-Class Support Vector Machines (OCSVM)** (MANEVITZ; YOUSEF, 2001): $\gamma = 1 * 10^{p}(), p \in [-3..1]$, $\upsilon = 0.05 + 0.1 * q, q \in [0..9]$, e os *kernels* Linear e Radial Basis Function[2].

- **DAE**: uma única camada oculta com $h \in \{2, 6, 12\}$ (número de neurônios na camada oculta) (MANEVITZ; YOUSEF, 2007), ADAM como otimizador com $\eta = 0.01$ (*learning rate*) (KINGMA; BA, 2015), os pesos dos vetores de documentos foram normalizados para 1 (vetores foram divididos pelas normalizações) (MANEVITZ; YOUSEF, 2007), 200 como o número máximo de iterações, e cosseno como medida de similaridade entre a entrada e saída. As funções de ativação da camada oculta e de saída são ReLU e sigmoide.

k-NND, k-Means, e DAE requerem um limiar que define se um novo texto pertence à classe de interesse. Foram considerados limiares definidos manual e automaticamente. No caso dos limiares manuais, foram definidos $limiar \in \{0.05 \times z, z \in \mathbb{N} : 1 \leq z \leq 19\}$. A abordagem 6σ foi usada para seleção de limiares de forma automática (MUIR, 2005). Neste caso, os *scores* $f(\mathbf{d}_i)$ são gerados para documentos de treinamento, a média (μ) e desvio padrão são calculados (σ), e então o $limiar \in \{\mu - 3\sigma, \mu - 2\sigma, \mu - 1\sigma, \mu, \mu + 1\sigma, \mu + 2\sigma, \mu + 3\sigma\}$.

Para **RC-SVM** (LI; LIU, 2003), foram usados $\alpha = \{0.2, 0.4, 0.6, 0.8, 1.0\}$, e $\beta = 1 - \alpha$ no algoritmo Rocchio. Também foram considerados $C \in \{0.1, 1.0, 10\}$, e kernel linear no algoritmo SVM.

Para o PU-LP composto apenas de nós de notícias (rede homogênea) (MA; ZHANG, 2017), foram considerados os seguintes parâmetros: $k = [5, 6, 7]$ e cosseno como medida de similaridade na construção da matriz k-NN. Para a extração de conjuntos de interesse e não interesse confiáveis, foram usados: $m = 2$, $\lambda = [0.6, 0.8]$, e $\alpha = [0.005, 0.01, 0.02]$. Os valores foram definidos conforme sugerido em MA; ZHANG (2017).

Para a abordagem PULP-FND, que contém relações entre nós de notícias e termos representativos (rede heterogênea), foram considerados os seguintes parâmetros: $\ell = 0.08$ para a seleção de unigramas e bigramas representativos. O valor de ℓ foi escolhido após uma análise estatística da amostra, indicando que cerca de 25% dos termos da BoW apresentavam tf-idf superior a 0.08. Após a inclusão de todas as notícias e termos na rede $\mathcal{N}$, foi realizada a normalização da rede a fim de mitigar possíveis distorções devido a diferentes intervalos de valores entre tipos de relações distintos. Desta forma, o peso da aresta de um objeto $o_i \in \mathcal{O}_l$, $o_j \in \mathcal{O}_m$, é dado pela Equação 4.1.

$$w_{o_i,o_j} = \frac{w_{o_i,o_j}}{\sum_{o_k} w_{o_i,o_k}}, o_i \in \mathcal{O}_l, o_j \in \mathcal{O}_m, \text{ and } o_k \in \mathcal{O}_m. \tag{4.1}$$

[2] https://scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html. Acessado em novembro de 2023.

Para propagação de rótulos, nos experimentos realizados contendo apenas redes homogêneas foram utilizadas as versões iterativas dos algoritmos de classificação transdutiva baseada em regularização GFHF (ZHU; GHAHRAMANI; LAFFERTY, 2003) e LLGC (ZHOU *et al.*, 2004), descritos na Subseção 3.6.2. Ambos foram escolhidos por serem bem estabelecidos na literatura (HUA; YANG; QIU, 2021). Para a abordagem PULP-FND, que considera redes de notícias e termos, foram utilizados LPHN (ROSSI, 2016) e GNetMine (JI *et al.*, 2010), descritos na Subseção 3.6.3, os quais são extensões dos algoritmos GFHF e LLGC.

Para os algoritmos de propagação de rótulos LLGC e GFHF (redes homogêneas) e GNetMine e LPHN (redes heterogêneas), foram usados limiar de convergência = 0,00005 e número máximo de iterações = 1.000. Para os algoritmos LLGC e GNetMine, $\alpha = \{0.1, 0.5\}$ e $\lambda = 1$ foram utilizados, como sugerido em SANTOS (2018). Tais parâmetros foram utilizados considerando os experimentos de todas as bases de dados.

Foi utilizado um esquema de validação cruzada de *folds* adaptado para problemas de OCL e PUL. Neste caso, o conjunto de notícias falsas ($\mathscr{D}^+$) foi aleatoriamente dividido em 10 *folds*. A fim de simular o ambiente de aprendizado semissupervisionado, foram realizados diferentes experimentos considerando como dados rotulados 1, 2 ou 3 *folds*. Os *folds* restantes e o conjunto de notícias reais são: (i) considerados como documentos de teste para algoritmos OCL; (ii) considerados como documentos não rotulados para algoritmos PUL.

Também foi proposto um modelo de referência que usa aprendizado semissupervisionado binário (BL) para avaliar o processo de rotulação dos conjuntos de interesse confiável (conjunto *RI*) e conjunto negativo confiável (conjunto *RN*) do algoritmo PU-LP. Para esta análise, o conjunto de notícias reais foi aleatoriamente dividido em 10 subconjuntos. No esquema de validação cruzada, para cada *fold* de notícias falsas utilizado no conjunto de treinamento, um *fold* de notícias reais também foi utilizado. Isto é, o algoritmo trabalha com *p folds* de notícias reais e *p folds* de notícias falsas no conjunto inicialmente rotulado, $p = \{1, 2, 3\}$. Com a rede obtida pela matriz *k*-NN e considerando o conjunto de treinamento como o conjunto de nós rotulados, o algoritmo de propagação de rótulos infere a classe das notícias remanescentes por meio da estrutura da rede. Foram utilizados valores de *k* no intervalo de $[5, 7]$. Como algoritmos de propagação, foram avaliados GFHF e LLGC, com os mesmos parâmetros considerados na versão de redes homogêneas de PU-LP.

Como medida de avaliação, foram usadas $F_1 = (2 \cdot precisão \cdot revocação)/(precisão + revocação)$ considerando notícias falsas como a classe positiva (*fake* F_1). Também foi avaliada a média da F_1 para ambas as classes (macro F_1).

4.2.2 Resultados e Discussões

Nesta seção são apresentados os resultados alcançados considerando a avaliação experimental apresentada anteriormente. O objetivo é demonstrar que abordagens OCL e PUL podem

ser relevantes para classificação de notícias, em especial pelo fato destes modelos de classificação usarem um conjunto pequeno de notícias falsas rotuladas, eliminando a necessidade de rotular notícias de classes não interessantes. Além disso, deseja-se demonstrar que a abordagem proposta, PULP-FND, pode alcançar resultados tão bons quanto ao modelo de referência, e que a inclusão de termos na rede de notícias melhora o desempenho de algoritmos de propagação de rótulos, em especial quando notícias falsas estão distribuídas no espaço de características.

Tabela 4 – Resultados das bases de dados Fact-checked News, Fake.BR e FakeNewsNet usando Bag-of-Words como modelo de representação. 10%, 20% e 30% indicam a porcentagem de notícias falsas rotuladas no conjunto de treinamento. Para o algoritmo de referência (BL), o conjunto rotulado possui a mesma porcentagem de notícias reais. Os melhores resultados dos algoritmos PUL e OCL estão destacados em cinza.

	Fake F_1			**Macro F_1**		
	10%	**20%**	**30%**	**10%**	**20%**	**30%**
Fact-checked news						
PULP-FND (LPHN)	0.8325	0.8722	0.8721	0.8479	**0.8844**	**0.8892**
PULP-FND (GNM)	**0.8692**	**0.8748**	**0.8745**	**0.8702**	0.8824	0.8873
PU-LP (GFHF)	0.8646	0.8637	0.8506	0.8681	0.8687	0.8641
PU-LP (LLGC)	0.8618	0.8681	0.8643	0.8629	0.8732	0.8770
RC-SVM	0.0210	0.1604	0.7388	0.0239	0.1601	0.7508
k-NND	0.7185	0.7037	0.6772	0.7469	0.7394	0.7269
k-Means	0.7402	0.7250	0.7094	0.7538	0.7373	0.7384
OCSVM	0.7793	0.7832	0.7736	0.7108	0.7043	0.6886
DAE	0.7146	0.7215	0.7055	0.5399	0.6648	0.6620
BL (GFHF)	0.8775	0.8858	0.8840	0.8745	0.8838	0.8820
BL (LLGC)	0.8098	0.8856	0.7911	0.8388	0.8879	0.8274
Fake.BR						
PULP-FND (LPHN)	0.3423	0.4241	0.5547	0.5000	0.5650	0.6395
PULP-FND (GNM)	0.6496	**0.6701**	0.6652	**0.6647**	**0.6948**	**0.7107**
PU-LP (GFHF)	**0.6538**	0.6494	0.6451	0.6242	0.6245	0.6586
PU-LP (LLGC)	0.6468	0.6574	0.6502	0.6481	0.6776	0.6951
RC-SVM	0.2530	0.5564	**0.6865**	0.3263	0.5914	0.7048
k-NND	0.6477	0.6272	0.6005	0.6433	0.6175	0.5899
k-Means	0.6444	0.6193	0.5908	0.6433	0.6175	0.5878
OCSVM	0.5602	0.5363	0.5078	0.5765	0.5505	0.5220
DAE	0.6435	0.6170	0.5848	0.5399	0.6648	0.6620
BL (GFHF)	0.6918	0.6806	0.6889	0.6890	0.6802	0.6871
BL (LLGC)	0.6420	0.5520	0.5208	0.6949	0.6407	0.6222
FakeNewsNet						
PULP-FND (LPHN)	0.4473	0.4209	0.3901	0.5453	0.5176	0.4796
PULP-FND (GNM)	0.4827	0.4787	0.4500	0.6226	0.6131	0.5848
PU-LP (GFHF)	0.4741	0.4711	0.4512	0.6213	0.6168	0.5985
PU-LP (LLGC)	0.4723	0.4669	0.4408	0.6114	0.6018	0.5732
RC-SVM	0.1436	0.4340	0.5123	0.2601	0.5316	0.5278
k-NND	**0.5492**	**0.5431**	**0.5273**	0.6547	0.6446	0.6336
k-Means	0.5379	0.5253	0.5106	0.6555	0.6449	0.6333
OCSVM	0.4364	0.4182	0.3932	0.5955	0.5868	0.5745
DAE	0.4968	0.5009	0.4740	**0.6791**	**0.6803**	**0.6711**
BL(GFHF)	0.5078	0.5057	0.4893	0.6838	0.6820	0.6696
BL (LLGC)	0.3830	0.4192	0.4023	0.6280	0.6453	0.6327

Nas Tabelas 4 a 7 são apresentados os melhores resultados de F_1 de interesse (*fake* F_1) e macro obtidos por cada algoritmo, considerando o conjunto de parâmetros definidos na

Tabela 5 – Resultados das bases de dados derivadas da coleção *FakeNewsCorpus* usando Bag-of-Words como modelo de representação. PULP-FND denota a abordagem proposta, que considera redes heterogêneas. PU-LP considera redes homogêneas (notícias). 10%, 20% e 30% indicam a porcentagem de notícias falsas rotuladas no conjunto de treinamento. Para o algoritmo de referência (BL), o conjunto rotulado possui a mesma porcentagem de notícias reais. Os melhores resultados dos algoritmos PUL e OCL estão destacados em cinza.

	Fake F_1			**Macro F_1**		
	10%	**20%**	**30%**	**10%**	**20%**	**30%**
FakeNewsCorpus 0						
PULP-FND (LPHN)	0.7521	0.7878	0.8012	0.7811	0.8125	0.8333
PULP-FND (GNM)	**0.8279**	**0.8389**	**0.8441**	**0.8334**	**0.8516**	**0.8622**
PU-LP (GFHF)	0.8113	0.8114	0.8106	0.8194	0.8270	0.8295
PU-LP (LLGC)	0.8218	0.8305	0.8390	0.8280	0.8444	0.8585
RC-SVM	0.6564	0.6859	0.6729	0.7081	0.7167	0.7555
k-NND	0.6835	0.6762	0.6664	0.7257	0.7262	0.7217
k-Means	0.6634	0.6571	0.6287	0.6921	0.7071	0.6860
OCSVM	0.6564	0.6354	0.6113	0.6837	0.6641	0.6414
DAE	0.6569	0.6420	0.6118	0.6556	0.6439	0.6454
BL (GFHF)	0.8283	0.8263	0.7092	0.8327	0.8357	0.8343
BL (LLGC)	0.7092	0.8410	0.6812	0.7643	0.8502	0.7541
FakeNewsCorpus 1						
PULP-FND (LPHN)	0.7453	0.7832	0.7928	0.7660	0.8045	0.8244
PULP-FND (GNM)	**0.8271**	**0.8352**	**0.8403**	**0.8328**	**0.8480**	**0.8586**
PU-LP (GFHF)	0.7997	0.7988	0.7926	0.8113	0.8140	0.8189
PU-LP (LLGC)	0.8173	0.8235	0.8307	0.8247	0.8358	0.8513
RC-SVM	0.5442	0.6973	0.6114	0.6176	0.7304	0.6928
k-NND	0.6523	0.6432	0.6192	0.7155	0.7128	0.6976
k-Means	0.6517	0.6301	0.6033	0.6827	0.6907	0.6663
OCSVM	0.5920	0.5659	0.5384	0.6644	0.6436	0.6201
DAE	0.6794	0.6687	0.6458	0.6758	0.6727	0.6714
BL (GFHF)	0.8111	0.8159	0.8145	0.8229	0.8272	0.8251
BL (LLGC)	0.7472	0.7129	0.6792	0.7807	0.7654	0.7483
FakeNewsCorpus 2						
PULP-FND (LPHN)	0.7007	0.7473	0.7554	0.7431	0.7823	0.8005
PULP-FND (GNM)	**0.8201**	**0.8374**	**0.8425**	**0.8294**	**0.8506**	**0.8623**
PU-LP (GFHF)	0.7943	0.8063	0.8096	0.8108	0.8222	0.8371
PU-LP (LLGC)	0.8112	0.8276	0.8346	0.8208	0.8405	0.8558
RC-SVM	0.4848	0.6389	0.5118	0.6139	0.7008	0.6542
k-NND	0.6580	0.6447	0.6237	0.7243	0.7266	0.7144
k-Means	0.6443	0.6171	0.6005	0.7046	0.7005	0.6799
OCSVM	0.5843	0.5651	0.5358	0.6688	0.6534	0.6317
DAE	0.6824	0.6678	0.6475	0.6704	0.6678	0.6475
BL (GFHF)	0.8309	0.8338	0.8308	0.8422	0.8449	0.8420
BL (LLGC)	0.7474	0.7493	0.7625	0.7939	0.7978	0.8060

Subseção 4.2.1, os modelos de representação Bag-of-Words e Doc2Vec e as seis coleções de notícias. As três primeiras colunas correspondem a F_1 *fake* e as três últimas colunas a F_1 macro. 10%, 20% e 30% indicam a porcentagem de notícias falsas usadas no conjunto de treinamento dos algoritmos. As primeiras duas linhas representam a abordagem PULP-FND, considerando algoritmos de propagação de rótulos LPHN e GNM. A terceira e quarta linhas representam o PU-LP com redes homogêneas, usando GFHF e LLGC como propagação de rótulos. Nas linhas seguintes estão os algoritmos OCL e PUL usados para avaliação da proposta. As duas últimas

Tabela 6 – Resultados das bases de dados Fact-checked News, Fake.BR e FakeNewsNet usando Doc2Vec como modelo de representação. PULP-FND denota a abordagem proposta, com redes heterogêneas, e PU-LP denota o uso de redes homogêneas. 10%, 20% e 30% indicam a porcentagem de notícias falsas rotuladas no conjunto de treinamento. Para o algoritmo de referência (BL), o conjunto rotulado possui a mesma porcentagem de notícias reais. Os melhores resultados dos algoritmos PUL e OCL estão destacados em cinza.

	Fake F_1			**Macro F_1**		
	10%	**20%**	**30%**	**10%**	**20%**	**30%**
Fact-checked news						
PULP-FND (LPHN)	0.8884	0.8851	0.8811	0.8955	0.8971	0.8995
PULP-FND (GNM)	0.9036	0.9131	0.9206	0.9076	0.9205	0.9312
PU-LP (GFHF)	0.9041	0.8895	0.8756	0.9054	0.8966	0.8894
PU-LP (LLGC)	**0.9110**	**0.9163**	**0.9214**	**0.9159**	**0.9243**	**0.9325**
RC-SVM	0.6117	0.5874	0.6416	0.4830	0.5080	0.6016
k-NND	0.6399	0.6162	0.5874	0.6472	0.6225	0.5973
k-Means	0.7116	0.7027	0.7197	0.6865	0.6905	0.6905
OCSVM	0.8490	0.7942	0.8144	0.8098	0.6979	0.7530
DAE	0.6863	0.6780	0.6671	0.6887	0.6842	0.6872
BL (GFHF)	0.9016	0.9016	0.8957	0.9023	0.9023	0.8964
BL (LLGC)	0.8308	0.8329	0.8216	0.8646	0.8661	0.8594
Fake.BR						
PULP-FND (LPHN)	0.4409	0.5764	0.5704	0.5304	0.6187	0.6611
PULP-FND (GNM)	0.6219	**0.6586**	**0.6671**	0.6570	**0.7031**	**0.7289**
PU-LP (GFHF)	0.6055	0.6158	0.6142	0.5843	0.6313	0.6658
PU-LP (LLGC)	0.5980	0.6358	0.6421	0.6397	0.6845	0.7086
RC-SVM	0.6139	0.6398	0.6273	0.5539	0.6125	0.5863
k-NND	0.6497	0.6323	0.6066	0.6469	0.6259	0.5988
k-Means	0.6460	0.6195	0.5946	0.6454	0.6184	0.5851
OCSVM	**0.8244**	0.6128	0.6580	**0.8172**	0.6104	0.6765
DAE	0.6429	0.6156	0.5836	0.6687	0.6634	0.6606
BL (GFHF)	0.6547	0.6520	0.6547	0.6356	0.6352	0.6352
BL (LLGC)	0.4698	0.4705	0.4705	0.5825	0.5997	0.5997
FakeNewsNet						
PULP-FND (LPHN)	0.4557	0.4171	0.4010	0.5830	0.5504	0.5699
PULP-FND (GNM)	0.5267	0.5179	0.5027	0.6748	0.6666	0.6623
PU-LP (GFHF)	0.4662	0.4517	0.4352	0.6527	0.6210	0.6283
PU-LP (LLGC)	0.5112	0.5083	0.4964	0.6669	**0.6677**	0.6649
RC-SVM	0.7027	0.7518	0.6694	0.5217	0.6195	0.5427
k-NND	0.8575	**0.8539**	**0.8499**	0.6564	0.6471	0.6342
k-Means	**0.8576**	0.8537	0.8497	0.6558	0.6460	0.6397
OCSVM	0.7827	0.7560	0.8345	**0.6930**	0.5523	**0.8121**
DAE	0.3886	0.3641	0.3411	0.6213	0.6131	0.6053
BL (GFHF)	0.4244	0.4107	0.4244	0.6483	0.6483	0.6424
BL (LLGC)	0.1388	0.1567	0.1567	0.5123	0.5031	0.5123

linhas correspondem aos resultados obtidos com o modelo de referência binário (BL), o qual possui notícias reais e falsas rotuladas no conjunto de treinamento, considerando GFHF e LLGC como algoritmos de propagação de rótulos. Em cinza estão destacados os melhores resultados considerando as abordagens OCL e PUL, as coleções de notícias, o conjunto inicialmente rotulado e as medidas F_1.

Se considerarmos apenas redes homogêneas, uma primeira percepção é de que no geral, PU-LP apresenta comportamento muito próximo ao modelo de referência binário, mesmo

Tabela 7 – Resultados das bases de dados derivadas da coleção FakeNewsCorpus usando Doc2Vec como modelo de representação. PULP-FND denota a abordagem proposta, com redes heterogêneas, e PU-LP denota o uso de redes homogêneas. 10%, 20% e 30% indicam a porcentagem de notícias falsas rotuladas no conjunto de treinamento. Para o algoritmo de referência (BL), o conjunto rotulado possui a mesma porcentagem de notícias reais. Os melhores resultados dos algoritmos PUL e OCL estão destacados em cinza.

	Fake F_1			**Macro F_1**		
	10%	**20%**	**30%**	**10%**	**20%**	**30%**
FakeNewsCorpus 0						
PULP-FND (LPHN)	0.7825	0.8149	0.8438	0.8051	0.8394	0.8694
PULP-FND (GNM)	0.8894	0.9056	**0.9056**	0.8956	0.9143	**0.9187**
PU-LP (GFHF)	0.8742	0.8754	0.8640	0.8773	0.8820	0.8789
PU-LP (LLGC)	**0.9012**	**0.9080**	0.9054	**0.9049**	**0.9159**	0.9176
RC-SVM	0.7804	0.7626	0.7428	0.4546	0.5504	0.6085
k-NND	0.7616	0.7561	0.7375	0.7423	0.7361	0.7192
k-Means	0.7625	0.7330	0.6981	0.7258	0.6883	0.6757
OCSVM	0.8214	0.8534	0.8585	0.8077	0.8455	0.8552
DAE	0.7804	0.7626	0.7428	0.7652	0.7620	0.7540
BL (GFHF)	0.9102	0.9102	0.9057	0.9072	0.9072	0.9023
BL (LLGC)	0.9054	0.9054	0.9303	0.9122	0.9122	0.9307
FakeNewsCorpus 1						
PULP-FND (LPHN)	0.7502	0.8044	0.8238	0.7732	0.8280	0.8489
PULP-FND (GNM)	0.8826	**0.8748**	**0.8788**	0.8822	0.8860	**0.8965**
PU-LP (GFHF)	0.8677	0.8495	0.8511	0.8640	0.8621	0.8695
PU-LP (LLGC)	**0.8859**	0.8746	0.8758	**0.8843**	**0.8884**	0.8944
RC-SVM	0.6720	0.6298	0.6726	0.6508	0.6301	0.6451
k-NND	0.7498	0.7483	0.7389	0.7557	0.7548	0.7414
k-Means	0.7066	0.6709	0.6759	0.7387	0.6987	0.6803
OCSVM	0.7953	0.8444	0.8602	0.8079	0.8372	0.8581
DAE	0.7531	0.7378	0.7219	0.7403	0.7410	0.7443
BL (GFHF)	0.8834	0.8834	0.8799	0.8801	0.8801	0.8736
BL (LLGC)	0.8140	0.8140	0.7297	0.8510	0.8510	0.7931
FakeNewsCorpus 2						
PULP-FND (LPHN)	0.7449	0.7667	0.8010	0.7712	0.7991	0.8378
PULP-FND (GNM)	0.8718	0.8916	0.8971	0.8805	0.9038	0.9128
PU-LP (GFHF)	0.8746	0.8826	0.8731	0.8811	0.8944	0.8893
PU-LP (LLGC)	**0.8813**	**0.8949**	**0.8989**	**0.8886**	**0.9062**	**0.9138**
RC-SVM	0.5836	0.5836	0.6241	0.5144	0.5507	0.6485
k-NND	0.7294	0.7296	0.7158	0.7329	0.9262	0.7107
k-Means	0.6915	0.6499	0.6612	0.7298	0.6495	0.6149
OCSVM	0.7955	0.8462	0.8616	0.8054	0.8358	0.8565
DAE	0.7270	0.7107	0.6914	0.7006	0.7054	0.6981
BL (GFHF)	0.9056	0.9038	0.8989	0.9037	0.9021	0.8972
BL (LLGC)	0.8936	0.9040	0.9012	0.8848	0.8974	0.8951

usando metade dos dados rotulados, indicando que identificar notícias falsas confiáveis e inferir conjuntos de notícias reais usando apenas o conjunto inicialmente rotulado e índice de Katz pode ser uma estratégia promissora no contexto de classificação de notícias.

Considerando o modelo de representação BoW (ver Tabelas 4 e 5), e desconsiderando o algoritmo BL de referência, pode-se observar que a abordagem proposta PULP-FND que realiza propagação de rótulos com o algoritmo GNetMine (PULP-FND-GNM) se destaca para a maioria das bases de dados, tanto na medida F_1 *fake* quanto na F_1 macro. Exceções ocorrem para

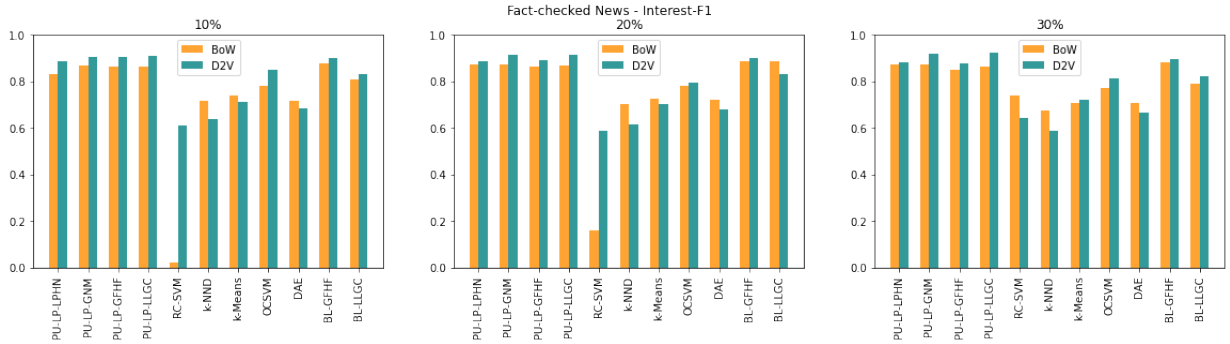
as coleções de notícias FakeNewsNet e Fake.BR. Quanto a Fake.BR, a versão homogênea de PU-LP se destaca para 10% de dados rotulados e RC-SVM apresenta melhor F_1 macro para 30% de dados rotulados. Já a coleção FakeNewsNet apresenta melhor F_1 *fake* com *k*-NND, enquanto DAE se destaca em F_1 macro.

Considerando Doc2Vec como modelo de representação (ver Tabelas 6 e 7), PULP-FND e PU-LP se destacam entre os melhores, em especial com os algoritmos de progagação LLGC e GNetMine. Exceções ocorrem com as bases de dados Fake.BR e FakeNewsNet. Na base Fake.BR o algoritmo OCSVM se destaca com 10% de dados rotulados. Na base FakeNewsNet algoritmos baseados em agrupamento e densidade apresentam melhores resultados.

As discussões dos resultados são guiadas pelas seguintes questões de pesquisa: i) Como o modelo de representação pode influenciar no comportamento dos resultados? ii) Qual grupo de algoritmos se destaca na detecção de notícias falsas? iii) Qual abordagem apresenta melhor desempenho geral? iv) O número de notícias falsas rotuladas aumenta significativamente o desempenho da abordagem? As respostas são apresentadas a seguir.

Para uma análise mais aprofundada a respeito de como o modelo de representação de notícias pode influenciar no comportamento dos resultados, nas Figuras 9-14 são apresentados gráficos de barras que comparam a F_1 *fake* obtida usando os modelos BoW (em laranja) e D2V (em verde). A análise é realizada para cada base de dados, considerando 10%, 20% e 30% de notícias falsas rotuladas.

Figura 9 – Comparação dos modelos de representação BoW (laranja) e D2V (verde) para a coleção Fact-checked news, considerando a *fake* F_1 (*Interest-*F_1) dos modelos OCL, PUL e binário (modelo de referência), com 10%, 20% e 30% de notícias falsas rotuladas.



Além dos gráficos de barras, a biblioteca *t*-SNE do *Sklearn* (PEDREGOSA *et al.*, 2011) foi utilizada para fornecer uma melhor percepção de como as notícias de cada base de dados estão distribuídas no espaço de características. T-SNE converte similaridades entre instâncias de dados em probabilidades conjuntas, tentando minimizar a divergência de Kullback-Leibler (MAATEN; HINTON, 2008) entre probabilidades conjuntas de *embeddings* de baixa dimensão e dados de alta dimensão. Foram utilizados os seguintes parâmetros: número de componentes = 2, inicialização da *embedding* com PCA e perplexidade = 3. Visualizações em duas dimensões

Figura 10 – Comparação dos modelos de representação BoW (laranja) e D2V (verde) para a coleção Fake.BR, considerando a F_1 *fake* (*Interest-*F_1) dos modelos OCL, PUL e binário (modelo de referência), com 10%, 20% e 30% de notícias falsas rotuladas.

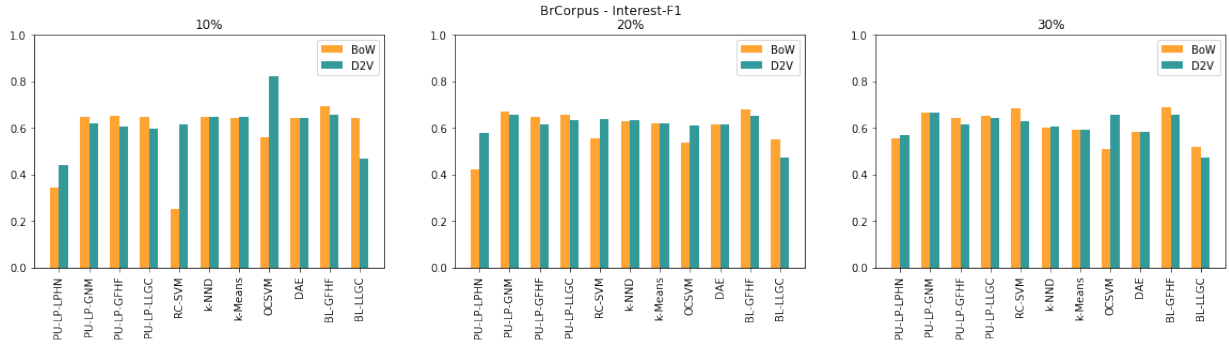


Figura 11 – Comparação dos modelos de representação BoW (laranja) e D2V (verde) para a coleção FakeNewsNet, considerando a *fake* F_1 (*Interest-*F_1) dos modelos OCL, PUL e binário (modelo de referência), com 10%, 20% e 30% de notícias falsas rotuladas.

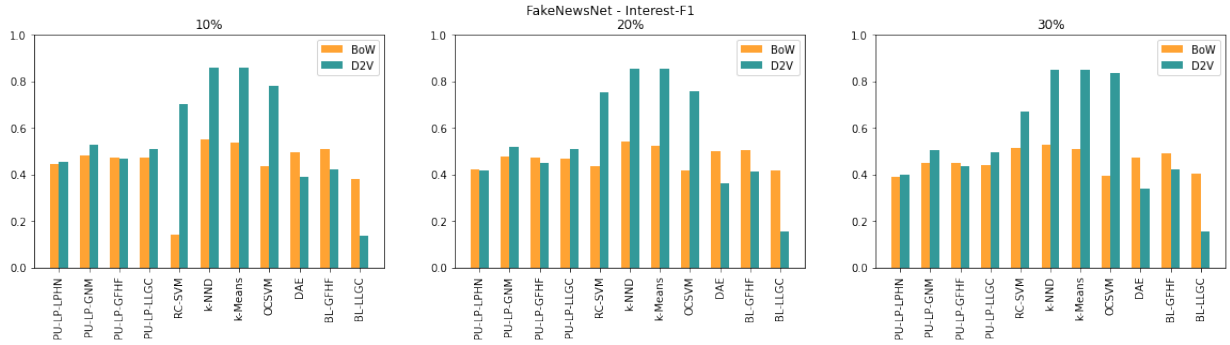
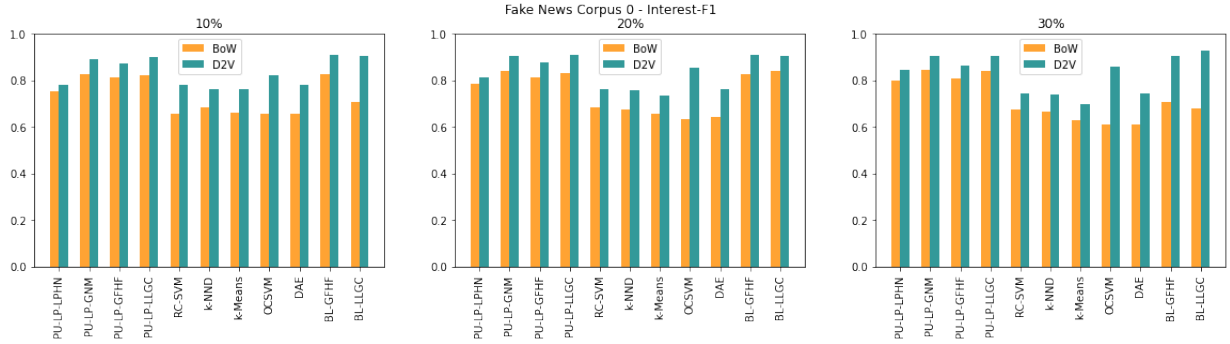


Figura 12 – Comparação dos modelos de representação BoW (laranja) e D2V (verde) para a coleção FakeNewsCorpus 0, considerando a *fake* F_1 (*Interest-*F_1) dos modelos OCL, PUL e binário (modelo de referência), com 10%, 20% e 30% de notícias falsas rotuladas.



são apresentadas nas Figuras 15-28. Nas figuras são apresentadas as distribuições do modelo de representação BoW, seguido das quatro representações geradas com D2V.

Para a base Fact-checked news, a maioria dos algoritmos apresenta melhor F_1 com DV2. Nas Figuras 15 e 16 pode-se observar que D2V divide notícias reais e falsas de forma eficiente no espaço de características, em especial na representação 4, na qual notícias falsas estão concentradas juntas na região inferior. Este cenário favorece os algoritmos PUL RC-SVM, PU-LP e PULP-FND. RC-SVM aumenta sua *fake* F_1 em mais de 50% para 10 e 20% de dados

Figura 13 – Comparação dos modelos de representação BoW (laranja) e D2V (verde) para a coleção FakeNewsCorpus 1, considerando a *fake* F_1 (*Interest-*F_1) dos modelos OCL, PUL e binário (modelo de referência), com 10%, 20% e 30% de notícias falsas rotuladas.

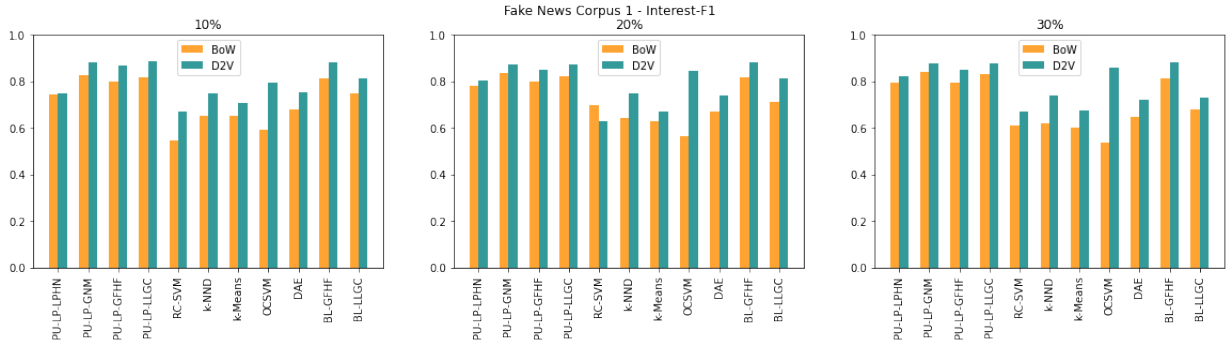
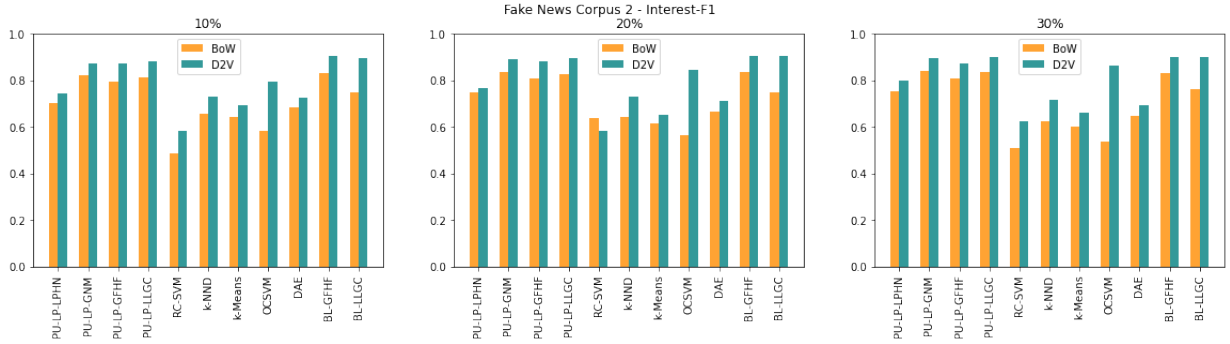


Figura 14 – Comparação dos modelos de representação BoW (laranja) e D2V (verde) para a coleção FakeNewsCorpus 2, considerando a *fake* F_1 (*Interest-*F_1) dos modelos OCL, PUL e binário (modelo de referência), com 10%, 20% e 30% de notícias falsas rotuladas.



rotulados. Isto se justifica por RC-SVM desempenhar melhor quando a classe negativa cobre uma grande região do espaço de características. Para PU-LP, o algoritmo é capaz de inferir conjuntos de interesse e não interesse confiáveis mais puros, favorecendo também a etapa de propagação de rótulos.

Para Fake.BR, que é balanceada e possui 6 diferentes assuntos, dos quais política (58%), celebridades (21.4%) e sociedade (17.7%) juntas compõem 97.1% da base de notícias, D2V também proporcionou um crescimento abrupto da *fake* F_1 para os algoritmos RC-SVM e OCSVM, considerando 10% de notícias rotuladas. A intuição é que D2V foi capaz de agrupar notícias no espaço de características considerando tanto a veracidade quanto os assuntos. Tal intuição é reforçada pelas Figuras 18 e 20. Na primeira (representação BoW), notícias de diferentes assuntos estão espalhadas pelo espaço de características, enquanto na segunda (representações geradas com D2V) estas parecem estar agrupadas.

A base FakeNewsNet apresenta uma dificuldade extra por conter notícias do domínio de celebridades, as quais tanto conteúdos reais quanto falsos possuem termos sensacionalistas, tornando a detecção de veracidade difícil até mesmo para humanos. Embora BoW tenha provido melhores resultados para o modelo de referência binário, a *fake* F_1 aumenta drasticamente

Figura 15 – Notícias da coleção Fact-checked plotadas em duas dimensões com a ferramenta t-SNE considerando o modelo de representação Bag-of-Words. Pontos em verde representam notícias reais e pontos em laranja representam notícias falsas.

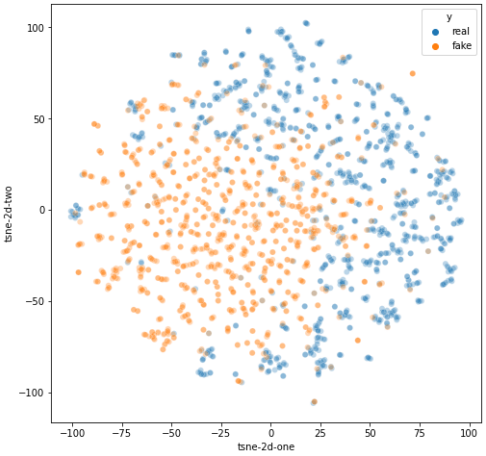


Figura 16 – Notícias da coleção Fact-checked New plotadas em duas dimensões com a ferramenta t-SNE considerando os modelos de representação Doc2Vec. Pontos em verde representam notícias reais e pontos em laranja representam notícias falsas.

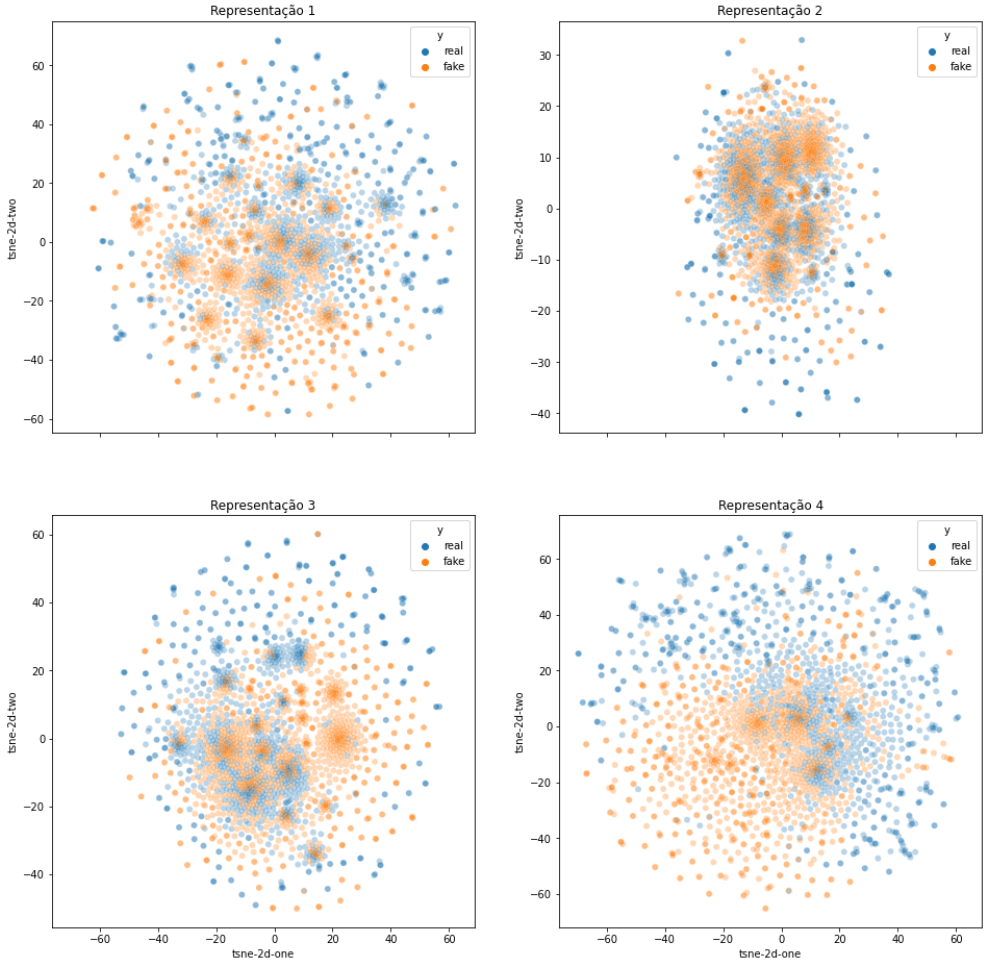


Figura 17 – Notícias da coleção Fake.BR plotadas em duas dimensões com a ferramenta t-SNE considerando o modelo de representação Bag-of-Words. Pontos em verde representam notícias reais e pontos em laranja representam notícias falsas.

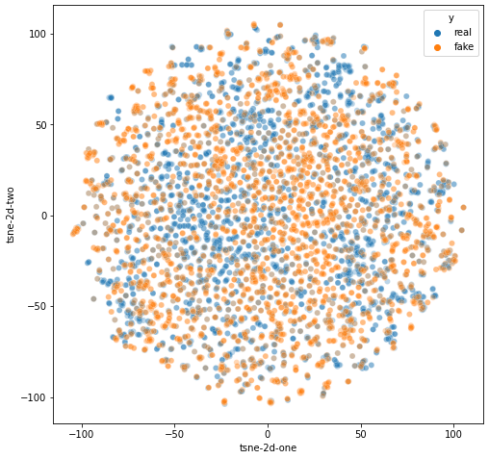
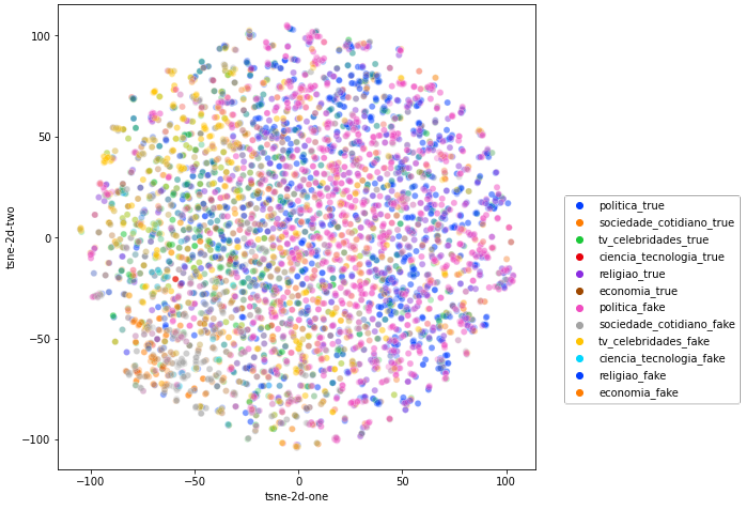


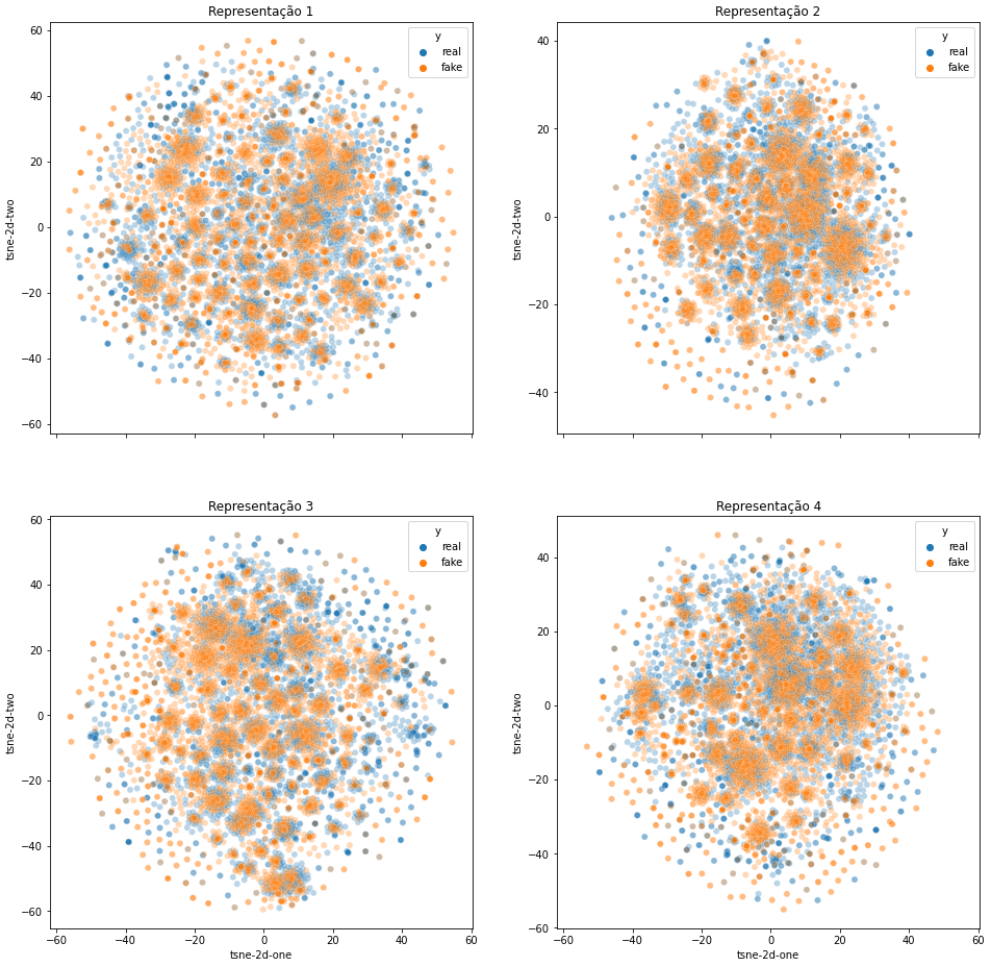
Figura 18 – Notícias da coleção Fake.BR plotadas em duas dimensões com a ferramenta t-SNE considerando o modelo de representação Bag-of-Words. As diferentes cores denotam como notícias reais e falsas de diferentes categorias estão organizadas no espaço de características.



quando *k*-NND, *k*-Means, OCSVM e RC-SVM utilizam D2V (ver Tabela 6), atingindo mais de 80% de *fake* F_1 mesmo em um cenário de extremo desbalanceamento. Nas Figuras 21 e 22 é possível perceber que D2V agrupa fake news de forma mais eficiente no espaço de características, favorecendo tanto o desempenho de algoritmos baseados em densidade e agrupamento, como a inferência do hiperplano de separação de OCSVM. Além disso, o fato de 75.7% das notícias em FakeNewsNet serem reais possibilitou que RC-SVM inferisse um conjunto mais puro de outliers, favorecendo seu desempenho.

Para as bases de dados derivadas da coleção FakeNewsCorpus, Doc2Vec supera Bag-of-Words em unanimidade (Figuras 12-14). Além disso, nas Figuras 23-28 tem-se a percepção de que D2V (em especial as representações 3 e 4) tende a separar de maneira mais efetiva notícias reais e falsas no espaço de características. A partir desta discussão, pode-se concluir que representar notícias com D2V é uma estratégia mais promissora.

Figura 19 – Notícias da coleção Fake.BR plotadas em duas dimensões com a ferramenta t-SNE considerando o modelo de representação Doc2Vec. Pontos em verde representam notícias reais e pontos em laranja representam notícias falsas.



Para uma análise aprofundada a respeito de qual grupo de algoritmos (OCL e PUL) se destaca na detecção de notícias falsas, foram plotados a *fake* F_1 (Figuras 30 e 30) e F_1 macro (Figuras 31 e 32) considerando as coleções de notícias, o modelo de representação D2V e o tipo de algoritmo. Barras em tons avermelhados correspondem a algoritmos PUL, barras em tons verdes correspondem a algoritmos OCL e barras amarelas correspondem ao modelo de referência binário. As anotações 1, 2 e 3 acima das barras representam a quantidade de *folds* usadas no treinamento dos algoritmos (10, 20 e 30% de dados rotulados, respectivamente).

No geral, OCL supera PUL apenas para a base de dados FakeNewsNet e considerando a medida *fake* F_1. Como discutido anteriormente, os algoritmos baseados em densidade e agrupamento (K-Means e k-NND) ultrapassam o desempenho até mesmo do modelo de referência em mais de 30% de *fake* F_1 usando o modelo de representação D2V. A hipótese para o pobre desempenho de abordagens PUL na base FakeNewsNet consiste de que notícias falsas estão agrupadas em diferentes regiões do espaço de características (ver Figuras 21 e 22). Neste caso, notícias podem ter sido incluídas incorretamente na inferência de conjuntos de interesse e não interesse confiáveis. Apesar do pobre desempenho de PUL nesta base de dados, pode-se observar

Figura 20 – Notícias da coleção Fake.BR plotadas em duas dimensões com a ferramenta t-SNE considerando o modelo de representação Doc2Vec. As diferentes cores denotam como notícias reais e falsas de diferentes categorias estão organizadas no espaço de características.

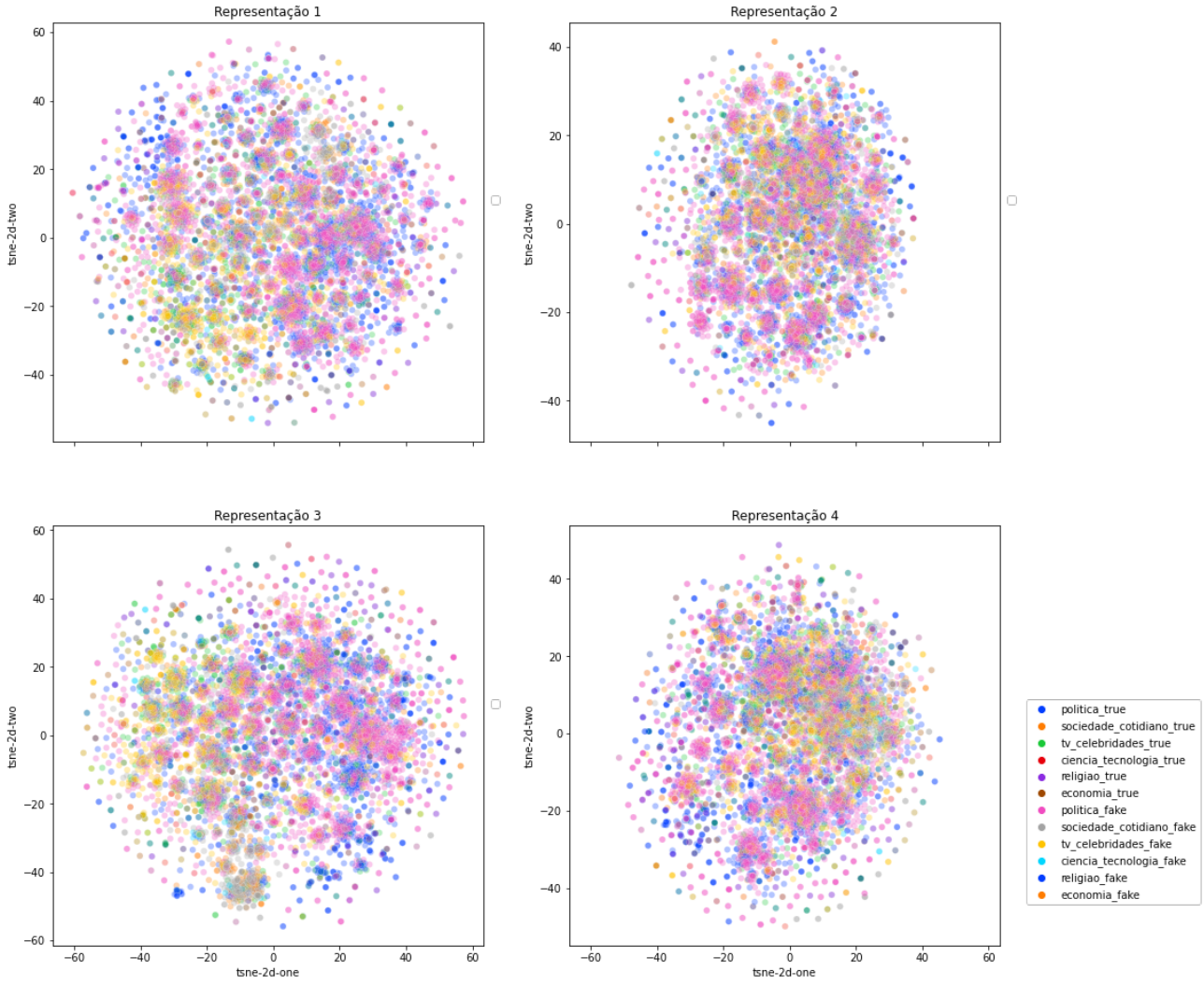


Figura 21 – Notícias da coleção FakeNewsNet plotadas em duas dimensões com a ferramenta t-SNE considerando o modelo de representação Bag-of-Words. Pontos em verde representam notícias reais e pontos em laranja representam notícias falsas.

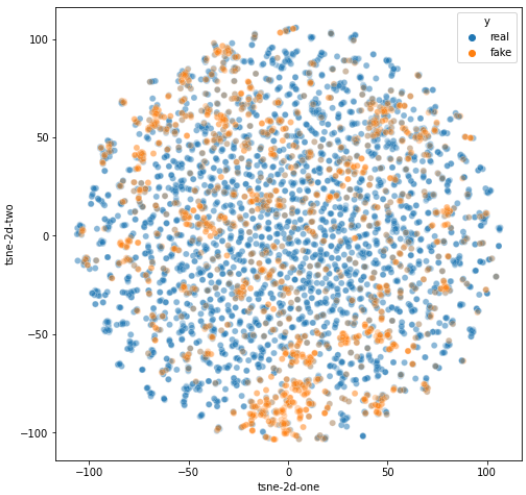


Figura 22 – Notícias da coleção FakeNewsNet plotadas em duas dimensões com a ferramenta t-SNE considerando o modelo de representação Doc2Vec. Pontos em verde representam notícias reais e pontos em laranja representam notícias falsas.

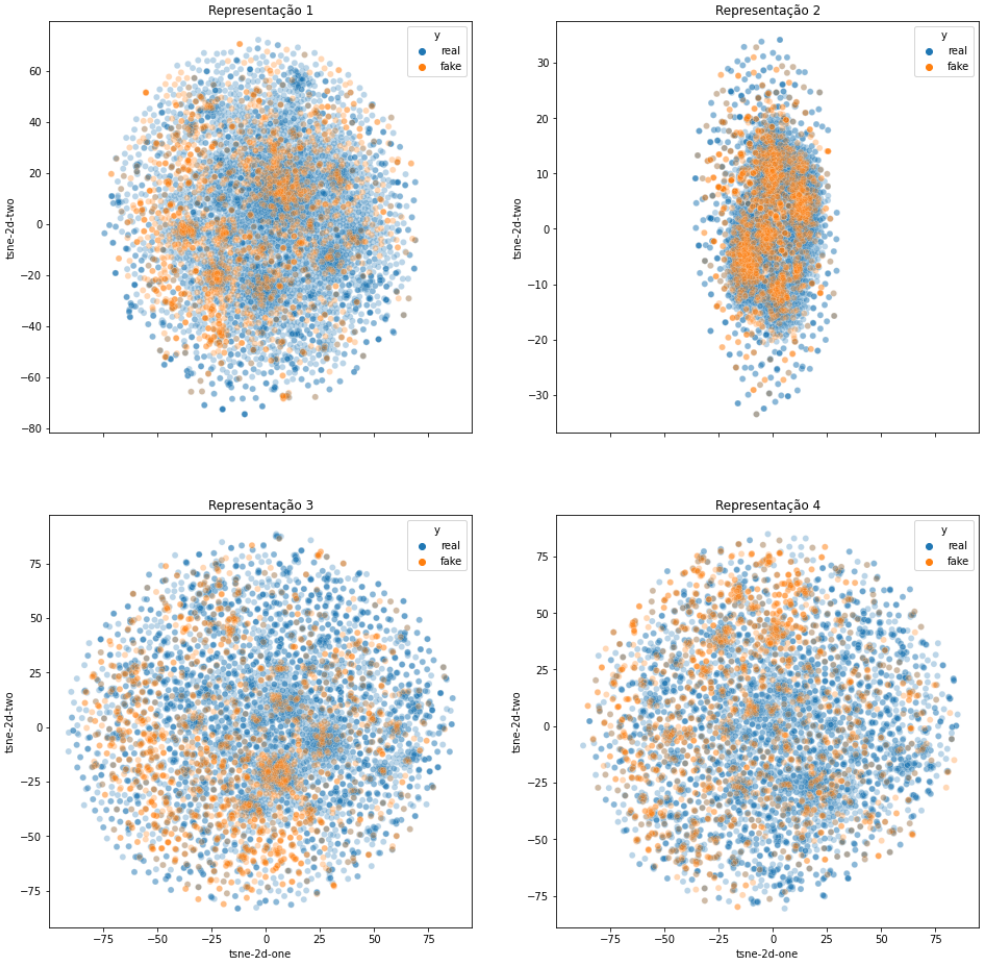


Figura 23 – Notícias da coleção FakeNewsCorpus 0 plotadas em duas dimensões com a ferramenta t-SNE considerando o modelo de representação Bag-of-Words. Pontos em verde representam notícias reais e pontos em laranja representam notícias falsas.

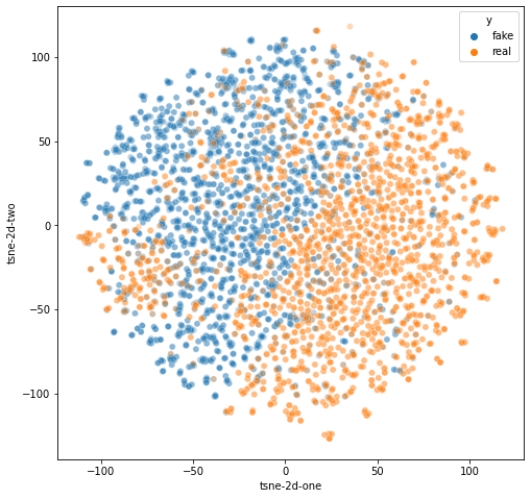


Figura 24 – Notícias da coleção FakeNewsCorpus 0 New plotadas em duas dimensões com a ferramenta t-SNE considerando o modelo de representação Doc2Vec. Pontos em verde representam notícias reais e pontos em laranja representam notícias falsas.

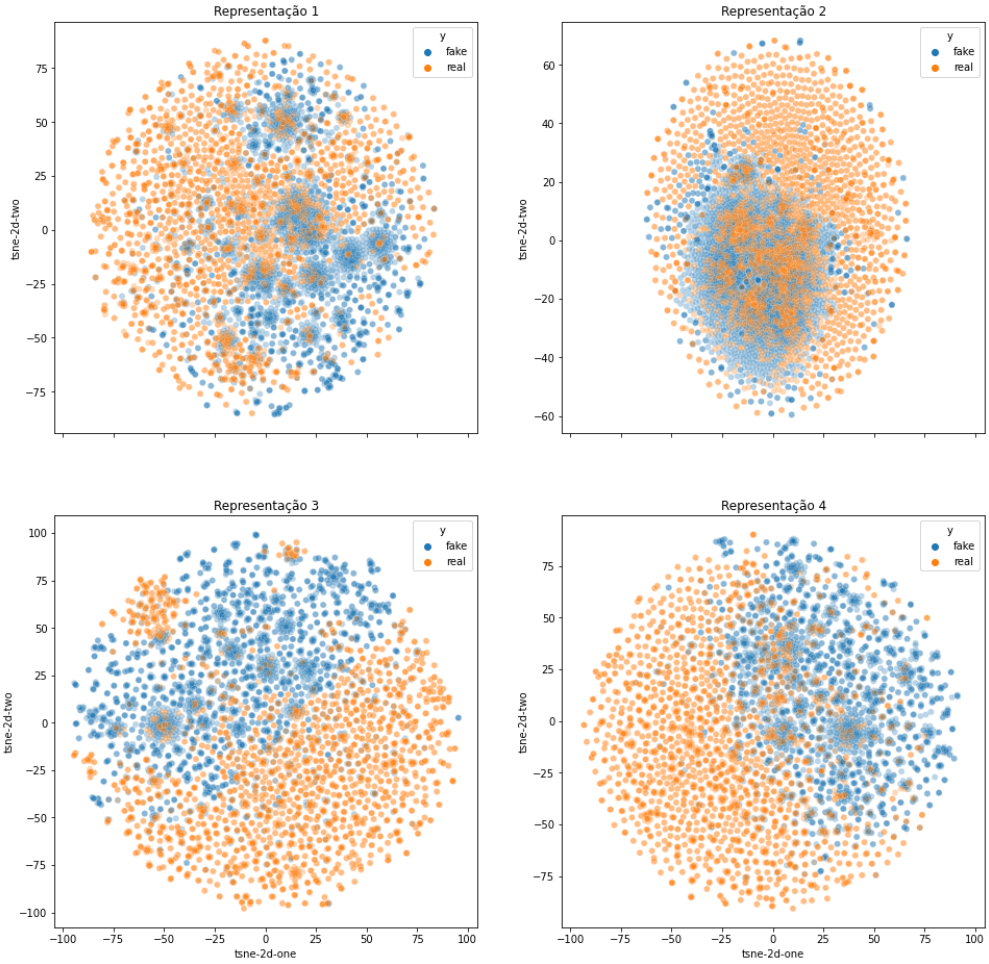


Figura 25 – Notícias da coleção FakeNewsCorpus 1 plotadas em duas dimensões com a ferramenta t-SNE considerando o modelo de representação Bag-of-Words. Pontos em verde representam notícias reais e pontos em laranja representam notícias falsas.

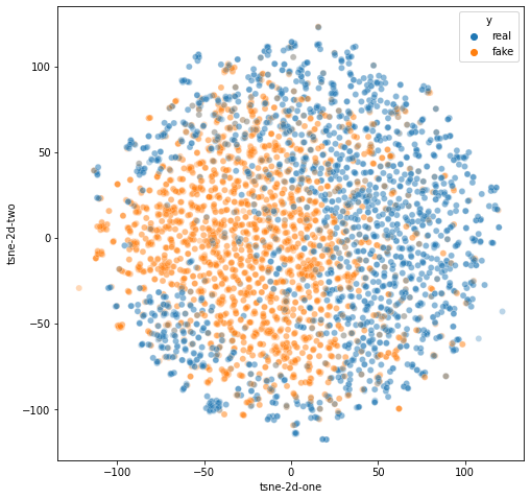


Figura 26 – Notícias da coleção FakeNewsCorpus 1 plotadas em duas dimensões com a ferramenta t-SNE considerando o modelo de representação Doc2Vec. Pontos em verde representam notícias reais e pontos em laranja representam notícias falsas.

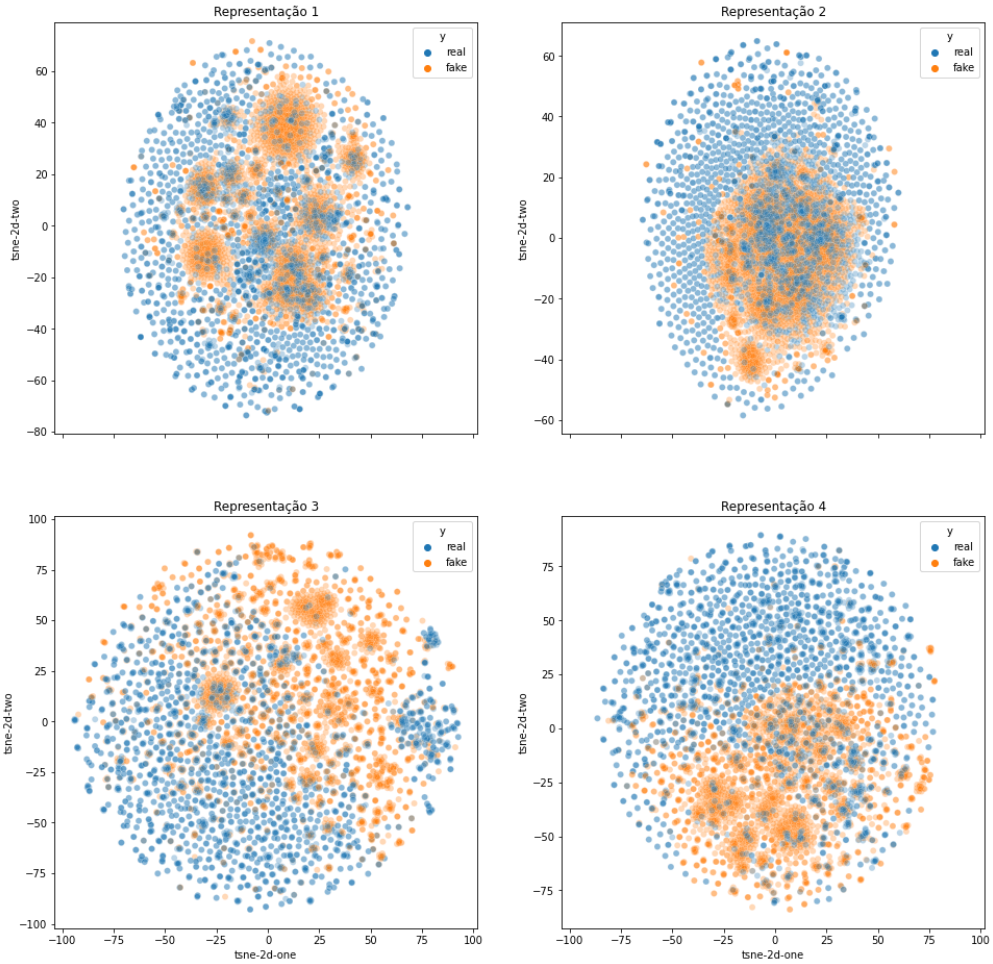


Figura 27 – Notícias da coleção FakeNewsCorpus 2 plotadas em duas dimensões com a ferramenta t-SNE considerando o modelo de representação Bag-of-Words. Pontos em verde representam notícias reais e pontos em laranja representam notícias falsas.

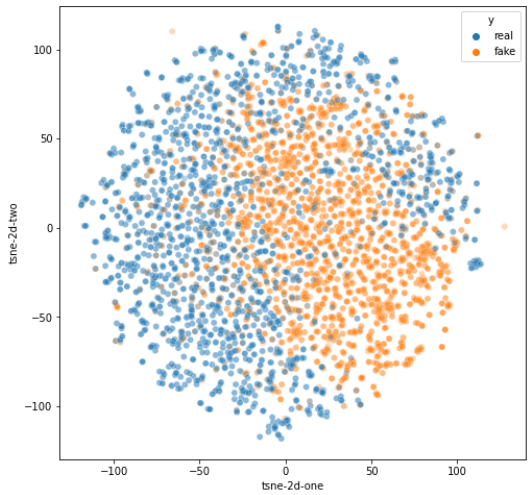
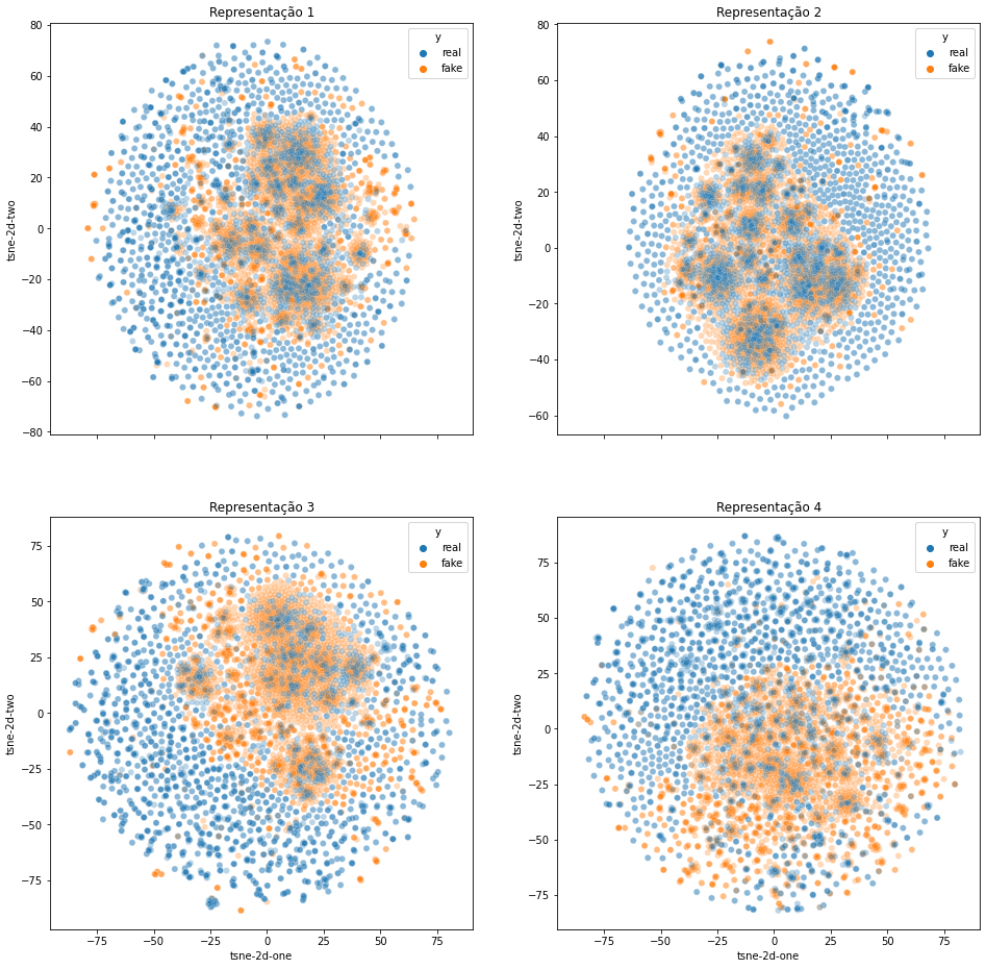


Figura 28 – Notícias da coleção FakeNewsCorpus 2 plotadas em duas dimensões com a ferramenta t-SNE considerando o modelo de representação Doc2Vec. Pontos em verde representam notícias reais e pontos em laranja representam notícias falsas.



que a abordagem PULP-FND que inclui termos na rede de notícias supera os resultados tanto da abordagem homogênea quanto do modelo de referência (ver Tabelas 6 e 7).

No geral, PUL supera OCL tanto para *fake* F_1 quanto para a F_1 macro na base Fake.BR. Além disso, PULP-FND (GNM) e OCSVM superam o modelo de referência binário. Nesta base de dados, que possui fake news de diferentes tópicos, a inclusão de termos em redes de notícias de PU-LP proporciona melhoria nos resultados. Neste caso, pode-se inferir que a presença de diferentes tipos de padrões na rede contribuiu para discriminar notícias falsas quando estas estavam distribuídas em diferentes regiões do espaço de características. É importante mencionar que FAUSTINI; COVÕES (2019) também realizaram experimentos com a base Fake.BR, atingindo *fake* F_1 de 67% com 90% de notícias falsas rotuladas. Os resultados atingidos pela abordagem proposta superam os de Faustini, et al. com apenas 20% de dados rotulados. Para aumentar o desempenho desta base de dados, talvez seja necessário garantir que o conjunto inicialmente rotulado possua notícias dos seis tópicos, ou realizar a classificação considerando um tópico por vez, já que os erros de classificação são principalmente associados com notícias pertencentes a tópicos com pouca representatividade (SILVA *et al.*, 2020).

Figura 29 – F_1 *fake* (interest-F_1) considerando o modelo de representação D2V e as bases de dados FakeNewsNet, Fake.BR e Fact-checked News. Barras em tons avermelhados, verdes e amarelas correspondem respectivamente a algoritmos PUL, OCL e o modelo de referência. Números 1, 2, e 3 representam o número de *folds* usados para treinar os algoritmos.

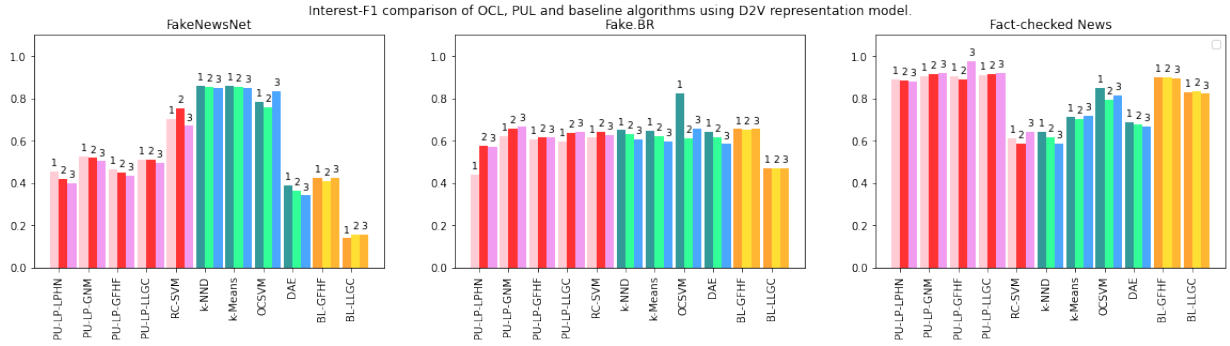
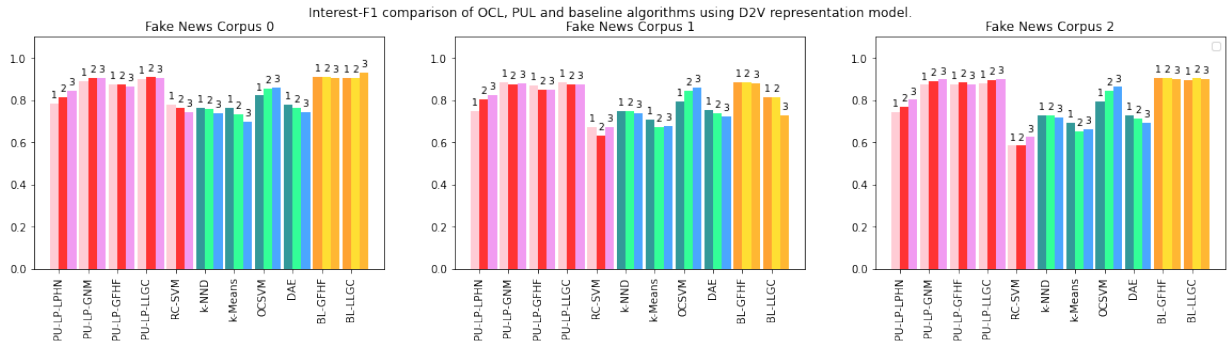


Figura 30 – F_1 *fake* (interest-F_1) considerando o modelo de representação D2V e as três bases de dados derivadas da coleção FakeNewsCorpus. Barras em tons avermelhados, verdes e amarelas correspondem respectivamente a algoritmos PUL, OCL e o modelo de referência. Números 1, 2, e 3 representam o número de *folds* usados para treinar os algoritmos.



PUL também supera abordagens OCL para Fact-checked News e para as bases derivadas de FakeNewsCorpus. PU-LP e PULP-FND apresentam melhores desempenhos gerais, além de estarem próximos aos resultados do modelo de referência, atingindo de 87% a 93% de F_1 macro e *fake*. Para estas quatro coleções é importante destacar que o modelo de representação D2V realiza uma boa separação de notícias reais e falsas no espaço de características. Desta forma, a inclusão de termos na rede de notícias não causa alteração significativa nos resultados.

Portanto, pode-se concluir que algoritmos OCL baseados em densidade e agrupamento são mais vantajosos quando notícias falsas estão agrupadas em diferentes regiões do espaço de características e possuem pouca representatividade na base de dados. Por outro lado, abordagens PUL podem inferir conjuntos mais puros de notícias falsas e reais confiáveis e obter resultados similares ou superiores a algoritmos binários semissupervisionados utilizando modelos de representação adequados.

Para uma análise de qual abordagem apresenta melhor desempenho geral, nas Tabelas 8 e 9 são apresentados os ranqueamentos médios dos algoritmos considerando a *fake* F_1 e macro para as representações BoW e D2V. 10%, 20% e 30% indicam as porcentagens de dados rotulados.

Figura 31 – F_1 macro considerando o modelo de representação D2V e as bases de dados FakeNewsNet, Fake.BR e Fact-checked News. Barras em tons avermelhados, verdes e amarelas correspondem respectivamente a algoritmos PUL, OCL e o modelo de referência. Números 1, 2, e 3 representam o número de *folds* usados para treinar os algoritmos.

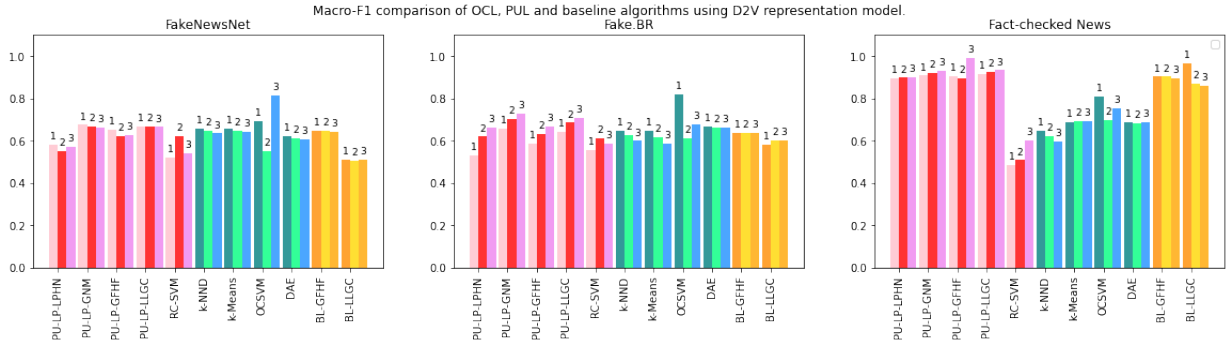
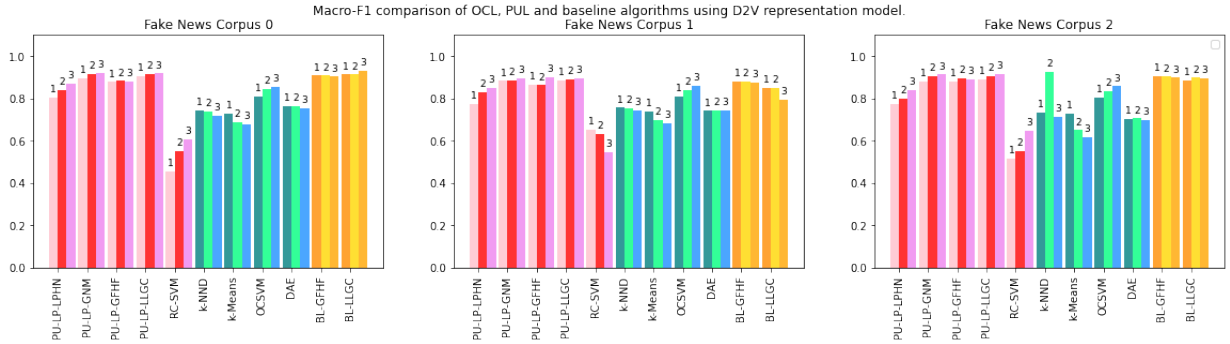


Figura 32 – F_1 macro considerando o modelo de representação D2V e as três bases de dados derivadas da coleção FakeNewsCorpus. Barras em tons avermelhados, verdes e amarelas correspondem respectivamente a algoritmos PUL, OCL e o modelo de referência. Números 1, 2, e 3 representam o número de *folds* usados para treinar os algoritmos.



A última coluna apresenta a média dos ranqueamentos médios. Os melhores desempenhos considerando PUL e OCL estão destacados em cinza.

Quanto a *fake* F_1, tanto para BoW quanto para D2V a abordagem proposta se destaca, considerando o algoritmo de propagação de rótulos GNetMine. A única exceção ocorre para 10% de dados rotulados e representação D2V, na qual PU-LP homogêneo apresenta melhor desempenho geral. Considerando a média dos ranqueamentos, o desempenho de PULP-FND com GNM fica muito próximo ao modelo de referência binário.

Quanto a F_1 macro, a abordagem PULP-FND também supera os demais algoritmos com BoW, cuja média dos ranqueamentos fica igual ao modelo de referência. Para notícias representadas com D2V, pode-se observar que o ranqueamento médio do algoritmo PULP-FND (em especial usando GNM como algoritmo de propagação) é muito próximo ao do PU-LP, que se destaca com LLGC como algoritmo de propagação de rótulos.

Com isso, pode-se concluir que entre as abordagens avaliadas destaca-se PULP-FND. Quando o modelo de representação é capaz de separar notícias reais e falsas no espaço de características, PU-LP homogêneo apresenta melhores resultados em comparação a versão

Tabela 8 – Ranqueamento médio e desvio padrão dos algoritmos OCL, PUL e binário (BIN), considerando 10%, 20% e 30% de notícias falsas rotuladas e resultados de *fake* F_1. Na última coluna é apresentada a média dos ranqueamentos médios. O melhor desempenho considerando PUL e OCL são destacados em cinza, desconsiderando os algoritmos de referência semissupervisionados binários.

			Fake F_1			
Rep. Model	Type	Algorithm	10%	20%	30%	Avg ± Std
BoW	PUL	PULP-FND (LPHN)	6.5 ± 2.1	6.8 ± 2.6	6.2 ± 3.2	6.5 ± 2.6
		PULP-FND (GNM)	**2.5 ± 1.4**	**2.3 ± 1.5**	**2.5 ± 2.3**	**2.4 ± 1.7**
		PU-LP (GFHF)	3.7 ± 1.4	4.8 ± 1.0	4.7 ± 1.0	4.4 ± 1.1
		PU-LP (LLGC)	4.7 ± 1.6	3.8 ± 1.8	3.7 ± 2.3	4.1 ± 1.9
		RC-SVM	10.9 ± 0.2	8.3 ± 1.5	6.5 ± 3.7	8.6 ± 1.8
	OCL	k-NND	6.2 ± 3.1	6.8 ± 3.3	7.0 ± 3.3	6.7 ± 3.2
		k-Means	7.0 ± 2.7	7.5 ± 3.1	7.8 ± 2.6	7.4 ± 2.8
		OCSVM	9.3 ± 1.3	10.2 ± 1.6	10.0 ± 1.5	9.8 ± 1.5
		DAE	7.3 ± 2.1	7.5 ± 2.1	7.8 ± 1.9	7.5 ± 2.0
	BIN	BL (GFHF)	1.5 ± 0.8	2.3 ± 1.2	2.8 ± 1.6	2.2 ± 1.2
		BL (LLGC)	6.5 ± 2.2	5.5 ± 3.6	7.0 ± 2.0	6.3 ± 2.6
D2V	PUL	PULP-FND (LPHN)	7.7 ± 2.0	7.3 ± 1.6	7.2 ± 2.1	7.4 ± 1.9
		PULP-FND (GNM)	4.3 ± 1.2	**2.8 ± 1.5**	**2.8 ± 1.5**	**3.3 ± 1.4**
		PU-LP (GFHF)	5.0 ± 2.2	5.3 ± 1.4	5.5 ± 0.8	5.3 ± 1.5
		PU-LP (LLGC)	**3.8 ± 3.1**	3.2 ± 1.7	3.4 ± 1.7	3.5 ± 2.2
		RC-SVM	8.8 ± 2.9	8.1 ± 3.7	8.3 ± 3.1	8.4 ± 3.2
	OCL	k-NND	7.2 ± 3.8	7.0 ± 3.5	7.3 ± 3.5	7.2 ± 3.6
		k-Means	7.2 ± 3.8	7.8 ± 3.4	8.2 ± 3.3	7.7 ± 3.5
		OCSVM	4.7 ± 2.2	6.0 ± 2.0	4.7 ± 2.0	5.1 ± 2.1
		DAE	8.1 ± 1.8	8.9 ± 0.7	9.1 ± 0.5	8.7 ± 1.0
	BIN	BL (GFHF)	3.2 ± 3.1	3.0 ± 3.0	3.3 ± 2.4	3.2 ± 2.8
		BL (LLGC)	6.2 ± 3.9	6.5 ± 3.9	6.3 ± 4.5.0	6.3 ± 4.1

heterogênea. Quando notícias falsas estão distribuídas no espaço de características, a inclusão de termos na rede de notícias aumenta o desempenho de classificação.

Analisando ainda as Figuras 29 e 30 podemos concluir que, em geral, os resultados não apresentaram grande variação considerando o número de *folds* do conjunto de treinamento. Até mesmo algoritmos como OCSVM atingiram máxima F_1 macro e *fake* F_1 para Fake.BR usando apenas 10% de dados rotulados.

Os resultados atingidos encorajam a pesquisa por abordagens de classificação capazes de detectar notícias falsas a partir de poucos dados rotulados.

4.3 Avaliação de PULP-FND Considerando a Inclusão de Características Linguísticas na Rede Heterogênea

Considerando que PULP é um algoritmo baseado em redes, novas características foram avaliadas para serem incluídas na rede $\mathscr{N}$ de notícias e termos, que demonstraram efetividade na literatura para discriminação de notícias reais e falsas (SILVA *et al.*, 2020; VARGAS; PARDO, 2020). As características são listadas a seguir:

- Incerteza = *Número total de verbos modais e voz passiva*

Tabela 9 – Ranqueamento médio e desvio padrão dos algoritmos OCL, PUL e binário (BIN), considerando 10%, 20% e 30% de notícias falsas rotuladas e resultados de F_1 macro. Na última coluna é apresentada a média dos ranqueamentos médios. O melhor desempenho considerando PUL e OCL são destacados em cinza.

Macro F_1						
Rep. Model	Type	Algorithm	10%	20%	30%	Avg ± Std
BoW	PUL	PULP-FND (LPHN)	7.0 ± 2.4	6.7 ± 3.3	5.5 ± 3.4	6.4 ± 3.0
		PULP-FND (GNM)	**2.5 ± 1.9**	**2.5 ± 2.5**	**2.2 ± 2.4**	**2.4 ± 2.3**
		PU-LP (GFHF)	4.8 ± 1.7	5.2 ± 1.0	5.2 ± 0.8	5.1 ± 1.2
		PU-LP (LLGC)	4.0 ± 2.1	4.0 ± 2.2	3.7 ± 2.7	3.9 ± 2.3
		RC-SVM	10.5 ± 1.2	8.8 ± 1.5	7.0 ± 2.8	8.8 ± 1.8
	OCL	k-NND	6.4 ± 1.4	6.9 ± 1.0	7.2 ± 2.2	6.8 ± 1.5
		k-Means	6.8 ± 2.2	7.8 ± 1.9	8.2 ± 2.2	7.6 ± 2.1
		OCSVM	9.3 ± 0.8	10.2 ± 1.0	10.3 ± 1.2	9.9 ± 1.0
		DAE	8.3 ± 3.2	7.8 ± 3.8	7.7 ± 3.9	7.9 ± 3.6
	BIN	BL (GFHF)	1.7 ± 0.8	2.5 ± 1.0	3.0 ± 0.6	2.4 ± 0.8
		BL (LLGC)	4.7 ± 1.9	3.7 ± 2.0	6.2 ± 1.2	4.9 ± 1.7
D2V	PUL	PULP-FND (LPHN)	7.7 ± 2.1	7.2 ± 1.9	6.0 ± 2.0	7.0 ± 2.0
		PULP-FND (GNM)	3.0 ± 1.3	2.0 ± 0.6	**1.8 ± 0.8**	2.3 ± 0.9
		PU-LP (GFHF)	5.0 ± 1.8	5.2 ± 0.8	5.0 ± 1.1	5.1 ± 1.2
		PU-LP (LLGC)	**2.7 ± 1.9**	**1.3 ± 0.5**	**1.8 ± 0.8**	**1.9 ± 1.1**
		RC-SVM	10.7 ± 0.5	10.0 ± 1.7	10.3 ± 0.5	10.3 ± 0.9
	OCL	k-NND	7.2 ± 2.6	6.3 ± 3.4	8.7 ± 1.6	7.4 ± 2.5
		k-Means	8.0 ± 2.4	8.5 ± 2.0	9.2 ± 2.3	8.6 ± 2.2
		OCSVM	4.5 ± 2.7	7.5 ± 1.6	4.8 ± 2.4	5.6 ± 2.2
		DAE	7.5 ± 2.8	7.7 ± 2.3	8.0 ± 1.1	7.7 ± 2.1
	BIN	BL (GFHF)	4.0 ± 2.5	3.5 ± 0.5	4.2 ± 1.5	3.9 ± 1.5
		BL (LLGC)	5.8 ± 3.7	6.8 ± 3.4	6.2 ± 3.4	6.3 ± 3.5

- Não imediatismo = *Número total de pronomes pessoais na 1ª e 2ª pessoa do singular*
- Emotividade = $\frac{\textit{Número total de adjetivos + número total de advérbios}}{\textit{número total de sujeitos + número total de verbos}}$
- Pausalidade = $\frac{\textit{Número total de sinais de pontuação}}{\textit{Número total de sentenças}}$
- Média de palavras por sentença = $\frac{\textit{Número total de palavras}}{\textit{Número total de sentenças}}$

Entre as características consideradas, foram avaliadas quais delas possuíam maiores correlações com o atributo alvo, e quais se destacavam em mais de uma base de dados. As características selecionadas a serem incluídas na rede foram: pausalidade, emotividade e número médio de palavras por sentença. Na Tabela 10 são apresentadas os valores de correlações. As novas características foram incluídas na rede de notícias $\mathcal{N}$ como novos nós não rotulados f_j, $0 < j <$ *número total de características*. Novas arestas entre notícias e as características foram adicionadas, cujo peso w_{d_i,f_j} corresponde ao valor normalizado da característica f_j para a notícia d_i, conforme a Equação 4.2. Após a propagação de rótulos, os nós de características incluídos também terão valores de classificação associados aos rótulos do problema.

$$w_{d_i,f_j} = \frac{w_{d_i,f_j}}{\max_{\mathbf{d}_k \in \mathscr{D}} w_{d_k,f_j}} \tag{4.2}$$

Tabela 10 – Correlações das principais características linguísticas considerando todas as bases de dados.

	Fact-checked News	Fake.BR	FakeNewsNet	FakeNewsCorpus 0	FakeNewsCorpus 1	FakeNewsCorpus 2
Emotividade	0.1749	0.1314	0.0899	0.0696	0.1504	0.0390
Pausalidade	-0.2851	-0.4733	-0.0224	-0.4727	-0.4889	-0.4719
Nº médio de palavras p/ sentença	-0.1888	-0.5833	0.0039	-0.3273	-0.2247	-0.3204

Doze redes distintas foram criadas, considerando a combinação de termos relevantes e novas características propostas. Na Tabela 11 são apresentadas as redes e características incluídas em cada uma. A rede 1, que contém apenas notícias, é homogênea, enquanto as demais são heterogêneas.

Tabela 11 – Lista de características incluídas em cada uma das doze redes heterogêneas propostas.

Característica/Rede	1	2	3	4	5	6	7	8	9	10	11	12
Notícias	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Termos representativos		✓				✓	✓	✓	✓	✓	✓	✓
Emotividade			✓			✓			✓	✓		✓
Pausalidade				✓			✓		✓		✓	✓
Média de palavras por sentença					✓			✓		✓	✓	✓

Após a construção da rede, a próxima etapa consiste na propagação de rótulos por meio de um algoritmo de aprendizado transdutivo. Nesta etapa foi considerado apenas o algoritmo GNetMine, que apresentou melhor desempenho em relação ao LPHN na grande maioria dos casos de teste.

4.3.1 Configuração Experimental e Critérios de Avaliação

Após a inclusão de todas as características na rede de notícias $\mathscr{N}$, foi realizada a normalização das relações da rede a fim de mitigar possíveis distorções devido aos diferentes intervalos de valores entre distintos tipos de relações. Assim, o peso da aresta de um objeto $o_i \in \mathscr{O}_l$, $o_j \in \mathscr{O}_m$, é dado por:

$$w_{o_i,o_j} = \frac{w_{o_i,o_j}}{\sum_{o_k} w_{o_i,o_k}}, o_i \in \mathscr{O}_l, o_j \in \mathscr{O}_m, \text{ and } o_k \in \mathscr{O}_m \tag{4.3}$$

Na análise experimental, foi considerado o algoritmo de referência de aprendizado semi-supervisionado binário (BL) para avaliar o processo de rotulação de exemplos de interesse e não interesse confiáveis em PU-LP. No algoritmo BL foram incluídas as mesmas características na rede em relação a PULP-FND, usando o algoritmo GNetMine para propagação de rótulos. A seleção de termos representativos, as configurações do algoritmo de propagação GNM, bem como as características do algoritmo BL foram as mesmas consideradas na Subseção 4.2.1.

4.3.2 Resultados e Discussões

Nas Tabelas 12 e 13 são apresentados os melhores resultados de F_1 da classe *fake* e F_1 macro atingidos para cada configuração de rede, utilizando BoW como modelo de representação.

Nas Tabelas 14 e 15 são apresentados os melhores resultados de F_1 e F_1 macro utilizando D2V como modelo de representação. Os resultados são divididos por base de dados (colunas). As primeiras 12 linhas correspondem aos resultados do algoritmo PULP-FND, e as 12 últimas correspondem aos resultados do modelo de referência binário. Redes 1 a 12 possuem as combinações de características propostas na Seção 4.3 (ver Tabela 11). 10%, 20%, e 30% indicam a porcentagem de notícias falsas usadas no conjunto de treinamento.

Nas Tabelas 16 e 17 são apresentadas análises de ranqueamento médio e desvio padrão das redes heterogêneas propostas, considerando respectivamente a *fake* F_1 e a F_1 macro.

Tabela 12 – F_1 de interesse das abordagens PULP-FND e modelo de referência usando Bag-of-Words como modelo de representação e GNetMine como algoritmo de propagação de rótulos.

	Bag-of-Words																	
	Fact-checked News			Fake.BR			FakeNewsNet			FakeNewsCorpus 0			FakeNewsCorpus 1			FakeNewsCorpus 2		
	10%	20%	30%	10%	20%	30%	10%	20%	30%	10%	20%	30%	10%	20%	30%	10%	20%	30%
Rede	**Positive and Unlabeled Learning by Label Propagation for Fake News Detection**																	
1	0.862	0.868	0.864	0.647	0.657	0.650	0.472	0.467	0.441	0.822	0.831	0.839	0.817	0.824	0.831	0.811	0.828	0.835
2	**0.869**	**0.875**	**0.875**	**0.650**	**0.670**	0.665	0.483	0.479	0.450	0.828	0.839	0.844	**0.827**	**0.835**	0.840	**0.820**	0.837	0.843
3	0.747	0.814	0.840	0.514	0.533	0.675	0.459	0.503	0.423	0.815	0.826	0.836	0.812	0.816	0.825	0.806	0.828	0.838
4	0.793	0.843	0.870	0.514	0.563	0.589	0.470	0.467	0.440	0.814	0.841	0.848	0.677	0.755	0.786	0.688	0.759	0.790
5	0.836	0.865	0.874	0.498	0.541	0.571	0.476	0.473	**0.451**	0.759	0.828	0.849	0.769	0.823	0.840	0.738	0.817	0.837
6	0.839	0.868	0.856	0.586	0.583	**0.676**	0.512	**0.512**	0.436	0.821	0.828	0.836	0.820	0.826	0.830	0.815	0.830	0.838
7	0.861	0.873	0.873	0.622	0.664	0.659	0.477	0.475	0.445	**0.848**	**0.866**	**0.869**	0.811	0.835	**0.845**	0.812	**0.842**	**0.850**
8	0.863	0.871	0.871	0.616	0.660	0.658	0.477	0.475	0.447	0.821	0.840	0.846	0.822	0.833	0.838	0.814	0.834	0.844
9	0.793	0.856	0.862	0.578	0.580	0.671	**0.514**	0.508	0.437	0.842	0.857	0.862	0.817	0.831	0.835	0.815	0.834	0.844
10	0.797	0.857	0.858	0.577	0.580	0.670	0.494	0.502	0.436	0.820	0.833	0.838	0.820	0.827	0.830	0.813	0.830	0.839
11	0.855	0.871	0.869	0.592	0.651	0.655	0.476	0.474	0.445	0.834	0.859	0.864	0.805	0.828	0.840	0.798	0.834	0.846
12	0.778	0.852	0.864	0.578	0.579	0.665	0.500	0.502	0.438	0.834	0.851	0.859	0.815	0.827	0.833	0.806	0.830	0.842
Rede	**Modelo de Referência with Semi-supervised Learning**																	
1	0.810	0.886	0.791	0.642	0.552	0.521	0.383	0.419	0.402	0.709	0.841	0.681	0.747	0.713	0.679	0.747	0.749	0.763
2	**0.892**	**0.912**	**0.920**	**0.768**	0.736	0.751	0.456	0.454	0.433	**0.847**	**0.864**	**0.872**	**0.845**	**0.865**	**0.876**	**0.859**	**0.874**	**0.883**
3	0.871	0.906	0.918	0.727	0.756	0.777	0.184	0.188	0.082	0.828	0.852	0.866	0.830	0.853	0.867	0.837	0.861	0.875
4	0.839	0.887	0.902	0.532	0.613	0.636	0.479	**0.526**	**0.512**	0.795	0.825	0.838	0.678	0.759	0.791	0.711	0.786	0.813
5	0.862	0.897	0.910	0.516	0.589	0.610	0.229	0.134	0.142	0.768	0.822	0.844	0.771	0.821	0.842	0.781	0.834	0.853
6	0.883	0.906	0.917	0.743	**0.764**	**0.783**	0.373	0.359	0.369	0.842	0.858	0.871	0.840	0.859	0.873	0.851	0.868	0.880
7	0.875	0.901	0.912	0.737	0.705	0.706	**0.524**	0.507	0.471	0.831	0.855	0.870	0.810	0.842	0.858	0.825	0.857	0.870
8	0.879	0.904	0.914	0.730	0.705	0.697	0.401	0.399	0.371	0.832	0.853	0.867	0.833	0.853	0.868	0.843	0.864	0.877
9	0.879	0.901	0.912	0.723	0.752	0.772	0.445	0.425	0.456	0.833	0.855	0.869	0.822	0.846	0.862	0.828	0.857	0.873
10	0.879	0.902	0.913	0.720	0.750	0.771	0.368	0.359	0.358	0.833	0.852	0.867	0.835	0.854	0.867	0.843	0.862	0.876
11	0.869	0.897	0.909	0.717	0.687	0.675	0.475	0.463	0.410	0.818	0.849	0.866	0.801	0.836	0.856	0.812	0.851	0.868
12	0.877	0.899	0.910	0.695	0.735	0.761	0.421	0.408	0.426	0.821	0.849	0.866	0.813	0.842	0.859	0.816	0.852	0.869

Apesar da proposta de inclusão de novas características na rede de notícias, a adição de termos (Rede 2) ainda apresenta destaque em relação as demais. Quanto ao PU-LP, a adição de termos tende a aumentar o desempenho, principalmente quando notícias são representadas com BoW (Tabelas 12-15). Nas Tabelas 16 e 17 também é possível observar que a Rede 2 apresentou melhor ranqueamento médio, com baixo desvio padrão em relação as demais redes propostas. Para BoW com 20% e 30% de dados rotulados, a inclusão adicional de pausalidade também contribuiu na melhoria de desempenho (SILVA *et al.*, 2020).

Apesar dos problemas já discutidos da base de dados FakeNewsNet, a abordagem PULP-FND ainda apresenta melhor *fake* F_1 em relação ao modelo de referência. Resultados inferiores também ocorrem na base de dados Fake.BR. No entanto, a adição de informações extras na rede heterogênea também proporciona melhorias nos resultados.

PU-LP gerou resultados que variam de 87 a 92% de *fake* F_1 e macro para as bases Fact-checked News e FakeNewsCorpus, com D2V como modelo de representação e adição

Tabela 13 – F_1 macro das abordagens PULP-FND e modelo de referência usando Bag-of-Words como modelo de representação e GNetMine como algoritmo de propagação de rótulos.

	Bag-of-Words																	
	Fact-checked News			Fake.BR			FakeNewsNet			FakeNewsCorpus O			FakeNewsCorpus 1			FakeNewsCorpus 2		
	10%	20%	30%	10%	20%	30%	10%	20%	30%	10%	20%	30%	10%	20%	30%	10%	20%	30%
Rede	**Positive and Unlabeled Learning by Label Propagation for Fake News Detection**																	
1	0.863	0.873	0.877	0.648	0.678	0.695	0.611	0.602	0.573	0.828	0.844	0.859	0.825	0.836	0.851	0.821	0.841	0.856
2	**0.870**	**0.882**	0.887	**0.665**	0.695	**0.711**	0.623	0.613	0.585	0.833	0.852	0.862	0.833	0.848	0.859	**0.829**	0.851	0.862
3	0.785	0.844	0.839	0.608	0.636	0.681	0.648	**0.679**	**0.692**	0.820	0.837	0.853	0.814	0.826	0.842	0.816	0.841	0.858
4	0.817	0.861	0.889	0.609	0.654	0.681	0.613	0.601	0.573	0.838	0.864	0.876	0.738	**0.803**	**0.833**	0.749	0.807	0.837
5	0.848	0.876	**0.889**	0.602	0.642	0.673	0.603	0.612	0.590	0.790	0.849	0.873	0.795	0.844	0.866	0.778	0.843	0.866
6	0.850	**0.882**	0.860	0.645	0.664	0.688	**0.666**	0.667	0.662	0.823	0.840	0.852	0.821	0.836	0.847	0.822	0.842	0.856
7	0.865	0.880	0.887	0.658	**0.696**	0.710	0.619	0.609	0.579	**0.857**	**0.878**	**0.887**	**0.822**	0.849	0.866	0.828	**0.858**	**0.873**
8	0.866	0.878	0.884	0.655	0.695	0.710	0.615	0.610	0.581	0.831	0.851	0.864	0.829	0.846	0.857	0.825	0.848	0.864
9	0.815	0.873	0.854	0.641	0.664	0.688	0.664	0.660	0.642	0.847	0.867	0.880	0.826	0.845	0.854	0.826	0.849	0.865
10	0.818	0.874	0.855	0.642	0.664	0.688	0.659	0.662	0.660	0.827	0.844	0.855	0.824	0.840	0.847	0.823	0.842	0.858
11	0.861	0.878	0.883	0.642	0.688	0.708	0.615	0.608	0.579	0.844	0.872	0.883	0.816	0.842	0.861	0.817	0.850	0.868
12	0.804	0.870	0.850	0.642	0.663	0.688	0.661	0.661	0.644	0.842	0.862	0.877	0.822	0.843	0.852	0.821	0.845	0.862
Rede	**Modelo de referência com abordagem de classificação semissupervisionada binária**																	
1	0.839	0.888	0.827	0.695	0.641	0.622	0.628	0.645	0.633	0.764	0.850	0.754	0.781	0.765	0.748	0.794	0.798	0.806
2	**0.891**	**0.911**	**0.919**	**0.778**	0.754	0.762	0.669	0.665	0.651	**0.852**	**0.868**	0.877	**0.850**	**0.869**	**0.880**	**0.864**	**0.879**	**0.888**
3	0.864	0.903	0.916	0.717	0.750	0.772	0.523	0.526	0.473	0.834	0.857	0.871	0.836	0.858	0.872	0.845	0.867	0.880
4	0.849	0.890	0.904	0.623	0.680	0.699	0.667	**0.698**	**0.688**	0.818	0.842	0.853	0.732	0.793	0.818	0.756	0.814	0.835
5	0.866	0.898	0.910	0.614	0.666	0.681	0.546	0.500	0.504	0.793	0.836	0.856	0.796	0.836	0.855	0.804	0.848	0.864
6	0.880	0.904	0.916	0.741	**0.763**	**0.782**	0.622	0.611	0.623	0.845	0.862	0.875	0.842	0.861	0.876	0.855	0.871	0.884
7	0.877	0.902	0.912	0.759	0.735	0.734	**0.700**	0.687	0.665	0.843	0.864	**0.877**	0.825	0.852	0.866	0.838	0.865	0.878
8	0.879	0.904	0.913	0.755	0.730	0.728	0.639	0.634	0.616	0.840	0.859	0.873	0.840	0.859	0.873	0.851	0.870	0.882
9	0.877	0.899	0.911	0.735	0.759	0.777	0.658	0.647	0.665	0.842	0.862	0.875	0.832	0.853	0.868	0.839	0.864	0.878
10	0.876	0.900	0.912	0.733	0.758	0.777	0.619	0.610	0.617	0.840	0.857	0.871	0.840	0.858	0.871	0.849	0.867	0.880
11	0.871	0.898	0.909	0.745	0.723	0.713	0.676	0.665	0.634	0.831	0.857	0.873	0.817	0.846	0.864	0.827	0.859	0.875
12	0.875	0.898	0.909	0.720	0.749	0.771	0.644	0.636	0.650	0.833	0.856	0.872	0.826	0.850	0.866	0.830	0.860	0.875

de termos à rede de notícias. Tais resultados são relevantes, em especial por considerarem um conjunto inicialmente pequeno de dados rotulados, diferente da maior parte das abordagens encontradas na literatura. Além disso, no modelo de referência binário, a inclusão de termos chega a aumentar a *fake* F_1 e macro substancialmente, o que indica que tal estratégia pode ser promissora para distinção de notícias reais e falsas no espaço de características.

4.4 Avaliação de PULP-FND Considerando Yake! para Inclusão de Termos na Rede Heterogênea

Os experimentos realizados pela autora (SOUZA *et al.*, 2021; SOUZA *et al.*, 2022) demonstraram um aumento do desempenho de classificação em PU-LP diante da inclusão de termos selecionados usando uma estratégia de BoW considerando unigramas e bigramas, na qual termos com valor de tf-idf acima de um limiar ℓ foram calculados e incluídos na rede $\mathscr{N}$.

Para adicionar termos que melhor caracterizem o conteúdo expresso nas notícias, foi proposta uma representação alternativa à BoW, que seleciona termos por meio de uma ferramenta específica de extração de palavras-chave (*keywords*). Yake! (CAMPOS *et al.*, 2020) usa estatísticas locais dos documentos para calcular a importância de termos, não condicionando sua relevância ao número de vezes em que ele ocorre no documento, além de ser uma ferramenta competitiva quando comparada com outras disponíveis na literatura (PISKORSKI *et al.*, 2021). Yake! recebe como entrada um texto e os seguintes parâmetros:

Tabela 14 – F_1 de interesse dos algoritmos PULP-FND e modelo de referência binário considerando o modelo de representação Doc2Vec e GNetMine para propagação de rótulos.

	Doc2Vec																	
	Fact-checked News			Fake.BR			FakeNewsNet			FakeNewsCorpus O			FakeNewsCorpus 1			FakeNewsCorpus 2		
	10%	20%	30%	10%	20%	30%	10%	20%	30%	10%	20%	30%	10%	20%	30%	10%	20%	30%
Rede	**Positive and Unlabeled Learning by Label Propagation for Fake News Detection**																	
1	**0.911**	**0.916**	**0.921**	0.598	0.636	0.642	0.511	0.508	0.496	**0.901**	**0.908**	0.905	**0.886**	0.875	0.876	**0.881**	**0.895**	**0.899**
2	0.904	0.913	**0.921**	**0.622**	**0.659**	**0.667**	**0.527**	0.518	0.503	0.889	0.906	0.906	0.883	0.875	0.879	0.872	0.892	0.897
3	0.798	0.855	0.834	0.491	0.528	0.634	0.479	0.519	0.521	0.881	0.894	0.895	0.867	0.862	0.865	0.854	0.877	0.885
4	0.814	0.878	0.901	0.536	0.563	0.570	0.493	0.485	0.472	0.860	0.876	0.884	0.777	0.846	0.877	0.788	0.850	0.873
5	0.848	0.893	0.907	0.526	0.546	0.555	0.491	0.499	0.490	0.861	0.889	0.895	0.868	**0.877**	0.881	0.859	0.871	0.867
6	0.870	0.904	0.868	0.562	0.576	0.634	0.519	**0.529**	**0.531**	0.881	0.896	0.896	0.875	0.867	0.871	0.863	0.880	0.889
7	0.883	0.901	0.914	0.596	0.637	0.643	0.512	0.497	0.479	0.893	0.907	**0.909**	0.870	0.875	**0.884**	0.863	0.876	0.886
8	0.888	0.902	0.914	0.591	0.633	0.641	0.517	0.503	0.481	0.878	0.896	0.900	0.875	0.872	0.875	0.864	0.879	0.888
9	0.842	0.894	0.858	0.551	0.571	0.634	0.515	0.515	0.516	0.885	0.895	0.901	0.872	0.871	0.875	0.862	0.872	0.883
10	0.843	0.895	0.859	0.550	0.570	0.634	0.508	0.517	0.525	0.878	0.894	0.895	0.875	0.868	0.870	0.862	0.875	0.885
11	0.875	0.897	0.910	0.577	0.623	0.634	0.511	0.497	0.475	0.881	0.900	0.904	0.866	0.873	0.881	0.858	0.868	0.882
12	0.828	0.889	0.855	0.549	0.568	0.634	0.507	0.511	0.513	0.880	0.891	0.897	0.872	0.870	0.875	0.860	0.866	0.879
Rede	**Modelo de Referência with Semi-supervised Learning**																	
1	0.831	0.833	0.822	0.470	0.471	0.471	0.139	0.157	0.157	0.905	0.905	0.930	0.814	0.814	0.730	0.894	0.904	0.901
2	**0.929**	**0.932**	**0.940**	0.716	0.739	0.727	0.292	0.304	0.320	**0.931**	**0.929**	**0.939**	**0.922**	**0.924**	**0.932**	**0.908**	**0.922**	**0.929**
3	0.859	0.909	0.931	0.721	0.738	0.755	0.160	0.081	0.043	0.921	0.921	0.931	0.907	0.907	0.919	0.893	0.907	0.917
4	0.854	0.893	0.913	0.570	0.570	0.589	0.408	**0.443**	**0.459**	0.887	0.887	0.898	0.848	0.848	0.877	0.779	0.853	0.880
5	0.876	0.906	0.921	0.544	0.544	0.558	0.133	0.063	0.029	0.905	0.905	0.920	0.900	0.900	0.914	0.876	0.899	0.913
6	0.917	0.921	0.936	**0.741**	**0.748**	**0.758**	0.307	0.332	0.315	0.922	0.921	0.932	0.910	0.911	0.924	0.898	0.911	0.921
7	0.911	0.914	0.929	0.683	0.690	0.711	**0.463**	0.452	0.447	0.917	0.916	0.930	0.903	0.905	0.918	0.885	0.905	0.918
8	0.913	0.917	0.930	0.644	0.682	0.674	0.292	0.317	0.310	0.919	0.918	0.929	0.909	0.910	0.922	0.896	0.909	0.921
9	0.915	0.918	0.932	0.705	0.725	0.726	0.422	0.409	0.413	0.915	0.914	0.928	0.903	0.904	0.916	0.886	0.902	0.915
10	0.915	0.919	0.932	0.697	0.720	0.719	0.334	0.347	0.322	0.917	0.916	0.927	0.906	0.907	0.918	0.894	0.905	0.916
11	0.906	0.911	0.926	0.694	0.663	0.719	0.416	0.403	0.406	0.911	0.909	0.925	0.898	0.899	0.913	0.875	0.897	0.911
12	0.912	0.917	0.930	0.658	0.687	0.690	0.405	0.394	0.396	0.911	0.910	0.924	0.886	0.906	0.922	0.878	0.897	0.911

- Tamanho da janela deslizante w usada no cálculo de características estatísticas;
- O número de n-gramas desejado;
- O algoritmo e um limiar usado para desduplicação;
- A língua do texto é necessária para identificação da lista específica de palavras-chave.

Como saída, o algoritmo gera uma lista ordenada de palavras-chave de forma que quanto menor a pontuação, maior a relevância do termo. Portanto, para a inserção de uma nova conexão na rede entre o documento d_i e o termo t_j, foi necessário normalizar os pesos das conexões. A normalização foi feita conforme a Equação 4.4.

$$w_{d_i,t_j} = \begin{cases} 0, & \text{se } score_{t_j} > 1 \\ (1 - (\frac{score_{t_j}}{1+score_{t_j}})), & \text{caso contrário} \end{cases} \tag{4.4}$$

Em ambas as abordagens foram mantidos termos que apresentam uma conexão com pelo menos dois documentos. Em seguida, toda a rede foi normalizada conforme a Equação 4.1, a fim de mitigar possíveis distorções devido a diferentes intervalos de valores entre tipos de conexões distintos. A rede normalizada foi usada como entrada do algoritmo de propagação de rótulos GNetMine.

Tabela 15 – F_1 macro das abordagens PULP-FND e modelo de referência binário com o modelo de representação Doc2Vec, usando GNetMine como algoritmo de propagação de rótulos.

	Doc2Vec																	
	Fact-checked News			Fake.BR			FakeNewsNet			FakeNewsCorpus O			FakeNewsCorpus 1			FakeNewsCorpus 2		
	10%	20%	30%	10%	20%	30%	10%	20%	30%	10%	20%	30%	10%	20%	30%	10%	20%	30%
Rede	**Positive and Unlabeled Learning by Label Propagation for Fake News Detection**																	
1	**0.916**	**0.924**	**0.933**	0.640	0.685	0.709	0.667	0.668	0.665	**0.905**	**0.916**	0.918	**0.884**	**0.888**	0.894	**0.889**	**0.906**	**0.914**
2	0.908	0.920	0.931	**0.657**	**0.703**	**0.729**	0.675	0.667	0.662	0.896	0.914	0.919	0.882	0.886	0.896	0.881	0.904	0.913
3	0.827	0.878	0.868	0.599	0.636	0.686	0.666	**0.698**	**0.707**	0.886	0.903	0.909	0.864	0.876	0.884	0.865	0.891	0.903
4	0.837	0.893	0.917	0.619	0.658	0.678	0.642	0.636	0.633	0.873	0.891	0.902	0.807	0.867	**0.897**	0.815	0.870	0.892
5	0.862	0.904	0.921	0.615	0.650	0.670	0.661	0.663	0.658	0.868	0.896	0.908	0.870	0.883	0.893	0.862	0.876	0.892
6	0.880	0.915	0.893	0.633	0.663	0.688	**0.676**	0.690	0.701	0.887	0.905	0.910	0.874	0.878	0.888	0.870	0.894	0.906
7	0.890	0.910	0.926	0.644	0.690	0.716	0.651	0.641	0.633	0.897	0.914	**0.920**	0.873	0.881	0.896	0.864	0.892	0.905
8	0.894	0.910	0.926	0.642	0.689	0.715	0.660	0.650	0.637	0.885	0.906	0.914	0.876	0.878	0.892	0.866	0.892	0.906
9	0.859	0.907	0.886	0.631	0.662	0.687	0.672	0.671	0.683	0.887	0.905	0.914	0.874	0.875	0.889	0.864	0.887	0.902
10	0.859	0.908	0.886	0.632	0.662	0.687	0.668	0.679	0.695	0.885	0.904	0.909	0.875	0.877	0.887	0.865	0.889	0.903
11	0.883	0.907	0.922	0.632	0.681	0.710	0.651	0.642	0.630	0.885	0.907	0.915	0.870	0.878	0.893	0.860	0.885	0.901
12	0.847	0.904	0.883	0.632	0.661	0.687	0.666	0.669	0.682	0.882	0.902	0.912	0.873	0.875	0.886	0.862	0.882	0.898
Rede	**Modelo de referência com abordagem de classificação semissupervisionada binária**																	
1	0.865	0.866	0.859	0.583	0.600	0.600	0.512	0.503	0.512	0.912	0.912	0.931	0.851	0.851	0.793	0.885	0.897	0.895
2	**0.929**	**0.932**	0.940	0.734	0.756	0.753	0.585	0.590	0.595	**0.932**	**0.929**	**0.939**	**0.921**	**0.922**	**0.931**	**0.910**	**0.923**	**0.930**
3	0.899	0.926	**0.941**	0.723	0.746	0.767	0.499	0.467	0.449	0.922	0.922	0.932	0.904	0.904	0.916	0.896	0.909	0.919
4	0.863	0.897	0.916	0.647	0.647	0.663	0.635	0.656	**0.667**	0.894	0.894	0.904	0.861	0.861	0.885	0.805	0.865	0.887
5	0.880	0.908	0.923	0.633	0.633	0.646	0.499	0.464	0.446	0.907	0.907	0.921	0.902	0.902	0.915	0.879	0.901	0.914
6	0.906	0.927	0.939	**0.742**	**0.759**	**0.778**	0.589	0.594	0.584	0.923	0.922	0.933	0.908	0.909	0.922	0.899	0.911	0.922
7	0.912	0.916	0.930	0.720	0.722	0.742	**0.671**	**0.663**	0.640	0.920	0.919	0.931	0.904	0.907	0.919	0.888	0.907	0.919
8	0.914	0.918	0.930	0.690	0.718	0.723	0.584	0.596	0.587	0.921	0.920	0.931	0.909	0.909	0.921	0.896	0.909	0.920
9	0.902	0.921	0.934	0.729	0.747	0.767	0.605	0.639	0.647	0.917	0.916	0.929	0.904	0.904	0.916	0.888	0.903	0.915
10	0.902	0.922	0.934	0.727	0.746	0.766	0.593	0.603	0.587	0.918	0.918	0.929	0.906	0.906	0.917	0.894	0.906	0.916
11	0.907	0.912	0.927	0.727	0.703	0.747	0.646	0.641	0.617	0.914	0.912	0.927	0.899	0.900	0.914	0.879	0.898	0.912
12	0.900	0.918	0.931	0.708	0.733	0.757	0.599	0.632	0.636	0.913	0.912	0.926	0.901	0.901	0.913	0.881	0.898	0.911

Tabela 16 – Análise de ranqueamento médio e desvio padrão das redes heterogêneas propostas considerando a medida *fake* F_1.

	Análise de ranqueamento médio											
	PULP-FND						Modelo de Referência					
	BoW			D2V			BoW			D2V		
Net	10%	20%	30%	10%	20%	30%	10%	20%	30%	10%	20%	30%
1	5.5 ± 2.9	8.0 ± 2.8	8.5 ± 1.9	2.0 ± 2.0	2.8 ± 2.4	3.5 ± 2.5	10.7 ± 1.5	10.7 ± 2.4	11.2 ± 2.0	10.3 ± 2.7	10.5 ± 2.0	10.3 ± 2.7
2	$\mathbf{2.3 \pm 2.1}$	3.0 ± 2.7	4.0 ± 2.8	$\mathbf{1.8 \pm 0.7}$	$\mathbf{2.3 \pm 0.8}$	$\mathbf{2.8 \pm 1.8}$	$\mathbf{1.5 \pm 1.2}$	$\mathbf{2.2 \pm 1.8}$	$\mathbf{2.3 \pm 2.2}$	$\mathbf{2.6 \pm 3.0}$	$\mathbf{2.5 \pm 3.2}$	$\mathbf{2.3 \pm 2.4}$
3	10.2 ± 1.6	9.8 ± 3.5	9.7 ± 3.9	10.7 ± 1.9	8.3 ± 4.1	8.3 ± 3.9	7.0 ± 2.8	5.3 ± 3.2	5.5 ± 3.7	5.8 ± 3.4	5.7 ± 3.6	5.2 ± 3.1
4	11.2 ± 0.8	10.3 ± 2.7	8.8 ± 3.2	11.2 ± 0.9	11.5 ± 0.8	9.7 ± 2.9	9.7 ± 3.8	9.2 ± 4.0	9.2 ± 4.0	10.0 ± 3.0	9.7 ± 3.8	9.5 ± 4.2
5	10.3 ± 2.0	9.8 ± 1.5	5.5 ± 4.5	9.7 ± 1.6	8.3 ± 3.7	8.3 ± 3.9	10.7 ± 0.8	10.8 ± 1.0	10.2 ± 0.8	10.2 ± 1.5	10.3 ± 1.2	10.2 ± 1.5
6	4.7 ± 2.3	6.5 ± 3.3	8.5 ± 3.9	4.7 ± 1.5	4.7 ± 3.1	6.3 ± 3.5	3.2 ± 2.9	3.2 ± 3.4	3.2 ± 2.9	2.7 ± 2.2	2.8 ± 2.1	2.8 ± 2.6
7	5.0 ± 2.9	$\mathbf{2.7 \pm 2.7}$	$\mathbf{3.0 \pm 2.5}$	4.3 ± 2.1	4.7 ± 3.6	3.7 ± 3.4	5.3 ± 2.7	5.6 ± 2.5	5.7 ± 2.5	5.8 ± 2.6	5.3 ± 2.3	5.2 ± 2.3
8	4.2 ± 2.0	4.3 ± 1.7	5.2 ± 1.9	4.2 ± 2.4	5.3 ± 1.6	5.8 ± 2.2	4.3 ± 1.5	5.4 ± 2.0	5.2 ± 2.3	5.4 ± 2.7	5.3 ± 2.3	6.3 ± 2.5
9	4.5 ± 3.3	5.2 ± 2.8	5.5 ± 2.9	6.2 ± 1.9	6.9 ± 2.2	7.0 ± 2.2	5.2 ± 1.0	5.2 ± 1.5	4.7 ± 1.5	5.0 ± 2.0	5.7 ± 2.3	5.5 ± 2.1
10	6.5 ± 2.4	6.7 ± 1.2	8.5 ± 2.5	7.5 ± 1.8	7.4 ± 3.1	7.5 ± 3.1	5.3 ± 2.7	5.2 ± 2.1	5.5 ± 2.3	4.8 ± 1.2	4.8 ± 1.2	5.9 ± 1.7
11	7.0 ± 2.7	4.7 ± 2.4	4.5 ± 2.7	7.3 ± 2.6	7.0 ± 3.4	6.2 ± 3.1	7.8 ± 2.4	7.8 ± 2.4	8.7 ± 1.4	7.5 ± 2.7	8.5 ± 2.4	7.8 ± 2.4
12	6.7 ± 3.1	7.0 ± 2.5	6.3 ± 1.6	8.5 ± 1.0	8.3 ± 2.2	8.3 ± 2.2	7.3 ± 1.2	7.5 ± 1.0	6.8 ± 1.5	7.8 ± 1.9	6.8 ± 1.9	7.0 ± 2.8

4.4.1 Configuração Experimental e Critérios de Avaliação

A única mudança em relação à proposta anterior é a substituição de termos na rede, agora realizada por meio do algoritmo de extração de palavras-chave Yake!. Os parâmetros utilizados, escolhidos com base na avaliação experimental realizada em CAMPOS *et al.* (2020), são: algoritmo de desduplicação *Sequence Matcher* com limiar 0.9; tamanho de janela deslizante $w = 1$; e número de n-gramas = {1,3} (unigramas e trigramas).

Quanto ao número de palavras-chave k (*keywords*) selecionadas por documento, foram considerados quatro casos de teste, nos quais a inclusão de termos poderia ter (i) um valor fixo para todos os documentos, considerando uma porcentagem p do número médio de termos $\bar{M}$ da base de dados; ou (ii) um valor variável, considerando uma porcentagem p do número de termos x do documento atual. Se k é menor do que $p\%$ de $\bar{M}$, selecionam-se os melhores $p\%$ de

Tabela 17 – Análise de ranqueamento médio e desvio padrão das redes heterogêneas propostas considerando a medida F_1 macro.

	Análise de ranqueamento médio											
	PULP-FND						Modelo de Referência					
	BoW			D2V			BoW			D2V		
Net	10%	20%	30%	10%	20%	30%	10%	20%	30%	10%	20%	30%
1	6.3 ± 2.9	9.2 ± 2.2	8.5 ± 2.3	2.2 ± 1.8	**2.3 ± 2.2**	3.5 ± 2.5	10.7 ± 1.5	10.7 ± 2.4	11.2 ± 2	10.5 ± 1.5	11 ± 1.3	10.5 ± 2.3
2	**2.7 ± 2.6**	3.2 ± 2.2	4.8 ± 2.6	**2.0 ± 0.6**	2.7 ± 2.2	**2.8 ± 1.8**	**1.3 ± 0.8**	**2.0 ± 1.5**	**2.5 ± 2.1**	**2.3 ± 2.8**	**2.5 ± 3.2**	**2.8 ± 2.5**
3	9.8 ± 2.5	9.7 ± 4.3	9.0 ± 4.1	9.0 ± 3.0	8.2 ± 4.2	8.3 ± 3.9	8.0 ± 2.8	6.2 ± 2.8	6.0 ± 3.6	6.7 ± 3.3	5.2 ± 3.4	4.7 ± 3.5
4	9.7 ± 2.6	10.2 ± 3.1	8.8 ± 4.3	11.3 ± 0.8	11.5 ± 0.8	9.7 ± 2.9	10.0 ± 3.0	9.2 ± 4.0	9.2 ± 4	10.0 ± 3.5	9.7 ± 3.8	9.5 ± 4.2
5	10.8 ± 1.9	7.3 ± 2.1	5.0 ± 4.0	9.5 ± 1.9	9.0 ± 3.2	8.3 ± 3.9	10.5 ± 1.0	10.7 ± 1.2	10 ± 1.1	10.2 ± 1.2	10.0 ± 1.7	10 ± 1.7
6	6.0 ± 3.0	6.5 ± 3.8	8.0 ± 3.6	4.2 ± 1.8	4.3 ± 2	6.3 ± 3.5	3.7 ± 2.9	3.3 ± 2.9	3.2 ± 2.6	3.3 ± 2.3	3.0 ± 2.5	3.2 ± 2.9
7	3.5 ± 2.4	**2.7 ± 3.2**	**3.3 ± 3.5**	5.7 ± 3.6	5.0 ± 3.2	3.7 ± 3.4	4.2 ± 2.8	4.8 ± 2.3	5 ± 2.5	4.5 ± 2.2	5.0 ± 2.4	5.3 ± 2.2
8	4.5 ± 2.9	5.0 ± 2.1	5.3 ± 2.1	5.0 ± 2.8	5.2 ± 2.1	5.8 ± 2.2	4.0 ± 1.7	5.0 ± 2.4	5.7 ± 2.9	4.8 ± 3.3	5.3 ± 2.0	5.8 ± 2.6
9	4.8 ± 3.7	5.3 ± 2.2	6 ± 2.6	6.3 ± 2.5	7.2 ± 2.3	6.9 ± 2.2	5.3 ± 0.8	5.0 ± 2.1	4.3 ± 2.3	5.7 ± 1.8	5.3 ± 1.6	5.0 ± 2.4
10	6.5 ± 2.1	7.2 ± 2.3	7.8 ± 2.6	6.3 ± 2.3	6.7 ± 2.0	7.4 ± 3.1	6.2 ± 2.2	5.8 ± 2.5	6 ± 2.2	5.3 ± 1.2	5.2 ± 1.0	5.7 ± 1.6
11	6.7 ± 2.4	5.2 ± 2.9	4.3 ± 2.7	8.0 ± 2.4	7.2 ± 2.3	7.0 ± 3.4	6.8 ± 3.1	7.7 ± 2.7	7.8 ± 1.8	6.7 ± 3.6	8.0 ± 2.5	8.0 ± 1.7
12	6.7 ± 2.9	6.7 ± 2.3	7.0 ± 2.8	8.5 ± 1.8	8.8 ± 1.9	8.3 ± 2.2	7.3 ± 0.8	7.7 ± 1.0	7.2 ± 1.9	8.0 ± 1.5	7.8 ± 2.1	7.5 ± 2.8

$\bar{M}$ termos. Esta estratégia foi utilizada para garantir que cada notícia da rede possua $p\%$ de $\bar{M}$ como número mínimo de conexões. Esta condição aumenta o número de relações até mesmo quando notícias possuem poucas palavras, favorecendo a etapa de propagação de rótulos. Na Tabela 3 é apresentado o valor de $\bar{M}$ usado em cada base de dados para calcular o parâmetro k. O valor total de termos x do documento também é calculado. São considerados $p = \{5, 10\}$. Portanto:

$$\text{keywords } k = \begin{cases} 5\% \text{ de } \bar{M}; \\ 10\% \text{ de } \bar{M}; \\ \text{se } 5\% \text{ de } x < \bar{M} \text{ então } k = 5\% \text{ de } \bar{M}, \text{ caso contrário } k = 5\% \text{ de } x; \\ \text{se } 10\% \text{ de } x < \bar{M} \text{ então } k = 10\% \text{ de } \bar{M}, \text{ caso contrário } k = 10\% \text{ de } x. \end{cases}$$

Nos experimentos foram considerados apenas os modelos de representação 3 e 4 gerados com D2V (Ver Subseção 4.2.1), os quais forneceram melhores desempenhos de F_1 macro e de interesse, e GNetMine como algoritmo de propagação de rótulos. Na próxima seção são apresentados resultados e discussões.

4.4.2 Resultados e Discussões

Na Tabela 18 são apresentados os resultados atingidos com PU-LP (rede de notícias, homogênea) e PULP-FND (notícias e termos, heterogênea). Quanto ao segundo caso, são comparadas redes quando termos são extraídos com BoW (abordagem anterior) e com Yake!. A partir dos resultados é possível observar que Yake! supera BoW tanto nas bases de dados em português Fact-checked News e Fake.BR quanto nas bases em inglês FakeNewsCorpus 0 e FakeNewsCorpus 1. Exceções ocorrem apenas para as bases FakeNewsNet e FakeNewsCorpus 2.

Quanto a base Fake.BR, cujas notícias estão distribuídas no espaço de características considerando a veracidade e assunto, a inclusão de palavras-chave com Yake! aumenta os resultados em 3% em relação a rede homogênea. Para Fact-checked news, FakeNewsCorpus 0, FakeNewsCorpus 1 e FakeNewsCorpus 1, os modelos de representação são capazes de separar

Tabela 18 – Comparação de resultados atingidos em cada base de dados considerando diferentes configurações de redes. As três primeiras colunas são referentes a medida *fake* F_1, enquanto as três ultimas são referentes a F_1 macro. 10%, 20% e 30% representam as porcentagens de notícias falsas inicialmente rotuladas. Os resultados da primeira linha são referentes a redes compostas por notícias. Na segunda linha são apresentados resultados referentes a redes de notícias e termos extraídos com BoW e tf-idf, e na terceira linha redes de notícias e termos extraídos com Yake!.

	Fake F_1			**Macro F_1**		
	10%	**20%**	**30%**	**10%**	**20%**	**30%**
Rede	**Fact-checked News**					
PU-LP	0.9110	0.9163	0.9214	0.9159	0.9243	0.9325
PULP-FND (BoW)	0.9036	0.9131	0.9206	0.9076	0.9205	0.9312
PULP-FND (Yake!)	**0.9166**	**0.9247**	**0.9333**	**0.9208**	**0.9327**	**0.9433**
	FaKe.BR					
pu-lp	0.5980	0.6358	0.6421	0.6397	0.6845	0.7086
PULP-FND (BoW)	0.6219	0.6586	0.6671	0.6570	0.7031	0.7289
PULP-FND (Yake!)	**0.6330**	**0.6693**	**0.6805**	**0.6609**	**0.7072**	**0.7355**
	FakeNewsNet					
PU-LP	0.5112	0.5083	0.4964	0.6669	0.6677	**0.6649**
PULP-FND (BoW)	**0.5267**	**0.5179**	**0.5027**	0.6748	0.6666	0.6623
PULP-FND (Yake!)	0.5215	0.5149	0.5004	**0.6774**	**0.6711**	0.6648
	FakeNewsCorpus 0					
PU-LP	0.9012	0.9080	0.9054	0.9049	0.9159	0.9176
PULP-FND (BoW)	0.8894	0.9056	0.9056	0.8956	0.9143	0.9187
PULP-FND (Yake!)	**0.9250**	**0.9310**	**0.9309**	**0.9287**	**0.9379**	**0.9409**
	FakeNewsCorpus 1					
PU-LP	0.8859	0.8746	0.8758	0.8843	0.8884	0.8944
PULP-FND (BoW)	0.8826	0.8748	0.8788	0.8822	0.8860	0.8965
PULP-FND (Yake!)	**0.8939**	**0.8888**	**0.8947**	**0.8958**	**0.9025**	**0.9119**
	FakeNewsCorpus 2					
PU-LP	**0.8813**	**0.8949**	**0.8989**	**0.8886**	**0.9062**	0.9138
PULP-FND (BoW)	0.8718	0.8916	0.8971	0.8805	0.9038	0.9128
PULP-FND (Yake!)	0.8732	0.8919	0.8961	0.8812	0.9057	**0.9141**

notícias reais e falsas no espaço de características, de forma que a inclusão de termos proporciona pouco ganho em relação à rede homogênea. No entanto, a utilização de Yake! ainda supera os resultados da abordagem tf-idf.

Na Tabela 19 é apresentada uma análise dos parâmetros que obtiveram melhores desempenhos diante da abordagem PULP-FND usando Yake!, considerando a média dos 10 *folds* da validação cruzada. Bases de dados em português obtiveram melhores resultados com o modelo de representação 4 de Doc2Vec (segunda linha da tabela, parâmetro "D2V Rep. model"), enquanto o modelo de representação 3 (rep. 3) se destacou para bases de dados em inglês. Além disso, resultados mais altos foram atingidos na inclusão de unigramas, bigramas e trigramas na rede de notícias (n-gram=3). A única exceção ocorreu com a base FakeNewsNet, que obteve melhor desempenho com a inclusão apenas de unigramas (n-gram=1). Ainda considerando a tabela, a maioria das bases de dados obteve resultados superiores considerando $k = 5\%$ de $\bar{M}$. Exceções ocorreram apenas nas bases FakeNewsNet e Fake.BR, cuja média geral de termos por documento $\bar{M}$ é menor em relação as demais bases de dados (183.63). Isto mostra que a inclusão de muitos

termos e consequentemente aumentar o número de relações da rede nem sempre resulta em melhoria de desempenho do algoritmo de propagação de rótulos.

O valor do parâmetro $\alpha = 0.5$ de GNetMine se destaca em todas as bases de dados. Este parâmetro se refere a confiança dos objetos rotulados no conjunto de treinamento, isto é, ele permite que o algoritmo mude a classe de objetos inicialmente rotulados no caso da informação de classe de objetos vizinhos divergirem em relação à classe original. Portanto, uma menor confiança do algoritmo GNetMine associada com a inferência de potenciais notícias reais e falsas do algoritmo PU-LP pode estar beneficiando a abordagem.

Tabela 19 – Análise de melhores parâmetros de PULP-FND com Yake! considerando F_1 macro e *fake*.

	Fact-checked News			**Fake.BR**			**FakeNewsNet**			**FakeNewsCorpus 0**			**FakeNewsCorpus 1**			**FakeNewsCorpus 2**		
	***fake* F_1**																	
Parameter	10%	20%	30%	10%	20%	30%	10%	20%	30%	10%	20%	30%	10%	20%	30%	10%	20%	30%
D2V Rep. model	rep. 3	rep. 4	rep. 4	rep. 4	rep. 4	rep. 4	rep. 3	rep. 3	rep. 3	rep. 3	rep. 3	rep. 3	rep. 4	rep. 3	rep. 3	rep. 4	rep. 3	rep. 3
keyword fix/var	var	fix	fix	fix	fix	fix	var	var	var	fix	fix	fix	fix	fix	fix	fix	fix	var
% keyword	10%	5%	5%	10%	10%	10%	10%	10%	10%	5%	5%	5%	5%	5%	5%	5%	5%	5%
n-gram	3	3	3	3	3	3	1	1	1	3	3	3	3	3	3	1	3	3
PU-LP *k*-NN network	5	7	6	7	7	7	6	5	7	7	5	5	5	5	5	5	6	5
PU-LP α	0.005	0.005	0.02	0.02	0.005	0.005	0.005	0.005	0.01	0.005	0.005	0.005	0.005	0.005	0.005	0.005	0.005	0.005
PU-LP λ	0.6	0.8	0.8	0.6	0.6	0.6	0.8	0.8	0.8	0.6	0.8	0.6	0.8	0.8	0.6	0.6	0.6	0.6
GNM α	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Parameter	**F_1 macro**																	
D2V Rep. model	rep. 3	rep. 4	rep. 4	rep. 4	rep. 4	rep. 4	rep. 3	rep. 3	rep. 3	rep. 3	rep. 3	rep. 3	rep. 4	rep. 3	rep. 3	rep. 3	rep. 3	rep. 3
keyword fix/var	var	fix	fix	fix	fix	fix	var	var	var	fix	fix	fix	fix	fix	fix	fix	fix	var
% keyword	10%	5%	5%	10%	10%	10%	10%	10%	10%	5%	5%	5%	5%	5%	5%	5%	5%	5%
n-gram	3	3	3	3	3	3	1	1	1	3	3	3	3	3	3	3	3	3
PU-LP *k*-NN network	5	7	6	7	7	7	5	5	7	7	5	5	5	5	5	7	6	5
PU-LP α	0.005	0.005	0.02	0.02	0.01	0.02	0.02	0.02	0.01	0.005	0.005	0.005	0.005	0.005	0.005	0.005	0.005	0.005
PU-LP λ	0.6	0.8	0.8	0.8	0.6	0.6	0.6	0.6	0.8	0.6	0.8	0.6	0.8	0.8	0.6	0.8	0.6	0.6
GNM α	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5

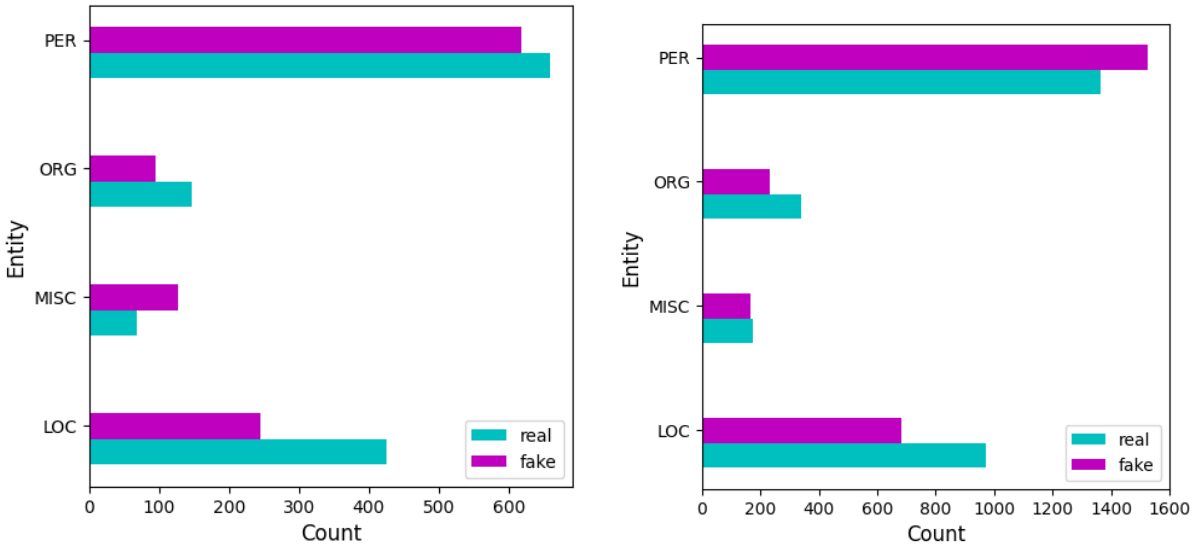
Considerando o conjunto de melhores parâmetros e a explicabilidade da abordagem, foram analisadas a classificação do algoritmo GNetMine atribuída aos termos extraídos com Yake!. GNetMine é um algoritmo semissupervisionado baseado em regularização de grafos, que atribui uma pontuação $f(c_{\text{real}}, c_{\text{fake}})$ para cada nó da rede, indicando o quanto aquele nó pertence à classe real e falsa.

Para cada nó termo, foram checados se a palavra-chave correspondia a alguma entidade e a qual classe ela estava mais relacionada. As entidades foram identificadas com o auxílio da biblioteca *Spacy* (HONNIBAL; MONTANI, 2017), que fornece implementações das línguas portuguesa e inglesa, a qual possui mais recursos. Tais contagens são apresentadas nas Figuras 33, 34 e 35.

Considerando a base de dados Fact-checked News (Figura 33a), que contém apenas notícias políticas, entidades do tipo LOC (localizações) e ORG (companhias, agências, instituições) aparecerem mais associadas às notícias reais, enquanto entidades do tipo MISC (diversas, isto é, eventos, nacionalidades, produtos, trabalhos de arte, etc.) apareceram mais relacionadas ao conteúdo falso. Não há diferença aparente considerando a entidade PER (pessoas, incluindo fictícias) e as classes real e falsa. Algo similar ocorreu com a base Fake.BR (Figura 33b), na qual a maioria das notícias são de política. No entanto, entidades do tipo MISC aparecem com a mesma frequência em ambas as classes.

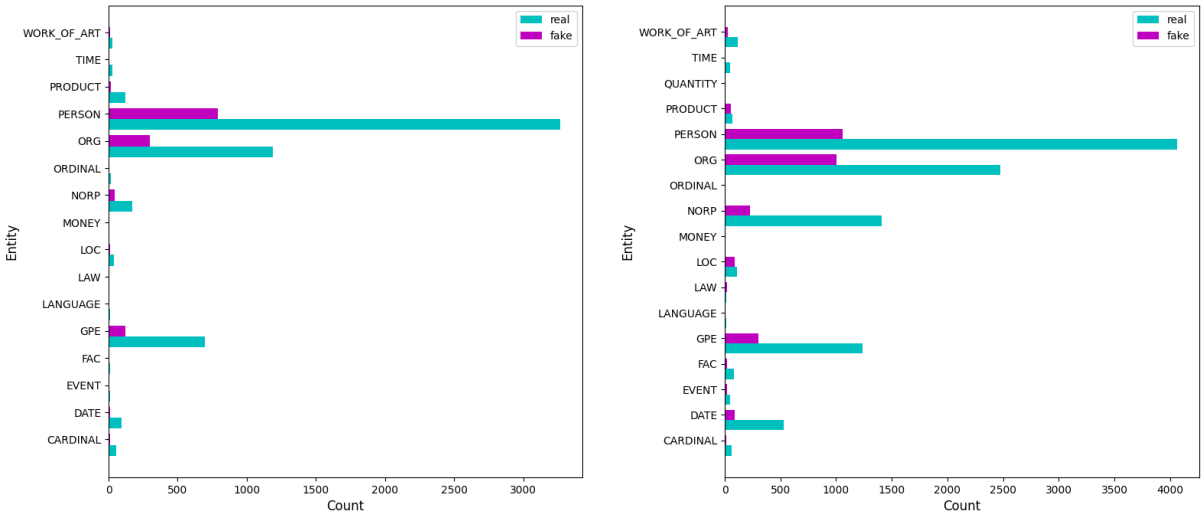
Como mencionado anteriormente, *Spacy* fornece um conjunto mais amplo de entidades

Figura 33 – Contagem de entidades nomeadas na rede *k*-NN classificadas pelo algoritmo GNetMine como pertencentes as classes falsas e reais para as bases de dados em inglês FakeNewsNet e FakeNewsCorpus 0, respectivamente.



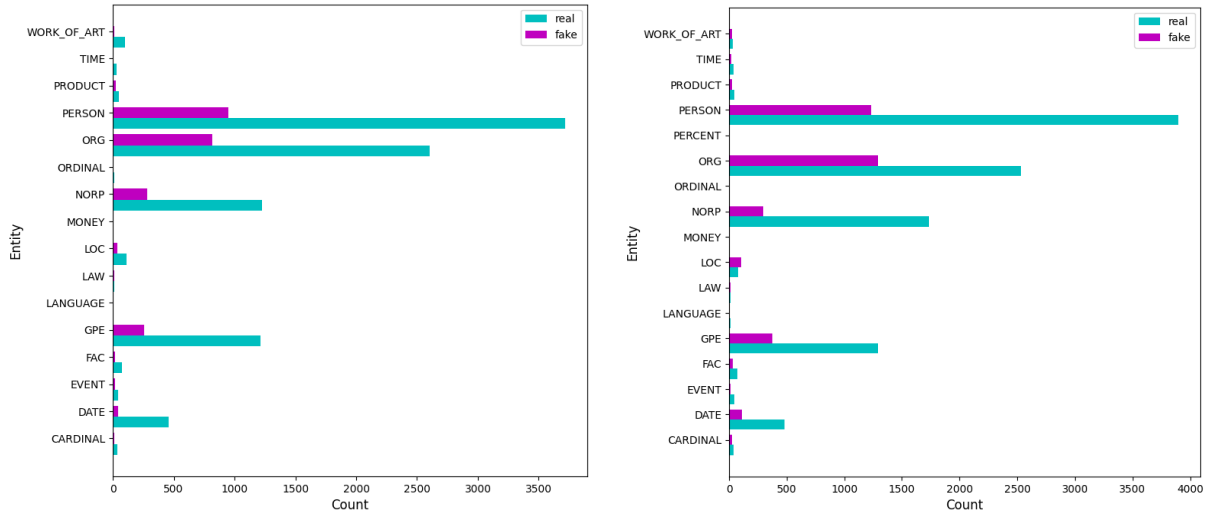
(a) Fact-checked News (b) Fake.BR

Figura 34 – Contagem de entidades nomeadas na rede *k*-NN classificadas pelo algoritmo GNetMine como pertencentes as classes falsas e reais para as bases de dados em inglês FakeNewsNet e FakeNewsCorpus 0, respectivamente.



(a) FakeNewsNet (b) FakeNewsCorpus 0

Figura 35 – Contagem de entidades nomeadas na rede k-NN classificadas pelo algoritmo GNetMine como pertencentes as classes falsas e reais para as bases de dados em inglês FakeNewsCorpus 1 e FakeNewsCorpus 2, respectivamente.



(a) FakeNewsCorpus 1

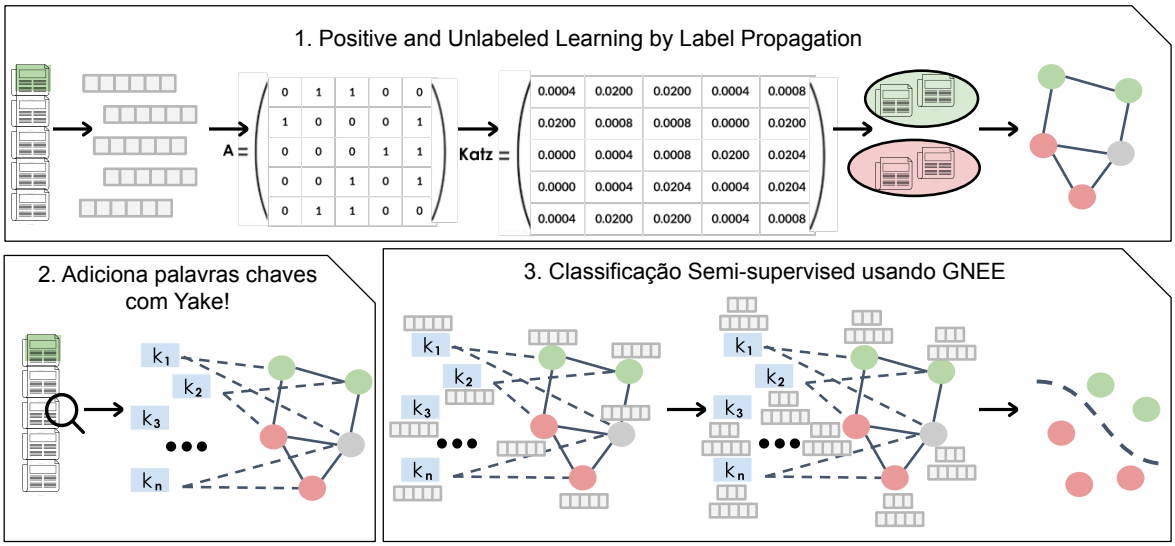
(b) FakeNewsCorpus 2

para bases de dados em inglês. Em FakeNewsNet (Figura 34a), que apresenta notícias de celebridades, é evidente que termos do tipo cardinal, date (data), GPE (países, cidades, estados), NORP (nacionalidades, religiões ou grupos políticos), ORG e PER estão mais associadas à classe real. Embora a base de dados seja desbalanceada, com mais de 70% de notícias reais, o padrão se repete nas demais bases em inglês. FakeNewsCorpus 0, FakeNewsCorpus 1 e FakeNewsCorpus 2 (Figuras 34b, 35a e 35b) apresentam diferentes assuntos, nas quais entidades do tipo cardinal, date, GPE, NORP, ORG e PER também aparecem mais relacionadas a classe real. Estas análises confirmam resultados encontrados na literatura, indicando que notícias reais tendem a apresentar dados concretos com maior riqueza de informações em relação a notícias falsas.

4.5 AK-PULP-FND: Atenção em Palavras-chave para Detecção de Fake News usando PU-LP

Nas abordagens anteriores foi demonstrado que o enriquecimento da rede de notícias com termos representativos extraídos de Yake! beneficiaram o processo de classificação, em conjunto com o algoritmo GNetMine, em especial quando notícias falsas estavam distribuídas no espaço de características. GNetMine considera o peso da aresta de nós vizinhos para calcular a informação de classe de um nó não rotulado. Entretanto, estes pesos precisam ser manualmente definidos e calibrados na construção da rede. Além disso, uma das premissas do GNetMine é que nós diretamente conectados no grafo são prováveis de pertencerem a mesma classe. Essa premissa pode restringir a capacidade do modelo se for considerado que as arestas não necessariamente codificam a similaridade de nós.

Figura 36 – AK-PULP-FND usando mecanismos de atenção que aprendem a importância de termos relevantes para classificação de notícias.



Para mitigar a limitação do GNetMine, foi proposta a substituição do algoritmo na abordagem PU-LP. A nova abordagem, denominada Atenção em Palavras-chave para Detecção de Fake News usando PU-LP (AK-PULP-FND), incorpora mecanismos de atenção em PU-LP por meio do algoritmo Graph Attention Neural Event Embedding (GNEE) (MATTOS; MARCACINI, 2021), que classifica nós não rotulados. GNEE é um algoritmo semissupervisionado baseado em GAT e regularização (ZHU; GHAHRAMANI; LAFFERTY, 2003) desenvolvido inicialmente para classificação de eventos e seus componentes, como localização e atores, em uma rede heterogênea. GNEE explora regularização em grafos para criar *embeddings* de nós componentes, propagando características textuais de eventos para as novas representações de componentes. A regularização é responsável por manter todos os nós no mesmo espaço de características. Posteriormente, o algoritmo explora a topologia do grafo, os vetores de características dos eventos e componentes, e a informação de nós rotulados para melhorar o processo de aprendizado de *embeddings* de baixa dimensão, usando GAT para aprender implicitamente quais representações de nós componentes são mais relevantes, atribuindo a elas diferentes importâncias no processo de classificação de um evento.

A abordagem AK-PULP-FND é apresentada na Figura 36. Após a extração dos conjuntos de notícias potencialmente reais e falsas efetuada pelo algoritmo PU-LP (etapa 1), e da adição de palavras-chave na rede usando Yake! (etapa 2), na etapa 3 aplica-se o algoritmo GNEE para a classificação de notícias não rotuladas, descrito a seguir.

Seja $\mathscr{N} = \langle \mathscr{O}, \mathscr{R}, \mathscr{W} \rangle$, na qual $\mathscr{O}$ é o conjunto de objetos, $\mathscr{R}$ é o conjunto de relações entre estes objetos, e $\mathscr{W}$ é o conjunto de pesos destas relações. O conjunto O é composto por dois tipos de objetos, $\mathscr{O}_{\mathscr{D}} \cup \mathscr{O}_T$, no qual $\mathscr{O}_D$ são notícias e $\mathscr{O}_T$ são termos. A informação textual da notícia $o_i \in \mathscr{O}_D$ é representada por um vetor de características $\vec{g}_{o_i} \in \mathbb{R}^m$ de espaço m-dimensional extraído com D2V, assim como nos experimentos anteriores. As notícias contêm

possíveis rótulos $\mathscr{C} = \{\text{falsa}, \text{real}\}$ e a rede contém alguns vértices rotulados $\mathscr{O}_L \in \mathscr{O}_D$ formando o conjunto de treinamento $\{(o_1, c_1), \ldots, (o_n, c_n)\}$ para aprendizado semi-supervisionado, com $n = |\mathscr{O}_L|$ e $c_j \in \mathscr{C}$. A rede neural de notícias é representada como uma função de mapeamento $h : \mathscr{N} = \langle \mathscr{O}, \mathscr{R}, \mathscr{W} \rangle \rightarrow \mathbb{R}^b$ dos vértices para um vetor de características b-dimensional, no qual b é um parâmetro pré-definido.

Para realizar a regularização, GNEE considera duas premissas, garantidas pela Equação 4.5: (i) vértices vizinhos possuirão vetores de características similares, ou seja, dois vértices vizinhos com w_{o_i,o_j} precisam ter uma baixa diferença de similaridade entre seus vetores de características (primeiro termo da equação); (ii) vetores de características de notícias permanecerão inalterados durante a regularização, assegurado pelo termo $\lim_{\mu \rightarrow \infty} \mu$ que garante que uma pequena diferença $(\mathbf{f}_{o_i} - \mathbf{g}_{o_i})^2$ penaliza muito a função objetivo $Q(\mathbf{F})$ (segundo termo da equação). Com isso, espera-se que um termo associado a múltiplas notícias tenha um vetor de características similar a elas.

$$Q(\mathbf{F}) = \frac{1}{2} \sum_{o_i, o_j \in \mathscr{O}} w_{o_i,o_j} (\mathbf{f}_{o_i} - \mathbf{f}_{o_j})^2 + \lim_{\mu \rightarrow \infty} \mu \sum_{o_i \in \mathscr{O}_L} (\mathbf{f}_{o_i} - \mathbf{g}_{o_j})^2. \tag{4.5}$$

A Equação 4.5 apresenta um caso particular de regularização que possui provas teóricas de convergência e pode ser resolvida por meio de métodos iterativos baseados em propagação de rótulos (ZHU; GHAHRAMANI; LAFFERTY, 2003). Após o passo de regularização, todos os vértices do grafo estarão representados no mesmo espaço de características $\mathbf{F}$.

O próximo passo envolve o aprendizado de *embeddings* no grafo semissupervisionado transdutivo por meio de mecanismos de atenção. A entrada consiste no conjunto de características regularizadas dos vértices $\mathbf{F} \in \mathbb{R}^{|\mathscr{O}| \times m}$, onde m é a dimensão das características textuais das notícias e $\mathscr{O}$ é o número total de vértices. GNEE permite o aprendizado de um novo conjunto de características de alto nível $\mathbf{Z} \in \mathbb{R}^{|\mathscr{O}| \times b}$, no qual b é a dimensão do novo espaço de características. Para isso, GNEE explora um mecanismo de *self-attention* compartilhado $\text{att} : \mathbb{R}^{\mathbf{Z}} \times \mathbb{R}^{\mathbf{Z}} \rightarrow \mathbb{R}$ proposto em VELICKOVIC *et al.* (2017), definido na Equação 4.6, na qual $A \in \mathbb{R}^{d \times m}$ é uma matriz de pesos e $\mathbf{z}_i$ e $\mathbf{z}_j$ são vetores de características dos vértices o_i e o_j, respectivamente.

$$a_{o_i,o_j} = \text{att}(\mathbf{A}\mathbf{z}_{o_i}, \mathbf{A}\mathbf{z}_{o_j}) \tag{4.6}$$

Um passo importante de redes de atenção em grafos é considerar relações entre notícias e termos nos mecanismos de atenção. Neste caso, a_{o_i,o_j} é apenas calculado para os nós vizinhos N_{o_i} do vértice o_i, seguido pela normalização pela função *Softmax*, conforme a Equação 4.7. Nesta equação, α_{o_i,o_j} indica a importância normalizada das características do vértice o_i para o vértice o_j considerando os k vértices da vizinhança N_{o_i}.

$$\alpha_{o_i,o_j} = \text{softmax}(a_{o_i,o_j}) = \frac{\exp(a_{o_i,o_j})}{\sum_{o_k \in N_{o_i}} \exp(a_{o_i,o_k})} \tag{4.7}$$

Os coeficientes de atenção $\alpha_{o_i,o_j} \forall o_j \in N_{o_i}$ são usados para aprender o vetor de características $\mathbf{z_{o_i}}$ por meio da combinação linear de todos os vetores de características dos vértices vizinhos, como definido na Equação 4.8. O termo σ representa alguma função não linear. Este processo é aplicado a todos os vértices para a obtenção do espaço de *embeddings* $\mathbf{Z}$ do grafo.

$$\mathbf{z}_{o_i} = \sigma\left(\sum_{o_j \in N_{o_i}} \alpha_{o_i,o_j} \mathbf{A}\mathbf{z}_{o_j}\right) \tag{4.8}$$

Em MATTOS; MARCACINI (2021) é utilizado um mecanismo de atenção para cada componente de evento incluído na rede heterogênea. Desta forma, espera-se que cada mecanismo de atenção possa aprender a influência de cada tipo de vértice componente na classificação de vértices de eventos. Considerando que além das notícias só foram incluídos termos na rede da abordagem AK-PULP-FND, este trabalho considera um único mecanismo de atenção durante os experimentos. As *embeddings* de baixa dimensão geradas por GNEE alimentam uma camada final com função logística de ativação Sigmoide, que realiza a classificação final das notícias.

Para avaliar o desempenho da abordagem AK-PULP-FND, foram considerados os algoritmos baseados em *One Class Graph Neural Networks* descritos na Seção 3.5: *One-class Graph Convolutional Network*, *One-class Graph Attention Network* e *One-class GraphSage*.

4.5.1 Configuração Experimental e Critérios de Avaliação

Devido ao alto custo de memória de GNEE e dos demais algoritmos de comparação baseados em OCGNNs, foram realizados experimentos considerando apenas as representações de Doc2Vec 3 e 4, baseadas em concatenação. Os parâmetros de PU-LP utilizados foram $k = [5,7]$, $\alpha = \{0.005, 0.01\}, \lambda = \{0.6, 0.8\}$ e $m = 2$. Os termos relevantes foram extraídos com Yake!. Os parâmetros utilizados foram: valor fixo de k correspondendo a 5% e 10% de $\bar{M}$. Os demais parâmetros foram os mesmos estabelecidos na Subseção 4.4.1.

Para **GNEE** foram considerados os parâmetros: $\alpha = 0.2$, *dropout* $= 0.5$, *epochs* $= 20$, *hidden layers* $= 8$, *threshold rate* $= 0.005$, *patience* $= 100$, *seed* $= 72$, *weight decay* $= 0.0005$, conforme as recomendações em MATTOS; MARCACINI (2021) e *attention heads* $= 1$. Cada *attention head* deriva *embeddings* de 8 dimensões. Para **OC-GCN**, **OC-GAT** e **OC-GrapgSAGE** (WANG *et al.*, 2021), foram exploradas combinações dos seguintes parâmetros: *patience* = $\{50, 100, 150\}$, *architecture* = $\{[128,128],[64,64],[32,32]\}$, $\nu = \{0.0001, 0.001, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$, *weight decay* = 0.0005, *learning rate* = $\{0.001, 0.005, 0.01\}$, *dropout* = 0.5, *seed*=81 e *epochs* = 500.

Para avaliar a efetividade de AK-PULP-FND na inferência dos conjuntos *RI* e *RN* (conjuntos de notícias de interesse e não interesse confiáveis), foi proposto um modelo de referência usando aprendizado semissupervisionado binário (BL). Neste algoritmo, o conjunto de notícias reais foi aleatoriamente dividido em 10 *folds*, e durante a validação cruzada foi usado

um *fold* de notícias reais para cada *fold* de notícias falsas escolhido. O algoritmo então calcula a matriz k-NN e segue os mesmos passos e parâmetros de PU-LP, usando Yake! para extração de palavras-chave e GNEE para classificação de nós não rotulados.

4.5.2 Resultados e Discussões

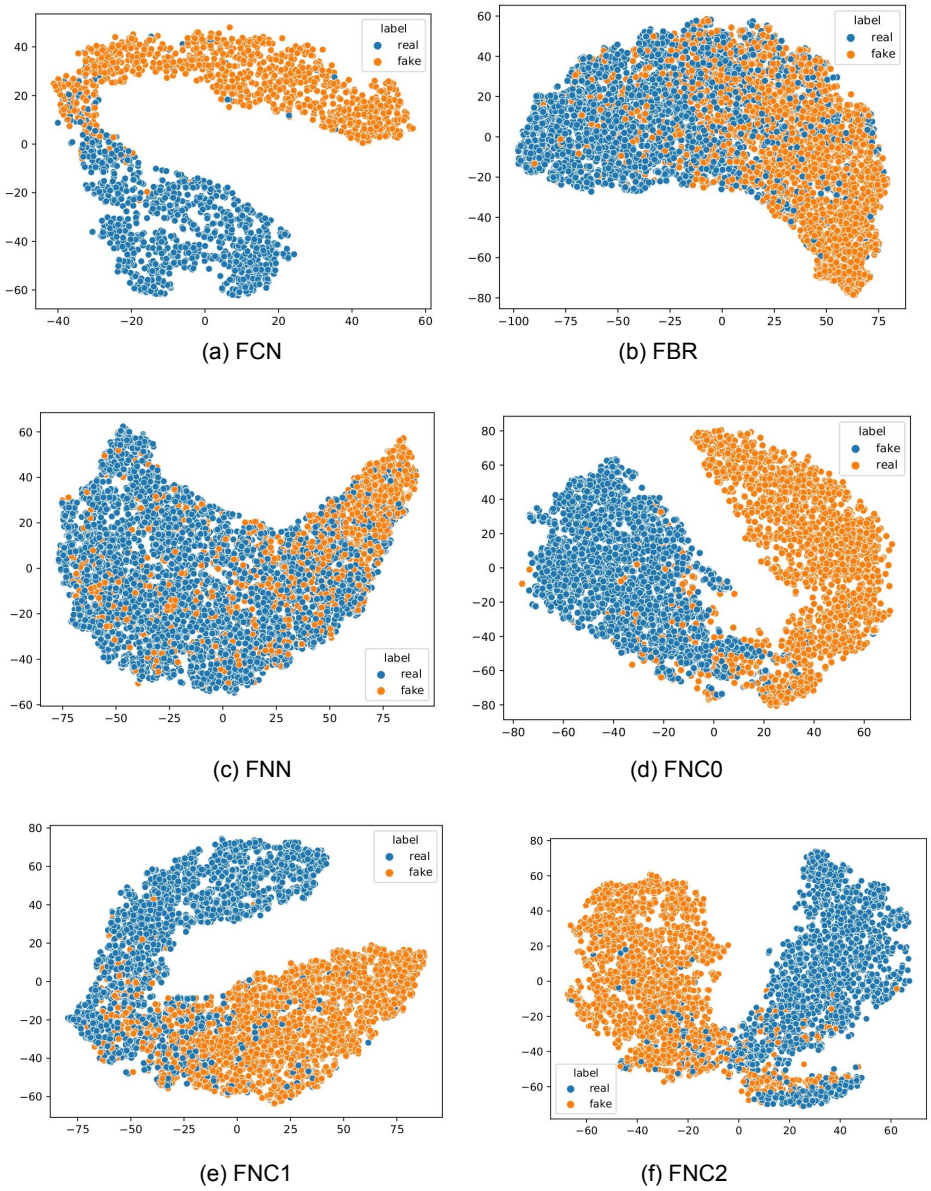
Na Tabela 20 são apresentados os resultados atingidos. A primeira linha corresponde ao PU-LP homogêneo (rede de notícias). A segunda linha corresponde a abordagem AK-PULP-FND, cuja rede é composta por notícias e termos extraídos com Yake!. Em ambos os algoritmos, os nós não rotulados são classificados com GNEE. As linhas 3 a 5 correspondem aos métodos estado da arte baseados em OCGNN. As duas últimas linhas correspondem ao modelo de referência binário semissupervisionado (BL GNEE), sendo a primeira composta de notícias e a segunda de notícias e termos extraídos com Yake!. As primeiras três colunas correspondem a *fake* F_1 e as três últimas colunas a F_1 macro. 10%, 20% e 30% indicam a porcentagem de notícias falsas rotuladas usadas no conjunto de treinamento dos algoritmos baseados em PU-LP e dos algoritmos baseados em uma única classe. Quanto ao algoritmo BL GNEE, são consideradas notícias reais e falsas no treinamento, considerando a mesma proporção.

Tabela 20 – Comparação das abordagens PU-LP, AK-PULP-FND, OCGNN e BL na detecção de notícias falsas. 10%, 20% e 30% indicam a porcentagem de notícias falsas usadas no conjunto rotulado. Os melhores resultados estão destacados em cinza.

	Fake F_1			**Macro F_1**		
	10%	**20%**	**30%**	**10%**	**20%**	**30%**
Fact-checked news						
PU-LP	0.9255	0.9205	0.9180	0.9263	0.9250	0.9270
AK-PULP-FND	**0.9489**	**0.9482**	**0.9468**	**0.9501**	**0.9518**	**0.9536**
OC-GraphSAGE	0.6858	0.7187	0.7577	0.6783	0.6781	0.6925
OC-GCN	0.7814	0.8189	0.8340	0.7597	0.7801	0.7879
OC-GAT	0.8687	0.8846	0.8925	0.8656	0.8744	0.8744
BL GNEE (Notícias)	0.9405	0.9433	0.9436	0.9400	0.9429	0.9432
BL GNEE (Notícias+Termos)	0.9607	0.9610	0.9627	0.9601	0.9606	0.9623
Fake.BR						
PU-LP	0.6816	0.6852	0.6734	0.6763	0.7062	0.7210
AK-PULP-FND	**0.7250**	**0.7376**	0.7269	**0.7310**	**0.7609**	**0.7691**
OC-GraphSAGE	0.6897	0.7143	**0.7408**	0.5510	0.5545	0.5610
OC-GCN	0.6897	0.7143	0.7407	0.5830	0.5821	0.5843
OC-GAT	0.6897	0.7146	0.7407	0.6258	0.6256	0.6206
BL GNEE (Notícias)	0.7147	0.7233	0.7273	0.7273	0.7349	0.7373
BL GNEE (Notícias+Termos)	0.7827	0.7892	0.7899	0.7928	0.7990	0.8001
FakeNewsNet						
PU-LP	0.5141	0.4858	0.4811	0.6533	0.6284	0.6429
AK-PULP-FND	0.5316	0.5001	0.4904	**0.6749**	**0.6492**	**0.6552**
OC-GraphSAGE	**0.8735**	**0.8860**	**0.8990**	0.5726	0.5710	0.5674
OC-GCN	**0.8735**	**0.8860**	0.8988	0.5601	0.5642	0.5667
OC-GAT	**0.8735**	**0.8860**	0.8988	0.5905	0.5843	0.5876
BL GNEE (Notícias)	0.2555	0.2784	0.2954	0.5658	0.5781	0.5872
BL GNEE (Notícias+Termos)	0.0080	0.0089	0.0077	0.4349	0.4354	0.4347
FakeNewsCorpus 0						
PU-LP	0.9138	0.9122	0.9072	0.9164	0.9178	0.9177
AK-PULP-FND	**0.9383**	**0.9303**	**0.9383**	**0.9395**	**0.9348**	**0.9461**
OC-GraphSAGE	0.8421	0.8448	0.8514	0.8449	0.8417	0.8358
OC-GCN	0.8484	0.8525	0.8538	0.8558	0.8503	0.8430
OC-GAT	0.8771	0.8868	0.8966	0.8656	0.8744	0.8744
BL GNEE (Notícias)	0.9292	0.9317	0.9324	0.9267	0.9295	0.9302
BL GNEE (Notícias+Termos)	0.9346	0.9361	0.9366	0.9319	0.9337	0.9343
FakeNewsCorpus 1						
PU-LP	0.8962	0.8974	0.8934	0.8985	0.9042	0.9067
AK-PULP-FND	**0.9098**	**0.9104**	**0.9078**	**0.9129**	**0.9171**	**0.9194**
OC-GraphSAGE	0.8281	0.8395	0.8469	0.8307	0.8327	0.8303
OC-GCN	0.8531	0.8565	0.8643	0.8549	0.8497	0.8482
OC-GAT	0.8790	0.8823	0.8917	0.8762	0.8707	0.8719
BL GNEE (Notícias)	0.9121	0.9141	0.9155	0.9122	0.9148	0.9163
BL GNEE (Notícias+Termos)	0.9178	0.9182	0.9188	0.9185	0.9194	0.9202
FakeNewsCorpus 2						
PU-LP	0.8925	**0.9022**	0.9068	**0.8983**	0.9114	0.9201
AK-PULP-FND	**0.8931**	0.9018	**0.9110**	0.8976	**0.9119**	**0.9238**
OC-GraphSAGE	0.8205	0.8239	0.8326	0.8291	0.8216	0.8149
OC-GCN	0.8334	0.8378	0.8389	0.8407	0.8359	0.8308
OC-GAT	0.8696	0.8762	0.8883	0.8681	0.8638	0.8678
BL GNEE (Notícias)	0.9198	0.9205	0.9196	0.9201	0.9213	0.9210
BL GNEE (Notícias+Termos)	0.9224	0.9263	0.9276	0.9224	0.9267	0.9284

Na Tabela 20, a abordagem AK-PULP-FND supera os demais algoritmos para a maior parte das bases de dados. A classificação de nós não rotulados com GNEE após a inferência

Figura 37 – Representações de notícias reais e falsas plotadas em duas dimensões com a ferramenta t-SNE das *embeddings* aprendidas com GNEE, considerando a topologia da rede (relações entre notícias e termos) e mecanismos de atenção.



dos conjuntos *RP* e *RN* com PU-LP e a inclusão de termos relevantes com Yake! aumentou o desempenho de classificação em relação ao algoritmo GNetMine (Tabela 18), especialmente usando 10% de notícias falsas inicialmente rotuladas. Exceções ocorreram apenas com a base de dados FakeNewsNet, que assim como nos experimentos anteriores, apresentou melhor *fake* F_1 com abordagens baseadas em OCL.

Na Tabela 20, as abordagens OC-GCN, OC-GAT e OC-GraphSAGE apresentaram resultados similares. Isto ocorreu já que os algoritmos possuem a OCGNN como método base, cujo objetivo é encapsular os exemplos da classe de notícias falsas. Neste processo, os pesos das camadas GNN são atualizados considerando a mesma função de perda. Portanto, independente

da camada GNN usada (GCN, GAT ou GraphSAGE), os resultados podem ser muito próximos, já que a função de perda obteve melhor desempenho com representações geradas de uma única forma, obtidas por meio da GNN.

Conforme esperado, o modelo de referência binário semissupervisionado BL superou os demais algoritmos, avaliado com notícias reais e falsas inicialmente rotuladas. Novamente, a única exceção ocorreu com a base FakeNewsNet, que possui extremo desbalanceamento. Considerando que as notícias falsas estão distribuídas no espaço de características, e que as notícias estão relacionadas considerando medidas de similaridade, o desbalanceamento contribuiu para que o algoritmo binário classificasse a maioria dos exemplos não rotulados como notícias reais, resultando em um desempenho inferior em relação ao PU-LP e demais algoritmos de comparação.

Considerando os melhores resultados de cada base de dados, atingidos por meio de redes compostas de notícias e termos, na Figura 37 é apresentada a plotagem em duas dimensões das embeddings aprendidas com GNEE. Exceto para a base FakeNewsNet, é possível observar que o mecanismo de atenção, unido com a inclusão de termos, foi capaz de realizar boa separação de notícias reais e falsas no espaço de características, o que justifica a melhoria de desempenho.

4.6 Considerações Finais

Neste trabalho foram propostas abordagens baseadas no algoritmo PU-LP que realizam a detecção de notícias falsas por meio de redes heterogêneas. Nas abordagens, a similaridade de cosseno entre duas representações de notícias é responsável por uni-las na rede e fornecer o peso da relação entre elas. Neste contexto, foram realizadas análises semi-automáticas nas bases de notícias, com a intenção de avaliar em quais cenários esta estratégia pode ser benéfica e em quais ela apresenta limitações.

Para realizar estes experimentos, foram escolhidas aleatoriamente notícias falsas em cada base de dados, e considerando a matriz de adjacência obtida pelo melhor modelo de representação Doc2Vec (que gerou os melhores resultados para a base de dados, conforme a Tabela 19), foram recuperadas as 5 notícias mais similares e mais dissimilares em relação àquela previamente selecionada. A partir deste conjunto de notícias, foram realizadas análises considerando o conteúdo original da notícia (sem pré-processamento), bem como palavras-chave extraídas com Yake!.

Para a base Fake.BR foi selecionada a seguinte notícia falsa sobre política:

```
Fake.BR/fake/165: Veja quem são os políticos que embolsaram propina da Odebrecht (e
    quanto eles ganharam). A delação da Odebrecht, também conhecida como Delação do Fim
    do Mundo, pode colocar na cadeia os políticos mais influentes do governo Temer. Veja
    quem são e quanto cada um embolsou em propina. A Odebrecht gastou pelo menos 88
    milhões em propina, caixa dois e doações legais para campanhas de 48 políticos entre
    2006 e 2014 [...].

Palavras-chave (Yake!): Cláudio Melo Filho, construtora Cláudio Melo, presidente Michel
    Temer, pra Geraldo Alckmin, milhões pra Geraldo, legenda receberam cerca, Melo Filho,
    Cláudio Melo, Michel Temer, Geraldo Alckmin, José Serra, delação premiada, MIRA DA
    ODEBRECHT, governo Temer, Odebrecht gastou, ajudas no Congresso,construtora Cláudio,
    presidente Michel.
```

Ao selecionar as notícias mais próximas de **Fake.BR/fake/165**, foram obtidas apenas uma notícia falsa e quatro notícias reais, cujas distâncias variaram de $0,61$ a $0,64$. Como notícias mais dissimilares, foram recuperadas cinco notícias reais, com distância variando de $1,19$ a 1.21. A seguir, é apresentado o conteúdo de uma das notícias mais próximas e a mais distante, ambas reais, para discussão.

```
Fake.BR/real/2684 - Distância=0.62: Delações da Odebrecht apontam propina para pais e
    filhos, maridos e mulheres, em pagamentos de caixa 2. Em alguns casos, familiares
    chegavam a receber dinheiro dos pagamentos não oficiais; boa parte dos casos é entre
    parentes já envolvidos na política. As delações divulgadas nesta quarta-feira (13)
    com depoimentos dos donos e ex-executivos da Odebrecht mostram o recebimento de caixa
    2 por políticos e, muitas vezes, seus parentes [...].
```

```
Fake.BR/real/89 - Distância=1.21: Ativista responde piada considerada gordofóbica de
    Danilo Gentili 'Nunca foi motivo de riso ser visto como doente', diz Alexandra Gurgel
    do canal Alexandrismos Alexandra Gurgel é a mulher da foto que abre a reportagem ``A
    gente não quer mais ser visto como doente': a vida de quem é alvo de gordofobia', da
    BBC Brasil, e também foi alvo de piadas consideradas gordofóbicas de Danilo Gentili
    [...].
```

A notícia real mais próxima em conteúdo a notícia falsa apresenta palavras-chave relacionadas, como “delação”, ‘Odebrecht”, “governo” e “executivo”. Por outro lado, a notícia mais distante, também verdadeira, contém termos relacionados a piadas de gordofobia.

Sobre o conjunto de notícias mais próximas a **Fake.BR/fake/165**, pode-se observar que a minoria é falsa, enquanto as cinco mais distantes são verdadeiras. O primeiro caso pode ser justificado considerando como a base de dados Fake.BR foi criada (SILVA *et al.*, 2020). Os autores coletaram e checaram manualmente 3.600 notícias falsas. Em seguida, os autores construíram um *crawler* capaz de recuperar notícias reais em sites confiáveis brasileiros, considerando um conjunto de palavras-chave extraídas das notícias falsas. Assim, a principal ideia era que notícias falsas tivessem suas correspondentes notícias reais presentes na base de dados, o que pode prejudicar o comportamento de PU-LP, especialmente na etapa de inferência de conjuntos de interesse e não interesse confiáveis.

Para a base em português Fact-checked News, foi selecionada a seguinte notícia falsa:

```
Factcheckednews/fake/4:  #Verificamos: É antiga foto que mostra mísseis na fronteira com
    o Brasil [...]. Circula nas redes sociais a ``informação'' de que a Venezuela teria
    posicionado mísseis na fronteira com o Brasil desde o dia 22 de fevereiro de 2019.
    Por meio do projeto de verificação de notícias, usuários do Facebook solicitaram que
    esse material fosse analisado. Confira a seguir o trabalho de verificação da Lupa:
    ``Exclusivo-Venezuela Posiciona Mísseis S-300 na Fronteira com o Brasil'' [...] Nota:
    esta reportagem faz parte do projeto de verificação de notícias no Facebook. Dúvidas
    sobre o projeto? Entre em contato direto com o Facebook. Editado por: Maurício Moraes
    e Cristina Tardáguila.

Palavras-chave (Yake!): Venezuela Posiciona Mísseis, Brasil Reportagem publicada,
    site América Militar, Venezuela teria posicionado, Lupa questionou órgãos, teria
    posicionado mísseis, material fosse analisado, suposta ação venezuelana, existia
    informação oficial, questionou órgãos brasileiros, mísseis na fronteira, Venezuela
    Posiciona, Maurício Moraes, Cristina Tardáguila, Defesa Aérea, Brasil Reportagem,
    Secretaria Especial, Especial de Comunicação, Comunicação Social, Social da
    Presidência, Presidência da República, Moraes e Cristina, Posiciona Mísseis, América
    Militar, Venezuela teria.
```

Na seleção de notícias mais próximas em conteúdo, foram recuperadas 5 notícias falsas, com distâncias variando de $0,63$ a $0,69$. Quanto as mais distantes, foram recuperadas três notícias reais e duas notícias falsas, com distâncias variando de $1,08$ a $1,10$. A seguir é apresentado o conteúdo da notícia falsa mais próxima e da notícia real mais distante:

```
Factcheckednews/fake/8 - Distância=0.63: Verificamos: É de 2016 vídeo que mostra tanques
    de guerra na fronteira com a Venezuela. Gravação viralizou na internet com se tivesse
    sido filmada no Brasil [...] Circula nas redes sociais um vídeo que mostra veículos
    do Exército Brasileiro se movimentando à noite. A legenda ``informa'' que a cena
    é recente e que o Brasil [...] Confira a seguir o trabalho de verificação da Lupa:
    [...] Nota: esta reportagem faz parte do projeto de verificação de notícias no
    Facebook. Dúvidas sobre o projeto? Entre em contato direto com o Facebook. Editado
    por: Cristina Tardáguila e Clara Becker.
```

```
Factcheckednews/real/507 Distância=1.10180: ``Se for expulsar todo mundo, não vai ficar
    muita gente, diz Alckmin ''O presidente do PSDB, Geraldo Alckmin, afirmou hoje ao
    blog que não houve acordão no partido para salvar o deputado Aécio Neves (MG) de um
    pedido de expulsão. A Executiva Nacional do PSDB se encontrou ontem e resolveu que,
    antes de decidir sobre o caso de Aécio, réu por corrupção no STF, precisa mudar o
    Estatuto da legenda e criar um Código de Ética [...].
```

Fact-checked News foi construída a partir da coleta de notícias em sites jornalísticos e sites de checagem de conteúdos brasileiros. Por um lado, a abordagem baseada em similaridade funciona bem nesta base de dados. Por outro lado, notícias reais e falsas estão bem discriminadas no espaço de características devido à diferença de linguagem entre ambas. Notícias reais apresentam linguagem formal, a fim de descrever os fatos. Para notícias falsas, embora palavras como ‘verificamos”, “falso” e “fake” tenham sido excluídas na fase de pré-processamento, há expressões que ainda poderiam enviesar o processo de representação de notícias como dados estruturados, como “projeto de verificação”, “LUPA”, “Editado por” e “Facebook”, que deveriam ter sido eliminadas na fase de pré-processamento.

Para a base em inglês FakeNewsNet, foi selecionada a seguinte notícia falsa:

```
FakeNewsNet/fake/3537769872: UPDATE: This article previously referred to Kanye West's
    album as 'Love Everyone,' and has been edited to reflect its title 'ye.' That's not
    being facetious. Considering the controversy surrounding Kanye West, his loosening
    grip on occupying the pinnacle of hip hop and ye being his eighth studio album, this
    album has the distinction of being a make-or-break point of the Chicago rapper and
    producer's career [...].

Palavras-chave (Yake!): West, Kanye, album, music, career, albums, Life, spotlight, Love,
    wasn, past, classic, questionable, important, artist, fans, Graduation, Yeezus, Video,
    Trump, questions, question, rapper, rappers, line, antics, pinnacle, hip, couple,
    controversy, surrounding, hop, point, individual, spent, felt, lot, top, shift, rap,
    ambitious, arguably, status, JAY-Z, acclaim, delivered, lyrics, time, greatest, great,
    explain, pressure, release, significant, rhymes, moment, UPDATE.
```

Entre as notícias mais próximas em conteúdo de **FakeNewsNet/fake/3537769872**, tem-se uma notícia falsa e quatro notícias reais, com distância variado de $0,72$ a $0,73$. No grupo de notícias mais distantes encontram-se uma notícia falsa e 4 notícias reais, cuja distância varia de $1,05$ a $1,09$. A seguir é apresentado o conteúdo da notícia mais próxima e da mais distante:

```
FakeNewsNet/real/935540 - Distância=0.72: It's no secret that Kanye West is causing a
    swarm of media speculation about his strange behavior. From calling slavery ``a
    choice'' to openly supporting President Trump, West has sparked just as much outrage
    as there is intrigue surrounding his celebrity. Basically, it's been two weeks of
    wild behavior and lots of questions. And now there's even more of that to contemplate.
    So, what's happened now? [...].
```

```
FakeNewsNet/real/4643978776 - Distância=1.07: George Clooney once again this week found
    himself having to issue a statement denying he helped enable perpetrators of sexual
    assault in Hollywood after an actress said he helped ``blacklist'' her after she
    complained about rampant sexual harassment on the set of ``E.R.''. In a series of
    tweets over the week, Vanessa Marquez called ``B.S.'' on Clooney's statements of
    concern about the female victims of workplace [...].
```

Como as notícias são relacionadas a celebridades, ambas reais e falsas apresentam linguagem informal e frequentemente sensacionalista, como "*facetious*", "*loosening*" (conteúdo falso), "*it's no secret that*", "*outrage*", "*intrigue*" (conteúdo real), além de abordar temas polêmicos como assédio sexual entre artistas. Este tipo de linguagem justifica o porquê notícias reais e falsas estão muito próximas no espaço de características, desfavorecendo abordagens baseadas em similaridade de conteúdo para detectar notícias falsas.

Para as bases de dados derivadas de FakeNewsCorpus, foi selecionada a seguinte notícia falsa:

```
FakeNewsCorpus/fake/2: Veteran Commentator Calls Out the Growing Ethnonationalism at Fox
    News (and Its SO Ugly) x\% of readers think this story is Fact Add your two cents
    Headline: Bitcoin and Blockchain Searches Exceed Trump! Blockchain Stocks Are Next!
    Lets be honest: This is pretty much all of Fox News commentary portion now, since the
    network became Trump Pravda Commentator and The Weekly Standard founder, Bill Kristol,
    sat for an interview with CNBC that was released on Thursday Among various topics, he
    noted how Tucker Carlson, a Fox News superstar, has changed since he began with The
    Weekly Standard Speaking [...].

Palavras-chave (Yake!): Black Lives Matter, nominee Hillary Clinton, State Hillary
    Clinton, nominee Donald Trump, Lives Matter movement, Party nominee Hillary, United
    States Supreme, States Supreme Court, Lives Matter protesters, order Historian
    Michael, Donald Trump speaks, campaign site Trump, presidential nominee Hillary,
    Party nominee Donald, Democratic Party nominee, Republican Party nominee, September
    Trump received, President Chuck Canterbury, activists interrupted Clinton, FOP
    National President, National President Chuck, South Carolina BLM, presidential
    nominee Donald, cities Democratic Party, African American men, bring back law,
    Carolina BLM activists, Matter protesters argues, Hillary Clinton, restricted police
    departments, police department body, criminal justice reform, Black Lives.
```

Entre as notícias mais próximas de **FakeNewsCorpus/fake/2** estão quatro notícias falsas e uma notícia real, com distância variando de $0,67$ a $0,70$. Entre as notícias mais distantes, todas são verdadeiras, cujas distâncias variam de $1,02$ a $1,03$. A seguir é apresentado o conteúdo da notícia mais próxima e mais distante:

```
FakeNewsCorpus/fake/4685 - Distância=0.67: Trump made Corey Booker cry x\% of readers
    think this story is Fact Add your two cents Headline: Bitcoin and Blockchain Searches
    Exceed Trump! Blockchain Stocks Are Next! In a lengthy and emotional speech Tuesday,
    Sen. Corey Booker said news that President Donald Trump allegedly referred to certain
    countries as sh thole or sh thouse places caused him to shed tears of rage Booker
    made the remarks during a 10 minute lecture to Homeland Security Secretary Kirstjen
    Nieslen, who he accused of having amnesia for not confirming whether Trump made the
    remarks at a meeting last week [...].
```

```
FakeNewsCorpus/real/46305 - Distância=1.03851: A member of the Black Lives Matter
    protesters argues with a police officer as they shut down the main road to the
    Minneapolis St. Paul Airport following a protest at the Mall of America in
    Bloomington, Minnesota, December 23, 2015 Demonstrations by Black Lives Matter
    to protest police killings of black men took place in Minnesota and California on
    Wednesday, a day the activist group dubbed ``Black Xmas'' to show it could impact the
    economy on one of the busiest shopping days of the year [...].
```

Nesta base de dados, expressões como "*x% of readers think this story is fact/fake*" podem estar enviesando o processo de aprendizado do modelo de representação. Além disso, assim como a base Fact-checked News, há diferença clara na linguagem usada em sites jornalísticos e em notícias falsas, que apresentam expressões informais e relacionadas a checagem de conteúdo.

Entre as bases de dados utilizadas nos experimentos, a que possui menor viés é a Fake.BR. Apesar de ser construída com notícias falsas e suas correspondentes reais, observou-se que a inclusão de termos relevantes extraídos com Yake! e incorporação deles na rede de notícias,

além do uso de GNEE, baseado em atenção, permitiu um aumento considerável da *fake* F_1 considerando poucas notícias falsas rotuladas. Para as bases de dados com expressões que podem enviesar o processo de classificação, Yake! foi hábil para identificar tais termos como relevantes, contribuindo para o aumento de desempenho de classificação.

Outro ponto relevante a ser discutido é que foram utilizadas bases de dados comumente aplicadas na literatura, como FakeNewsNet (com notícias de GossipCop) e FakeNewsCorpus, e foram apresentadas análises de como as notícias estão distribuídas no espaço de características, justificando o porquê. Tais análises são fundamentais para entender a qualidade das bases de dados utilizadas e questionar o alto desempenho de algoritmos de classificação propostos na literatura, que com frequência não são interpretáveis, e aplicados a um problema complexo como a detecção de notícias falsas.

Com a dificuldade evidente de se encontrar bases de dados de alta qualidade, além do fato de que algoritmos treinados com notícias de um determinado assunto geralmente não desempenham bem na discriminação de notícias de assuntos distintos, conclui-se a importância da continuidade de pesquisas que conduzam experimentos de detecção de notícias falsas diante de poucos dados rotulados.

CAPÍTULO

5

CONCLUSÕES

Neste capítulo são listadas as conclusões, sumarizadas considerando as questões de pesquisa apresentadas na introdução desta tese. Também são apresentadas as publicações realizadas ao longo do período de Doutorado, bem como trabalhos futuros que podem dar continuidade a esta pesquisa.

5.1 Contribuições Científicas

Neste trabalho foi proposta a abordagem PULP-FND para detecção de notícias falsas fundamentada no algoritmo *Positive and Unlabeled Learning by Label Propagation*, um algoritmo PUL, totalmente baseado em redes, capaz de aprender um modelo de classificação com poucos dados rotulados da classe de interesse - fake news. PU-LP infere conjuntos de notícias potencialmente reais e potencialmente falsas considerando o conjunto inicialmente rotulado e uma medida de similaridade baseada em caminhos. Por ser totalmente baseado em redes, foram analisadas características relevantes que pudessem ser incluídas, cujas relações pudessem adicionar novos padrões a estrutura e auxiliar na melhoria de classificação de algoritmos de propagação de rótulos, como *Label Propagation through Heterogeneus Networks* e GNetMine. As contribuições da abordagem PULP-FND podem ser sumarizadas com base nas questões de pesquisa **Q1** a **Q5** e objetivos apresentados na introdução desta tese.

Q1 *"Qual grupo de algoritmos (OCL x PUL) se destaca na detecção de notícias falsas?"*

Foram mapeados algoritmos de aprendizado de uma única classe e aprendizado positivo e não rotulado clássicos da literatura aplicáveis a dados textuais, analisando-se o desempenho destes algoritmos diante de bases de dados com diferentes cenários. Os algoritmos escolhidos foram *One-class k-Means, One-Class Suppport Vector Machine, k-Nearest Neighbor Density* e *Dense Auto Encoder*, além dos algoritmos PUL *Rocchio Support Vector Machine* e *Positive and Unlabeled Learning by Label Propagation*. Um modelo semissupervisionado binário foi

construído com base nos fundamentos do algoritmo PU-LP, para avaliar seu desempenho ao inferir conjuntos de notícias potencialmente reais e falsas a partir do conjunto inicial de notícias falsas rotuladas, servindo como modelo de referência.

Para avaliação dos algoritmos OCL e PUL, foram mapeadas bases de dados da literatura que apresentassem diferentes cenários quanto a linguagem, balanceamento das classes real e falsa, tipo de coleta dos dados e tópicos abordados. Foram escolhidas seis bases de dados: (i) Fake.BR, o primeiro corpus de referência para detecção de notícias falsas da língua portuguesa, balanceada, que contém notícias de seis assuntos distintos, coletadas manualmente; (ii) Fact-checked News, coletada automaticamente em sites de checagem de notícias brasileiros, balanceada e com conteúdo político: (iii) FakeNewsNet, contendo notícias em inglês sobre celebridades, na qual mais de 70% delas são reais, com linguagem sensacionalista; e (iv) FakeNewsCorpus, um repositório contendo milhares de notícias de 1001 domínios, coletadas automaticamente. FakeNewsCorpus foi utilizada para criação de 3 bases de dados balanceadas, denominadas pela autora como FakeNewsCorpus 0, FakeNewsCorpus 1 e FakeNewsCorpus 2.

Os resultados obtidos demonstraram que o desempenho de abordagens OCL e PUL está diretamente relacionado a como as representações de notícias estão distribuídas no espaço de características. Enquanto algoritmos OCL baseados em agrupamento e densidade se comportaram melhor em bases de dados desbalanceadas, cujas notícias falsas estavam distribuídas em pequenos grupos no espaço de características, abordagens PUL se destacaram quando notícias falsas estavam próximas entre si, minimamente separadas de notícias reais.

Q2 *"Como o modelo de representação utilizado para transformar notícias em dados estruturados pode influenciar no desempenho da abordagem de detecção de notícias falsas?"*

Foram mapeados modelos de representação capazes de transformar notícias completas em formato estruturado no espaço n-dimensional, com o intuito de utilizar os atributos aprendidos para calcular a similaridade de conteúdo elas e uni-las na estrutura da rede. Para realizar a transformação de notícias em formato estruturado, foram utilizados os modelos *Bag-of-Words* e *Doc2Vec*. Após plotadas as representações aprendidas de forma simplificada em duas dimensões pela ferramente t-SNE, notou-se que aquelas geradas pelo modelo Doc2Vec realizavam uma melhor separação de notícias reais e falsas no espaço de características. Os modelos gerados por D2V possibilitaram a melhoria de desempenho de classificação tanto para algoritmos PUL quanto para algoritmos OCL. Em algoritmos OCL, aqueles baseados em agrupamento e densidade foram os que apresentaram maiores ganhos de F_1.

Q3 *"Dentre as características textuais analisadas para inclusão na rede heterogênea, quais contribuem no desempenho da abordagem proposta de detecção de notícias falsas?"*

Construídas as redes de notícias para compor a abordagem PULP-FND, foram mapeadas novas características que pudessem ser extraídas do conteúdo da publicação e auxiliassem na discriminação de conteúdo real e falso se incorporadas na estrutura. Foram analisadas a inclusão

das seguintes características na rede de notícias: (i) termos representativos, extraídos a partir de uma *Bag-of-Words* com esquema de pesos tf-idf, (ii) emotividade, calculado em função do número de adjetivos, advérbios, sujeitos e verbos do texto; (iii) pausalidade, calculado em função do número de sinais de pontuação e sentenças do texto; e (iv) número médio de palavras por sentença. As características foram combinadas de diversas formas, analisando-se o desempenho de F_1 macro e da classe falsa. A rede que obteve melhor desempenho de classificação foi composta de notícias e termos. A inclusão de termos na rede de notícias aumentou o desempenho da abordagem, principalmente no uso do modelo de representação Bag-of-Words, e em bases de dados cujas notícias falsas estavam dispersas do espaço de características.

Q4 *"O desempenho da abordagem proposta, baseada em aprendizado de uma única classe semissupervisionado, supera algoritmos OCL e PUL da literatura? E algoritmos semissupervisionados binários?"*

Uma análise de ranqueamento médio comparando os desempenhos gerais dos algoritmos OCL e PUL demonstrou que a abordagem proposta PULP-FND, que adiciona termos representativos na rede de notícias e realiza a fase de propagação de rótulos usando o algoritmo GNetMine, obteve melhor desempenho geral. O ranqueamento médio de PULP-FND ficou bem próximos do modelo de referência binário semissupervisionado, mesmo utilizando apenas dados de notícias falsas no conjunto de treinamento.

Q5 *"O aumento do número de notícias falsas inicialmente rotuladas aumenta significativamente o desempenho de classificação dos algoritmos PUL?"*

Os experimentos demonstraram que os resultados de classificação considerando 10% e 30% de notícias falsas inicialmente rotuladas não apresentaram grande variação. Alguns algoritmos, como OCSVM, atingiram máxima F_1 macro e *fake* para a Fake.BR usando apenas 10% de dados rotulados. A abordagem PULP-FND atingiu mais de 88% de F_1 macro e *fake* em quatro bases de dados usando apenas 10% de dados rotulados. Tais resultados encorajam pesquisas por abordagens para detecção de notícias falsas a partir de poucos dados rotulados.

Embora a abordagem PULP-FND tenha superado de forma geral os demais algoritmos clássicos da literatura, ela apresentava algumas limitações. A primeira limitação encontrada correspondia a forma com que os termos eram selecionados para compor a rede heterogênea, diante da existência de ferramentas de extração de palavras-chave da literatura com maior eficiência para realizar a mesma tarefa. A segunda limitação encontrada foi sobre o algoritmo de propagação de rótulos GNetMine utilizado no contexto de classificação de notícias. GNetMine considera o peso das relações (arestas) de nós vizinhos para calcular a informação de classe de nós não rotulados, assumindo que nós diretamente conectados são prováveis de pertencerem à mesma classe. Esta característica pode restringir a capacidade do modelo já que no problema de detecção de notícias falsas nem sempre notícias relacionadas possuem o mesmo rótulo.

Para sanar estas limitações, foi proposta uma nova abordagem denominada AK-PULP-

FND. A nova abordagem consistiu na seleção de uma ferramenta de extração de palavras-chave sólida na literatura para extração mais assertiva de termos relevantes que pudessem compor a rede heterogênea, além da utilização de um algoritmo semissupervisionado estado da arte na etapa final de PU-LP, baseado em regularização e mecanismos de atenção. Mecanismos de atenção permitem o aprendizado implícito sobre quais características de nós vizinhos são mais relevantes para a classificação de uma notícia. Como a maioria dos nós vizinhos são termos, espera-se que o modelo aprenda quais deles são mais relevantes na classificação. As contribuições da abordagem AK-PULP-FND podem ser sumarizadas com base nas questões de pesquisa **Q6** a **Q8** e objetivos apresentados na introdução desta tese.

Q6 *"A inclusão de termos representativos na rede heterogênea, extraídos com ferramentas de extração de palavras-chave, pode aumentar o desempenho de classificação? Quais tipos de termos são relevantes na discriminação de conteúdo real e falso?"*

Foram mapeadas ferramentas de extração de palavras-chave disponíveis na literatura, bem como artigos que realizavam a comparação de desempenho destas ferramentas. A ferramenta escolhida foi a Yake!, por ser não supervisionada, independente de linguagem e ser capaz de detectar termos sem condicionar a sua relevância com a frequência que eles aparecem no documento. A inclusão de termos com a ferramenta Yake! ocasionou um aumento de F_1 macro e *fake*, em especial quando notícias falsas estavam dispersas no espaço de características, como a base Fake.BR. Os melhores resultados ocorreram com a inclusão de unigramas, bigramas e trigramas na rede de notícias, com exceção da base FakeNewsNet. Foi realizada uma análise sobre quais tipos de entidades estavam mais relacionados às classes real e falsa. Organizações, localizações, números cardinais, informações de datas e países apareceram mais associadas à classe real, conforme as classificações realizadas pelo algoritmo de propagação GNetMine.

Q7 *"Estratégias de classificação semi-supervisionadas baseadas em redes de atenção podem aumentar o desempenho de PU-LP? A integração de atenção no algoritmo PU-LP pode superar algoritmos estados da arte baseados em uma única classe na detecção de notícias falsas?"*

Foram mapeados algoritmos de classificação estado da arte que pudessem ser aplicados na abordagem AK-PULP-FND. O algoritmo escolhido foi GNEE, inicialmente proposto para classificação de eventos e seus componentes em uma rede heterogênea, baseado em regularização e redes de atenção. Aplicado ao contexto de detecção de notícias falsas, enquanto a regularização calcula *embeddings* de nós de termos com base nas notícias as quais eles estão relacionados, *Graph Attention Networks* classifica notícias considerando a estrutura do grafo, as características de termos aprendidas e o conjunto de notícias reais e falsas rotulados por PU-LP, além de atribuir diferentes importâncias aos nós vizinhos no processo de classificação. A abordagem AK-PULP-FND superou os resultados anteriores e de outros algoritmos estado da arte baseados em *One-class Graph Neural Networks*. AK-PULP-FND apresentou benefícios principalmente com a utilização de 10% de notícias falsas rotuladas, com exceção da base de dados FakeNewsNet.

As *embeddings* aprendidas no novo espaço de características de baixa dimensão foram capazes de gerar uma melhor separação de notícias reais e falsas.

Q8 *"Quais as vantagens e limitações da abordagem proposta baseada em PU-LP na classificação de notícias?"*

Foi realizada uma análise aprofundada sobre as vantagens e limitações das abordagens propostas, baseadas no algoritmo PU-LP, diante do problema de detecção de notícias falsas. Foram notados alguns pontos que merecem destaque.

A detecção de notícias falsas é um problema complexo. Notícias falsas são elaboradas com a intenção de enganar um determinado público alvo, atingindo suas emoções e desejos de compartilhar a informação. Podem apresentar características distintas, se adequando ao público de diversas mídias sociais. O conteúdo da publicação pode ser considerado falso pela distorção de fatos ou pelo apelo à emoção com dados não comprovados. Estas características fazem com que notícias falsas possam ter conteúdos similares a notícias reais, o que pode desfavorecer abordagens baseadas em similaridade de conteúdo.

Foram analisados o conteúdo original de notícias falsas selecionadas aleatoriamente, bem como o conteúdo daquelas mais similares e mais dissimilares a elas. A análise mostrou que em bases de dados cujas notícias foram coletadas de forma automática, tanto em links de origem duvidosa quanto em sites de checagem de informações, a presença de expressões de domínio que passaram despercebidas no pré-processamento podem estar enviesando os resultados, aumentando o desempenho de classificadores baseados em similaridade de conteúdo.

Uma das bases de dados, a Fake.BR, foi construída manualmente pelo Núcleo Interinstitucional de Linguística Computacional da Universidade de São Paulo, com curadoria. Fake.BR possui notícias reais e falsas com conteúdos similares, já que na construção do corpus foram selecionadas notícias reais que apresentassem correspondência com o tema das falsas. Além disso, as notícias estão dispersas no espaço de características, considerando tanto a veracidade como assuntos. Nesta base, a abordagem AK-PULP-FND proporcionou F_1 *fake* de 72% com 10% de dados rotulados, que anteriormente era 62% com PULP-FND e 59% com PU-LP. Tais resultados sugerem que o uso de Yake! para seleção de termos relevantes e de GNEE na classificação contribuíram na detecção de notícias falsas a partir de uma baixa quantidade de notícias falsas rotuladas, no cenário semissupervisionado, mesmo em situações consideradas adversas.

5.2 Publicações

Durante o desenvolvimento desta tese, as contribuições obtidas foram divulgadas por meio de publicações de artigos em periódicos, além de publicações e apresentação de artigos em conferências. Tais publicações são listadas a seguir, apresentando a relação de cada uma com este trabalho e indicando aquelas que estão relacionadas às questões de pesquisa estabelecidas.

Artigos completos publicados em periódicos

DE SOUZA, MARIANA CARAVANTI; NOGUEIRA, BRUNO MAGALHÃES; ROSSI, RAFAEL GERALDELI; MARCACINI, RICARDO MARCONDES; DOS SANTOS, BRUCCE NEVES; REZENDE, SOLANGE OLIVEIRA. *A network-based positive and unlabeled learning approach for fake news detection.* Machine learning, v. 111, n. 10, p. 3549-3592, 2022 (Qualis A2).

Neste artigo foram publicados os resultados correspondentes às questões de pesquisa **Q1**, **Q2**, **Q4** e **Q5**.

GÔLO, MARCOS PAULO SILVA; **DE SOUZA, MARIANA CARAVANTI;** ROSSI, RAFAEL GERALDELI; REZENDE, SOLANGE OLIVEIRA; NOGUEIRA, BRUNO MAGALHÃES; MARCACINI, RICARDO MARCONDES. *One-class learning for fake news detection through multimodal variational autoencoders.* Engineering Applications of Artificial Intelligence, v. 122, p. 106088, 2023 (Qualis A1).

Neste trabalho, realizado em parceria com outros pesquisadores, foram avaliados modelos de representação de notícias multimodais para detecção de notícias falsas usando aprendizado de uma única classe. A autora desta tese colaborou ativamente no pré-processamento das bases de dados, na análise de trabalhos relacionados existentes na literatura e na concepção do artigo.

Capítulos de livros publicados em anais de eventos

DE SOUZA, MARIANA CARAVANTI; NOGUEIRA, BRUNO MAGALHÃES; ROSSI, RAFAEL GERALDELI; MARCACINI, RICARDO MARCONDES; REZENDE, SOLANGE OLIVEIRA. *A Heterogeneous Network-Based Positive and Unlabeled Learning Approach to Detect Fake News.* 1 ed.: Springer International Publishing, v.13074, p. 3-18, 2021 (Qualis A4).

Este trabalho foi apresentado pela autora no evento *Brazilian Conference on Intelligent Systems*. Neste artigo foi publicado o resultado correspondente à questão de pesquisa **Q2**.

Trabalhos publicados em anais de eventos

GÔLO, MARCOS PAULO SILVA; **DE SOUZA, MARIANA CARAVANTI;** ROSSI, RAFAEL GERALDELI; REZENDE, SOLANGE OLIVEIRA; NOGUEIRA, BRUNO MAGALHÃES; MARCACINI, RICARDO MARCONDES. *Learning textual representations from multiple modalities to detect fake news through one-class learning.* Proceedings of the Brazilian Symposium on Multimedia and the Web, p. 197-204, 2021 (Qualis A4).
Este trabalho foi apresentado no evento WebMedia pelo autor principal, Marcos Paulo. O conteúdo do artigo é parte do trabalho realizado no periódico publicado em 2023. A autora desta tese colaborou ativamente no pré-processamento das bases de dados, na análise de trabalhos relacionados existentes na literatura e na concepção do artigo.

Artigo submetido

Além dos trabalhos apresentados que já foram publicados, o seguinte trabalho foi submetido, visando a publicação no curto prazo.

DE SOUZA, MARIANA CARAVANTI; GÔLO, MARCOS PAULO SILVA; JORGE, ALÍPIO MÁRIO GUEDES; AMORIN, EVELIN CARVALHO FREIRE DE AMORIM; CAMPOS, RICARDO; MARCACINI, RICARDO MARCONDES; REZENDE, SOLANGE OLIVEIRA. *Keywords attention for fake news detection using few positive labels.*
[Submetido]
Neste artigo serão publicados os resultados correspondentes às questões de pesquisa **Q6**, **Q7** e **Q8**. O artigo foi escrito em parceria com pesquisadores vinculados à Universidade do Porto, universidade na qual a autora realizou um período de mobilidade de 6 meses.

5.3 Trabalhos Futuros

A abordagem semissupervisionada de aprendizado de uma única classe baseada no algoritmo PU-LP é composta por várias etapas que podem ser exploradas no âmbito de detecção de notícias falsas, com o intuito de torná-la mais precisa sobre o problema (ver Figura 8). Na fase de coleta e transformação de notícias em dados estruturados, modelos de representação estado da arte podem ser investigados, como os baseados em *transformers*, que sejam capazes de identificar características que auxiliem na discriminação de notícias reais e falsas.

Outra etapa da abordagem que pode ser investigada é relacionada a inferência de conjuntos de notícias potencialmente reais e potencialmente falsas realizada pelo algoritmo PU-LP. Nesta fase, podem ser analisadas medidas de distância que sejam mais propícias no contexto de aprendizado de uma única classe, ou até mesmo incorporadas informações adicionais relevantes que auxiliem na identificação de conteúdo verídico e falso com maior precisão. Conjuntos mais puros levariam a melhores desempenhos de classificação.

Sobre a seleção de termos relevantes, novos testes podem ser realizados com ferramentas de extração de palavras-chave ou expressões relevantes. Além disso, nós da rede que apresentem termos relacionados podem ser unidos em um mesmo nó, com o intuito de reduzir o número de arestas e complexidade da estrutura. Uma vez que as etapas anteriores retornam um conjunto de notícias reais e falsas rotuladas, uma variedade de algoritmos estado da arte podem ser analisados na detecção de notícias falsas.

Apesar do trabalho desenvolvido nesta tese ter foco na detecção de notícias falsas, a abordagem pode ser aplicada na classificação textual de problemas nos quais seja complexo rotular exemplos que representem classes não interessantes.

—

REFERÊNCIAS

ABONIZIO, H. Q.; MORAIS, J. I. de; TAVARES, G. M.; JUNIOR, S. B. Language-independent fake news detection: English, portuguese, and spanish mutual features. **Future Internet**, Multidisciplinary Digital Publishing Institute, v. 12, n. 5, p. 87, 2020. Citado na página 57.

AGGARWAL, C. C. **Machine learning for text**. [S.l.]: Springer Science & Business Media, 2018. Citado nas páginas 28, 35, 36, 37, 38, 39, 40, 44 e 89.

AGGARWAL, C. C.; LI, N. On node classification in dynamic content-based networks. In: SIAM. **SDM 2011: Proceedings of the International Conference on Data Mining**. [S.l.], 2011. p. 355–366. Citado nas páginas 27, 86 e 90.

AGGARWAL, C. C. *et al.* Neural networks and deep learning. **Springer**, Springer, v. 10, n. 978, p. 3, 2018. Citado nas páginas 37, 39 e 44.

AHMED, H.; TRAORE, I.; SAAD, S. Detection of online fake news using n-gram analysis and machine learning techniques. In: **ISDDC 2017: International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments**. [S.l.: s.n.], 2017. p. 127–138. Citado nas páginas 27, 52 e 86.

AÏMEUR, E.; AMRI, S.; BRASSARD, G. Fake news, disinformation and misinformation in social media: a review. **Social Network Analysis and Mining**, Springer, v. 13, n. 1, p. 30, 2023. Citado nas páginas 25, 57 e 63.

AKOGLU, L.; TONG, H.; KOUTRA, D. Graph based anomaly detection and description: a survey. **Data Mining and Knowledge Discovery**, Springer, v. 29, n. 3, p. 626–688, 2015. Citado na página 68.

ASGHAR, M. Z.; HABIB, A.; HABIB, A.; KHAN, A.; ALI, R.; KHATTAK, A. Exploring deep neural networks for rumor detection. **Journal of Ambient Intelligence and Humanized Computing**, Springer, p. 1–19, 2019. Citado nas páginas 57 e 60.

BARBIER, G.; FENG, Z.; GUNDECHA, P.; LIU, H. Provenance data in social media. **Synthesis Lectures on Data Mining and Knowledge Discovery**, Morgan & Claypool Publishers, v. 4, n. 1, p. 1–84, 2013. Citado na página 53.

BEKKER, J.; DAVIS, J. Learning from positive and unlabeled data: a survey. **Machine Learning**, v. 1, p. 1–45, 2020. Citado nas páginas 26, 64, 67, 73 e 83.

BELLINGER, C.; SHARMA, S.; ZAIANE, O. R.; JAPKOWICZ, N. Sampling a longer life: Binary versus one-class classification revisited. In: **LIDTA 2017: International Workshop on Learning with Imbalanced Domains: Theory and Applications**. [S.l.: s.n.], 2017. p. 64–78. Citado na página 26.

BENGIO, Y.; DUCHARME, R.; VINCENT, P. A neural probabilistic language model. **Advances in neural information processing systems**, v. 13, 2000. Citado nas páginas 40 e 41.

BHUTANI, B.; RASTOGI, N.; SEHGAL, P.; PURWAR, A. Fake news detection using sentiment analysis. In: IEEE. **IC3 2019: Twelfth International Conference on Contemporary Computing**. [S.l.], 2019. p. 1–5. Citado nas páginas 57 e 58.

BONDIELLI, A.; MARCELLONI, F. A survey on fake news and rumour detection techniques. **Information Sciences**, Elsevier, v. 497, p. 38–55, 2019. Citado nas páginas 28, 51, 52, 53, 57, 58, 60 e 63.

BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, p. 5–32, 2001. Citado na página 57.

BREUNIG, M. M.; KRIEGEL, H.-P.; NG, R. T.; SANDER, J. Lof: identifying density-based local outliers. In: **SIGMOD 2000: Proceedings of the ACM International Conference on Management of Data**. [S.l.: s.n.], 2000. p. 93–104. Citado na página 70.

BREVE, F.; ZHAO, L.; QUILES, M.; PEDRYCZ, W.; LIU, J. *et al.* Particle competition and cooperation in networks for semi-supervised learning. **IEEE Transactions on Knowledge and Data Engineering**, v. 24, n. 9, p. 1686, 2012. Citado nas páginas 26, 48 e 64.

BRUNA, J.; ZAREMBA, W.; SZLAM, A.; LECUN, Y. Spectral networks and locally connected networks on graphs. **arXiv preprint arXiv:1312.6203**, 2013. Citado na página 49.

CAMPOS, R.; MANGARAVITE, V.; PASQUALI, A.; JORGE, A.; NUNES, C.; JATOWT, A. Yake! keyword extraction from single documents using multiple local features. **Information Sciences**, Elsevier, v. 509, p. 257–289, 2020. Citado nas páginas 86, 114 e 116.

CAPUANO, N.; FENZA, G.; LOIA, V.; NOTA, F. D. Content based fake news detection with machine and deep learning: a systematic review. **Neurocomputing**, Elsevier, 2023. Citado nas páginas 28, 57, 58 e 59.

CARVALHO, A.; FACELI, K.; LORENA, A.; GAMA, J. Inteligência artificial–uma abordagem de aprendizado de máquina. **Rio de Janeiro: LTC**, 2011. Citado na página 57.

CASTILLO, C.; MENDOZA, M.; POBLETE, B. Information credibility on twitter. In: **WWW 2011: Proceedings of the 20th International Conference on World Wide Web**. [S.l.: s.n.], 2011. p. 675–684. Citado nas páginas 38 e 52.

CHAKRAVARTHY, S.; VENKATACHALAM, A.; TELANG, A.; AERY, M. Infosift: a novel, mining-based framework for document classification. **International Journal of Next-Generation Computing**, p. 84–122, 2014. Citado nas páginas 86 e 90.

CHALAPATHY, R.; CHAWLA, S. Deep learning for anomaly detection: A survey. **arXiv preprint arXiv:1901.03407**, 2019. Citado na página 68.

CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: ACM. **SIGKDD 2016: Proceedings of the International Conference on Knowledge Discovery and Data Mining**. [S.l.], 2016. p. 785–794. Citado na página 57.

CHEN, W.; YEO, C. K.; LAU, C. T.; LEE, B. S. Behavior deviation: An anomaly detection view of rumor preemption. In: IEEE. **IEMCON 2016: Information Technology, Electronics and Mobile Communication Conference**. [S.l.], 2016. p. 1–7. Citado na página 62.

COVER, T.; HART, P. Nearest neighbor pattern classification. **IEEE transactions on information theory**, IEEE, v. 13, n. 1, p. 21–27, 1967. Citado na página 57.

CUI, B.; MA, K.; LI, L.; ZHANG, W.; JI, K.; CHEN, Z.; ABRAHAM, A. Intra-graph and inter-graph joint information propagation network with third-order text graph tensor for fake news detection. **Applied Intelligence**, Springer, p. 1–18, 2023. Citado na página 61.

DATTA, P. **Characteristic concept representations**. [S.l.]: University of California, Irvine, 1997. Citado nas páginas 62 e 69.

DEEPAK, P.; CHAKRABORTY, T.; LONG, C. *et al.* **Data Science for Fake News: Surveys and Perspectives**. [S.l.]: Springer Nature, 2021. v. 42. Citado nas páginas 26, 27, 85 e 90.

DEFFERRARD, M.; BRESSON, X.; VANDERGHEYNST, P. Convolutional neural networks on graphs with fast localized spectral filtering. **Advances in neural information processing systems**, v. 29, 2016. Citado na página 49.

DELALLEAU, O.; BENGIO, Y.; ROUX, N. L. Efficient non-parametric function induction in semi-supervised learning. In: **AISTATS 2005: International Conference on Artificial Intelligence and Statistics**. [S.l.: s.n.], 2005. v. 27, n. 28. Citado na página 79.

DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018. Citado na página 47.

DHIMAN, P.; KAUR, A.; IWENDI, C.; MOHAN, S. K. A scientometric analysis of deep learning approaches for detecting fake news. **Electronics**, MDPI, v. 12, n. 4, p. 948, 2023. Citado na página 59.

DONG, M.; YAO, L.; WANG, X.; BENATALLAH, B.; ZHANG, X.; SHENG, Q. Z. Dual-stream self-attentive random forest for false information detection. In: IEEE. **IJCNN 2019: International Joint Conference on Neural Networks**. [S.l.], 2019. p. 1–8. Citado nas páginas 57, 58 e 59.

ELTANBOULY, S.; BASHENDY, M.; ALNAIMI, N.; CHKIRBENE, Z.; ERBAD, A. Machine learning techniques for network anomaly detection: A survey. In: IEEE. **ICIoT 2020: International Conference on Informatics, IoT, and Enabling Technologies**. [S.l.], 2020. p. 156–162. Citado na página 68.

ENGELEN, J. E. V.; HOOS, H. H. A survey on semi-supervised learning. **Machine Learning**, Springer, v. 109, n. 2, p. 373–440, 2020. Citado nas páginas 26 e 64.

FAUSTINI, P.; COVÕES, T. F. Fake news detection using one-class classification. In: IEEE. **BRACIS 2019: Brazilian Conference on Intelligent Systems**. [S.l.], 2019. p. 592–597. Citado nas páginas 26, 54, 56, 62, 63, 84 e 107.

FAUSTINI, P. H. A.; COVÕES, T. F. Fake news detection in multiple platforms and languages. **Expert Systems with Applications**, Elsevier, 2020. Citado na página 57.

FELDMAN, R.; SANGER, J. *et al.* **The text mining handbook: advanced approaches in analyzing unstructured data**. [S.l.]: Cambridge university press, 2007. Citado nas páginas 38 e 90.

GAMMERMAN, A.; VOVK, V.; VAPNIK, V. Learning by transduction. In: MORGAN KAUFMANN PUBLISHERS INC. **UAI 1998: Conference on Uncertainty in Artificial Intelligence**. [S.l.], 1998. p. 148–155. Citado na página 79.

GOLBECK, J.; MAURIELLO, M.; AUXIER, B.; BHANUSHALI, K. H.; BONK, C.; BOUZAGHRANE, M. A.; BUNTAIN, C.; CHANDUKA, R.; CHEAKALOS, P.; EVERETT, J. B. *et al.* Fake news vs satire: A dataset and analysis. In: ACM. **WebSci 2018: Proceedings of the Conference on Web Science**. [S.l.], 2018. p. 17–21. Citado na página 55.

GÔLO, M.; MARCACINI, R.; ROSSI, R. Uma extensa avaliação empírica de técnicas de pré-processamento e algoritmos de aprendizado supervisionado de uma classe para classificação de texto. In: SBC. **ENIAC 2020: Encontro Nacional de Inteligência Artificial e Computacional**. [S.l.], 2020. p. 262–273. Citado nas páginas 27, 71, 84 e 86.

GROBELNIK, M. Many faces of text processing. In: **WIMS 2011: Proceedings of the International Conference on Web Intelligence, Mining and Semantics**. [S.l.: s.n.], 2011. p. 1–3. Citado na página 35.

GUACHO, G. B.; ABDALI, S.; SHAH, N.; PAPALEXAKIS, E. E. Semi-supervised content-based detection of misinformation via tensor embeddings. In: IEEE. **ASONAM 2018: ACM International Conference on Advances in Social Networks Analysis and Mining**. [S.l.], 2018. p. 322–325. Citado nas páginas 59 e 82.

HAMILTON, W.; YING, Z.; LESKOVEC, J. Inductive representation learning on large graphs. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2017. p. 1024–1034. Citado nas páginas 48, 50 e 73.

HAMMOND, D. K.; VANDERGHEYNST, P.; GRIBONVAL, R. Wavelets on graphs via spectral graph theory. **Applied and Computational Harmonic Analysis**, Elsevier, v. 30, n. 2, p. 129–150, 2011. Citado na página 49.

HAN, W.; MEHTA, V. Fake news detection in social networks using machine learning and deep learning: Performance evaluation. In: IEEE. **ICII 2019: International Conference on Industrial Internet**. [S.l.], 2019. p. 375–380. Citado na página 57.

HASSAN, N.; GOMAA, W.; KHORIBA, G.; HAGGAG, M. Credibility detection in twitter using word n-gram analysis and supervised machine learning techniques. **International Journal of Intelligent Engineering and Systems**, 2020. Citado nas páginas 27, 52 e 86.

HAWKINS, S.; HE, H.; WILLIAMS, G.; BAXTER, R. Outlier detection using replicator neural networks. In: SPRINGER. **DaWaK 2002: International Conference on Data Warehousing and Knowledge Discovery**. [S.l.], 2002. p. 170–180. Citado na página 66.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, MIT press, v. 9, n. 8, p. 1735–1780, 1997. Citado na página 44.

HONNIBAL, M.; MONTANI, I. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear. 2017. Citado na página 119.

HORNE, B. D.; ADALI, S. This just in: fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In: AAAI. **CWSM 2017: Conference on Web and Social Media**. [S.l.], 2017. Citado nas páginas 38, 52, 54 e 55.

HUA, Z.; YANG, Y.; QIU, H. Node influence-based label propagation algorithm for semi-supervised learning. **Neural Computing and Applications**, Springer, v. 33, n. 7, p. 2753–2768, 2021. Citado nas páginas 64, 86 e 92.

INAN, E. Zoka: a fake news detection method using edge-weighted graph attention network with transfer models. **Neural Computing and Applications**, Springer, v. 34, n. 14, p. 11669–11677, 2022. Citado nas páginas 52 e 61.

JASKIE, K.; SPANIAS, A. Positive and unlabeled learning algorithms and applications: A survey. In: IEEE. **IISA 2019: International Conference on Information, Intelligence, Systems and Applications**. [S.l.], 2019. p. 1–8. Citado nas páginas 73, 74, 75 e 83.

JI, M.; SUN, Y.; DANILEVSKY, M.; HAN, J.; GAO, J. Graph regularized transductive classification on heterogeneous information networks. In: SPRINGER. **PKDD 2010: Joint European Conference on Machine Learning and Knowledge Discovery in Databases**. [S.l.], 2010. p. 570–586. Citado nas páginas 27, 81 e 92.

KALIYAR, R. K.; GOSWAMI, A.; NARANG, P. Fakebert: Fake news detection in social media with a bert-based deep learning approach. **Multimedia Tools and Applications**, Springer, v. 80, n. 8, p. 11765–11788, 2021. Citado na página 59.

KALIYAR, R. K.; GOSWAMI, A.; NARANG, P.; SINHA, S. Fndnet–a deep convolutional neural network for fake news detection. **Cognitive Systems Research**, Elsevier, v. 61, p. 32–44, 2020. Citado nas páginas 57 e 58.

KANG, S.; HWANG, J.; YU, H. Multi-modal component embedding for fake news detection. In: IEEE. **IMCOM 2020: International Conference on Ubiquitous Information Management and Communication**. [S.l.], 2020. p. 1–6. Citado nas páginas 52 e 58.

KANNAN, R.; WOO, H.; AGGARWAL, C. C.; PARK, H. Outlier detection for text data: An extended version. **arXiv preprint arXiv:1701.01325**, 2017. Citado na página 66.

KATZ, L. A new status index derived from sociometric analysis. **Psychometrika**, Springer, v. 18, n. 1, p. 39–43, 1953. Citado nas páginas 26, 76, 77 e 89.

KHAN, J. Y.; KHONDAKER, M. T. I.; AFROZ, S.; UDDIN, G.; IQBAL, A. A benchmark study of machine learning models for online fake news detection. **Machine Learning with Applications**, Elsevier, v. 4, p. 100032, 2021. Citado nas páginas 44 e 59.

KHAN, S. S.; MADDEN, M. G. One-class classification: taxonomy of study and review of techniques. **The Knowledge Engineering Review**, Cambridge University Press, v. 29, n. 3, p. 345–374, 2014. Citado nas páginas 25, 26, 65 e 67.

KHANDELWAL, S.; KUMAR, D. Computational fact validation from knowledge graph using structured and unstructured information. In: ACM. **COMAD 2020: Conference on Data Science and Management of Data**. [S.l.], 2020. p. 204–208. Citado nas páginas 57 e 60.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. In: BENGIO, Y.; LECUN, Y. (Ed.). **ICLR 2019: International Conference on Learning Representations**. [S.l.: s.n.], 2015. Citado na página 91.

KIPF, T. N.; WELLING, M. Semi-supervised classification with graph convolutional networks. **arXiv preprint arXiv:1609.02907**, 2016. Citado nas páginas 48, 49, 50 e 73.

KSIENIEWICZ, P.; ZYBLEWSKI, P.; BOREK-MARCINIEC, W.; KOZIK, R.; CHORAŚ, M.; WOŹNIAK, M. Alphabet flatting as a variant of n-gram feature extraction method in ensemble classification of fake news. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 120, p. 105882, 2023. Citado nas páginas 51 e 52.

LANGLEY, P.; IBA, W.; THOMPSON, K. *et al.* An analysis of bayesian classifiers. In: CITESEER. **Aaai**. [S.l.], 1992. v. 90, p. 223–228. Citado na página 57.

LAZHAR, F. Fuzzy clustering-based semi-supervised approach for outlier detection in big text data. **Progress in Artificial Intelligence**, Springer, v. 8, n. 1, p. 123–132, 2019. Citado nas páginas 71 e 84.

LE, Q.; MIKOLOV, T. Distributed representations of sentences and documents. In: **ICML 2014: International Conference on Machine Learning**. [S.l.: s.n.], 2014. p. 1188–1196. Citado nas páginas 41, 42, 43, 44, 86 e 88.

LI, H.; CHEN, Z.; LIU, B.; WEI, X.; SHAO, J. Spotting fake reviews via collective positive-unlabeled learning. In: IEEE. **ICDM 2014: IEEE International Conference on Data Mining**. [S.l.], 2014. p. 899–904. Citado na página 73.

LI, Q.; HU, Q.; LU, Y.; YANG, Y.; CHENG, J. Multi-level word features based on cnn for fake news detection in cultural communication. **Personal and Ubiquitous Computing**, Springer, p. 1–14, 2019. Citado nas páginas 52 e 58.

LI, X.; LIU, B. Learning to classify texts using positive and unlabeled data. In: **IJCAI 2003: International Joint Conferences on Artificial Intelligence**. [S.l.: s.n.], 2003. v. 3, n. 2003, p. 587–592. Citado nas páginas 17, 74, 75, 84 e 91.

LIU, B.; LEE, W. S.; YU, P. S.; LI, X. Partially supervised classification of text documents. In: **ICML 2002: International Conference on Machine Learning**. [S.l.: s.n.], 2002. v. 2, p. 387–394. Citado nas páginas 75 e 84.

LIU, X.; NOURBAKHSH, A.; LI, Q.; FANG, R.; SHAH, S. Real-time rumor debunking on twitter. In: ACM. **CIKM 2015: Proceedings of the Conference on Information and Knowledge Management**. [S.l.], 2015. p. 1867–1870. Citado na página 56.

LIU, Y.; WU, Y.-F. B. Fned: a deep network for fake news early detection on social media. **TOIS 2020: Transactions on Information Systems**, ACM, v. 38, n. 3, p. 1–33, 2020. Citado nas páginas 26, 62, 63 e 84.

LÜ, L.; JIN, C.-H.; ZHOU, T. Similarity index based on local paths for link prediction of complex networks. **Physical Review E**, APS, v. 80, n. 4, p. 046122, 2009. Citado nas páginas 76, 77 e 89.

MA, J.; GAO, W.; MITRA, P.; KWON, S.; JANSEN, B. J.; WONG, K.-F.; CHA, M. Detecting rumors from microblogs with recurrent neural networks. AAAI Press, 2016. Citado na página 56.

MA, J.; GAO, W.; WONG, K.-F. Rumor detection on twitter with tree-structured recursive neural networks. In: . [S.l.]: Association for Computational Linguistics, 2018. Citado na página 56.

MA, S.; ZHANG, R. Pu-lp: A novel approach for positive and unlabeled learning by label propagation. In: IEEE. **ICMEW 2017: International Conference on Multimedia & Expo Workshops**. [S.l.], 2017. p. 537–542. Citado nas páginas 26, 73, 76, 77, 78, 85, 90 e 91.

MAATEN, L. Van der; HINTON, G. Visualizing data using t-sne. **Journal of machine learning research**, v. 9, n. 11, 2008. Citado na página 97.

MALLAT, S. **A wavelet tour of signal processing**. [S.l.]: Elsevier, 1999. Citado na página 49.

MANEVITZ, L.; YOUSEF, M. One-class document classification via neural networks. **Neurocomputing**, Elsevier, v. 70, n. 7-9, p. 1466–1481, 2007. Citado nas páginas 68, 72, 84, 86 e 91.

MANEVITZ, L. M.; YOUSEF, M. One-class svms for document classification. **Journal of Machine Learning Research**, v. 2, n. Dec, p. 139–154, 2001. Citado nas páginas 69, 71, 73, 84 e 91.

MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. **Introduction to information retrieval**. [S.l.]: Cambridge university press, 2008. Citado nas páginas 38 e 90.

MARTINČIĆ-IPŠIĆ, S.; MILIČIĆ, T.; TODOROVSKI, L. The influence of feature representation of text on the performance of document classification. **Applied Sciences**, Multidisciplinary Digital Publishing Institute, v. 9, n. 4, p. 743, 2019. Citado na página 88.

MATTOS, J. P. R.; MARCACINI, R. M. Semi-supervised graph attention networks for event representation learning. In: IEEE. **ICDM 2021: International Conference on Data Mining**. [S.l.], 2021. p. 1234–1239. Citado nas páginas 27, 86, 122 e 124.

MEEL, P.; VISHWAKARMA, D. K. Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities. **Expert Systems with Applications**, Elsevier, 2019. Citado nas páginas 28, 57, 58 e 63.

MIHALCEA, R.; STRAPPARAVA, C.; PULMAN, S. Computational models for incongruity detection in humour. In: **CICLING 2010: International Conference on Intelligent Text Processing and Computational Linguistics**. [S.l.]: Springer, 2010. p. 364–374. Citado nas páginas 27, 52 e 86.

MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. Efficient estimation of word representations in vector space. **arXiv preprint arXiv:1301.3781**, 2013. Citado na página 40.

MIKOLOV, T.; YIH, W.-t.; ZWEIG, G. Linguistic regularities in continuous space word representations. In: **NAACL 2013: Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**. [S.l.: s.n.], 2013. p. 746–751. Citado nas páginas 40, 41 e 42.

MISHRA, S.; SHUKLA, P.; AGARWAL, R. Analyzing machine learning enabled fake news detection techniques for diversified datasets. **Wireless Communications and Mobile Computing**, Hindawi, v. 2022, 2022. Citado nas páginas 25 e 63.

MONTEIRO, R.; SANTOS, R.; PARDO, T.; ALMEIDA, T. de; RUIZ, E.; VALE, O. Contributions to the study of fake news in portuguese: New corpus and automatic detection results. In: **PROPOR 2018: International Conference on Computational Processing of the Portuguese Language**. [S.l.]: Springer, 2018. p. 324–334. Citado na página 87.

MONTI, F.; FRASCA, F.; EYNARD, D.; MANNION, D.; BRONSTEIN, M. M. Fake news detection on social media using geometric deep learning. **arXiv preprint arXiv:1902.06673**, 2019. Citado na página 58.

MUIR, A. **Lean Six Sigma Statistics: Calculating Process Efficiencies in Transactional Project**. [S.l.]: McGraw Hill professional – Six sigma operational methods series, 2005. Citado na página 91.

NASEEM, U.; RAZZAK, I.; KHAN, S. K.; PRASAD, M. A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models. **Transactions on Asian and Low-Resource Language Information Processing**, ACM New York, NY, v. 20, n. 5, p. 1–35, 2021. Citado na página 40.

NI, S.; LI, J.; KAO, H.-Y. Mvan: Multi-view attention networks for fake news detection on social media. **IEEE Access**, IEEE, v. 9, p. 106907–106917, 2021. Citado na página 61.

NOGUEIRA, B. M.; MOURA, M. F.; CONRADO, M. S.; REZENDE, S. O. Avaliação de métodos não-supervisionados de seleção de atributos para mineração de textos. In: **WTI 2008: Workshop on Web and Text Intelligence**. [S.l.]: São Carlos: ICMC/USP, 2008. p. 59–66. Citado na página 37.

OLIVEIRA, N. R. D.; MEDEIROS, D. S.; MATTOS, D. M. A sensitive stylistic approach to identify fake news on social networking. **IEEE Signal Processing Letters**, IEEE, v. 27, p. 1250–1254, 2020. Citado nas páginas 62 e 63.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in python. **Journal of Machine Learning Research**, v. 12, n. 85, p. 2825–2830, 2011. Citado na página 97.

PENNEBAKER, J. W.; BOOTH, R. J.; BOYD, R. L.; FRANCIS, M. E. **Linguistic Inquiry and Word Count: Liwc 2015. Pennebaker Conglomerates**. [S.l.]: Inc, 2015. Citado na página 52.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: **EMNLP 2014: Proceedings of the conference on empirical methods in natural language processing**. [S.l.: s.n.], 2014. p. 1532–1543. Citado na página 43.

PERERA, P.; OZA, P.; PATEL, V. M. One-class classification: A survey. **arXiv preprint arXiv:2101.03064**, 2021. Citado nas páginas 25 e 26.

PÉREZ-ROSAS, V.; KLEINBERG, B.; LEFEVRE, A.; MIHALCEA, R. Automatic detection of fake news. **arXiv preprint arXiv:1708.07104**, 2017. Citado nas páginas 27, 38, 51, 52 e 86.

PIMENTEL, T.; MONTEIRO, M.; VIANA, J.; VELOSO, A.; ZIVIANI, N. A generalized active learning approach for unsupervised anomaly detection. **Stat**, v. 1050, p. 23, 2018. Citado na página 68.

PISKORSKI, J.; STEFANOVITCH, N.; JACQUET, G.; PODAVINI, A. Exploring linguistically-lightweight keyword extraction techniques for indexing news articles in a multilingual set-up. In: **EACL 2021: Hackashop on News Media Content Analysis and Automated Report Generation**. [S.l.: s.n.], 2021. p. 35–44. Citado nas páginas 86 e 114.

PITA, M.; PAPPA, G. L. Strategies for short text representation in the word vector space. In: IEEE. **BRACIS 2019: Brazilian Conference on Intelligent Systems**. [S.l.], 2018. p. 266–271. Citado na página 88.

PLATT, J. C.; SHAWE-TAYLOR, J.; SMOLA, A. J.; WILLIAMSON, R. C. *et al.* Estimating the support of a high-dimensional distribution. **MSR 1999: Technical Report of the Microsoft Research**, Citeseer, 1999. Citado na página 62.

PODDAR, K.; UMADEVI, K. *et al.* Comparison of various machine learning models for accurate detection of fake news. In: IEEE. **I-PACT 2019: Innovations in Power and Advanced Computing Technologies**. [S.l.], 2019. v. 1, p. 1–5. Citado na página 57.

QAZI, M.; KHAN, M. U.; ALI, M. Detection of fake news using transformer model. In: IEEE. **ICOMET 2020: International Conference on Computing, Mathematics and Engineering Technologies**. [S.l.], 2020. p. 1–6. Citado nas páginas 58 e 59.

RASHKIN, H.; CHOI, E.; JANG, J. Y.; VOLKOVA, S.; CHOI, Y. Truth of varying shades: Analyzing language in fake news and political fact-checking. In: **EMNLP 2017: Proceedings of the Conference on Empirical Methods in Natural Language Processing**. [S.l.: s.n.], 2017. p. 2931–2937. Citado na página 52.

RASOOL, T.; BUTT, W. H.; SHAUKAT, A.; AKRAM, M. U. Multi-label fake news detection using multi-layered supervised learning. In: **ICCAE 2019: Proceedings of the International Conference on Computer and Automation Engineering**. [S.l.: s.n.], 2019. p. 73–77. Citado na página 57.

REDDY, H.; RAJ, N.; GALA, M.; BASAVA, A. Text-mining-based fake news detection using ensemble methods. **International Journal of Automation and Computing**, Springer, p. 1–12, 2020. Citado nas páginas 57 e 58.

REIS, J.; CORREIA, A.; MURAI, F.; VELOSO, A.; BENEVENUTO, F. Explainable machine learning for fake news detection. In: ACM. **WebSci 2019: Proceedings of the Conference on Web Science**. [S.l.], 2019. p. 17–26. Citado nas páginas 51 e 52.

REN, Y.; WANG, B.; ZHANG, J.; CHANG, Y. Adversarial active learning based heterogeneous graph neural network for fake news detection. In: IEEE. **ICDM 2020: International Conference on Data Mining**. [S.l.], 2020. p. 452–461. Citado na página 60.

REZENDE, S. O.; PUGLIESI, J. B.; MELANDA, E. A.; PAULA, M. F. Mineração de dados. In: REZENDE, S. O. (Ed.). **Sistemas Inteligentes: Fundamentos e Aplicações**. 1. ed. [S.l.]: Manole, 2003. cap. 12, p. 307–335. Citado nas páginas 28, 35, 36 e 37.

RIBEIRO, V. H. P. **Identificação de notícias falsas em língua portuguesa**. Monografia (TCC) — Universidade Federal de Mato Grosso do Sul, 2019. Citado nas páginas 54 e 87.

ROHERA, D.; SHETHNA, H.; PATEL, K.; THAKKER, U.; TANWAR, S.; GUPTA, R.; HONG, W.-C.; SHARMA, R. A taxonomy of fake news classification techniques: Survey and implementation aspects. **IEEE Access**, IEEE, v. 10, p. 30367–30394, 2022. Citado na página 25.

ROSSI, R. G. **Classificação automática de textos por meio de aprendizado de máquina baseado em redes**. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo, 2016. Citado nas páginas 27, 37, 48, 79, 80, 81 e 92.

ROSSI, R. G.; LOPES, A. de A.; FALEIROS, T. de P.; REZENDE, S. O. Inductive model generation for text classification using a bipartite heterogeneous network. **Journal of Computer Science and Technology**, Springer, v. 29, n. 3, p. 361–375, 2014. Nenhuma citação no texto.

ROSSI, R. G.; LOPES, A. de A.; REZENDE, S. O. Optimization and label propagation in bipartite heterogeneous networks to improve transductive classification of texts. **Information Processing & Management**, Elsevier, v. 52, n. 2, p. 217–257, 2016. Citado nas páginas 26, 64, 76 e 83.

ROSSI, R. G.; REZENDE, S. O.; LOPES, A. de A. Term network approach for transductive classification. In: SPRINGER. **CICLing 2015: International Conference on Intelligent Text Processing and Computational Linguistics**. [S.l.], 2015. p. 497–515. Citado nas páginas 26, 27, 76, 85 e 86.

RUBIN, V. L.; CONROY, N.; CHEN, Y.; CORNWELL, S. Fake news or truth? using satirical cues to detect potentially misleading news. In: **Proceedings of the Workshop on Computational Approaches to Deception Detection**. [S.l.: s.n.], 2016. p. 7–17. Citado nas páginas 27, 52, 57 e 86.

RUFF, L.; ZEMLYANSKIY, Y.; VANDERMEULEN, R.; SCHNAKE, T.; KLOFT, M. Self-attentive, multi-context one-class classification for unsupervised anomaly detection on text. In: **ACL 2019: Proceedings of the Annual Meeting of the Association for Computational Linguistics**. [S.l.: s.n.], 2019. p. 4061–4071. Citado nas páginas 66, 72 e 84.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **nature**, Nature Publishing Group UK London, v. 323, n. 6088, p. 533–536, 1986. Citado na página 42.

SAIKIA, P.; GUNDALE, K.; JAIN, A.; JADEJA, D.; PATEL, H.; ROY, M. Modelling social context for fake news detection: A graph neural network based approach. In: IEEE. **IJCNN: International Joint Conference on Neural Networks**. [S.l.], 2022. p. 01–08. Citado nas páginas 52 e 61.

SALMAZZO, N. **Classificação one-class para predição de adaptação de espécies em ambientes desconhecidos**. Tese (Doutorado) — PhD thesis. Brazil: Universidade Federal do ABC, 2016. Citado na página 62.

SALTON, G. Automatic text processing: The transformation, analysis, and retrieval of. **Reading: Addison-Wesley**, 1989. Citado na página 86.

SANTIA, G. C.; WILLIAMS, J. R. Buzzface: A news veracity dataset with facebook user commentary and egos. In: AAAI. **ICWSM 2018: International Conference on Web and Social Media**. [S.l.], 2018. Citado na página 54.

SANTOS, B. N. **Classificação Transdutiva de Eventos usando Redes Heterogêneas**. Dissertação (Mestrado) — Universidade Federal de Mato Grosso do Sul, 2018. Citado na página 92.

SANTOS, R. L. de S.; PARDO, T. A. S. Fact-checking for portuguese: Knowledge graph and google search-based methods. In: SPRINGER. **PROPOR 2020: International Conference on Computational Processing of the Portuguese Language**. [S.l.], 2020. p. 195–205. Citado nas páginas 27, 51, 52, 53, 54, 55, 58, 60 e 86.

SCHEAFFER, R. L.; YOUNG, L. **Introduction to Probability and its Applications**. [S.l.]: Cengage Learning, 2009. Citado na página 68.

SCHUBERT, E.; ZIMEK, A.; KRIEGEL, H.-P. Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection. **Data Mining and Knowledge Discovery**, Springer, v. 28, n. 1, p. 190–237, 2014. Citado na página 70.

SHAHID, W.; JAMSHIDI, B.; HAKAK, S.; ISAH, H.; KHAN, W. Z.; KHAN, M. K.; CHOO, K.-K. R. Detecting and mitigating the dissemination of fake news: Challenges and future research opportunities. **IEEE Transactions on Computational Social Systems**, IEEE, 2022. Citado nas páginas 25, 57 e 63.

SHARMA, K.; QIAN, F.; JIANG, H.; RUCHANSKY, N.; ZHANG, M.; LIU, Y. Combating fake news: A survey on identification and mitigation techniques. **TIST 2019: Transactions on Intelligent Systems and Technology**, ACM, v. 10, n. 3, p. 1–42, 2019. Citado nas páginas 28, 51, 54, 57, 58 e 63.

SHI, C.; PHILIP, S. Y. **Heterogeneous Information Network Analysis and Applications**. [S.l.]: Springer, 2017. 1–227 p. (Data Analytics). Citado nas páginas 26, 27, 47, 64, 76, 83 e 85.

SHU, K.; BERNARD, H. R.; LIU, H. Studying fake news via network analysis: detection and mitigation. In: **Emerging Research Challenges and Opportunities in Computational Social Network Analysis and Mining**. [S.l.]: Springer, 2019. p. 43–65. Citado nas páginas 53 e 59.

SHU, K.; MAHUDESWARAN, D.; WANG, S.; LEE, D.; LIU, H. Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. **Big data**, Mary Ann Liebert, Inc., v. 8, n. 3, p. 171–188, 2020. Citado nas páginas 54, 55 e 87.

SHU, K.; SLIVA, A.; WANG, S.; TANG, J.; LIU, H. Fake news detection on social media: A data mining perspective. **SIGKDD 2017: International Conference on Knowledge Discovery and Data Mining - Explorations Newsletter**, ACM, v. 19, n. 1, p. 22–36, 2017. Citado na página 53.

SILVA, C. V. M.; FONTES, R. S.; JÚNIOR, M. C. Intelligent fake news detection: A systematic mapping. **Journal of Applied Security Research**, Taylor & Francis, p. 1–22, 2020. Citado nas páginas 28, 57 e 63.

SILVA, R. M.; SANTOS, R. L.; ALMEIDA, T. A.; PARDO, T. A. Towards automatically filtering fake news in portuguese. **Expert Systems with Applications**, Elsevier, v. 146, p. 113199, 2020. Citado nas páginas 107, 110, 113 e 129.

SILVERMAN, C. This analysis shows how viral fake election news stories outperformed real news on facebook. **BuzzFeed News**, v. 16, 2016. Citado na página 54.

SILVERMAN, C.; STRAPAGIEL, L.; SHABAN, H.; HALL, E.; SINGER-VINE, J. Hyper-partisan facebook pages are publishing false and misleading information at an alarming rate. **Buzzfeed News**, v. 20, 2016. Citado na página 54.

SINGH, V. K.; GHOSH, I.; SONAGARA, D. Detecting fake news stories via multimodal analysis. **Journal of the Association for Information Science and Technology**, Wiley Online Library, 2020. Citado na página 57.

SIVEK, S. C. Both facts and feelings: Emotion and news literacy. **Journal of Media Literacy Education**, ERIC, v. 10, n. 2, p. 123–138, 2018. Citado na página 25.

SIVEK, S. C. Both facts and feelings: Emotion and news literacy. **Journal of Media Literacy Education**, v. 10, n. 2, p. 123–138, 2018. Citado na página 27.

SONBHADRA, S. K.; AGARWAL, S.; NAGABHUSHAN, P. Target specific mining of covid-19 scholarly articles using one-class approach. **Chaos, Solitons & Fractals**, Elsevier, v. 140, p. 110155, 2020. Citado nas páginas 69 e 84.

SOUZA, J. V. de; JR, J. G.; FILHO, F. M. de S.; JULIO, A. M. de O.; SOUZA, J. F. de. A systematic mapping on automatic classification of fake news in social media. **Social Network Analysis and Mining**, Springer, v. 10, n. 1, p. 1–21, 2020. Citado nas páginas 57 e 60.

SOUZA, M. C. d.; NOGUEIRA, B. M.; ROSSI, R. G.; MARCACINI, R. M.; REZENDE, S. O. A heterogeneous network-based positive and unlabeled learning approach to detect fake news. In: SPRINGER. **BRACIS 2021: Brazilian Conference on Intelligent Systems**. [S.l.], 2021. p. 3–18. Citado nas páginas 26, 87 e 114.

SOUZA, M. C. de; NOGUEIRA, B. M.; ROSSI, R. G.; MARCACINI, R. M.; SANTOS, B. N. D.; REZENDE, S. O. A network-based positive and unlabeled learning approach for fake news detection. **Machine Learning**, Springer, v. 111, n. 10, p. 3549–3592, 2022. Citado nas páginas 26, 87 e 114.

TACCHINI, E.; BALLARIN, G.; VEDOVA, M. L. D.; MORET, S.; ALFARO, L. de. Some like it hoax: Automated fake news detection in social networks. **arXiv preprint arXiv:1704.07506**, 2017. Citado na página 56.

TAN, P.; STEINBACH, M.; KARPATNE, A.; KUMAR, V. **Introduction to Data Mining**. [S.l.]: Pearson, 2019. (What's New in Computer Science Series). ISBN 9780133128901. Citado nas páginas 67, 68, 70, 71, 84 e 90.

TAX, D. M.; DUIN, R. P. Support vector data description. **Machine learning**, Springer, v. 54, p. 45–66, 2004. Citado na página 72.

TAX, D. M. J. One-class classification: concept-learning in the absence of counter-examples [ph. d. thesis]. **Delft University of Technology**, 2001. Citado nas páginas 25, 26, 65, 66 e 67.

THORNE, J.; VLACHOS, A.; CHRISTODOULOPOULOS, C.; MITTAL, A. Fever: a large-scale dataset for fact extraction and verification. **arXiv preprint arXiv:1803.05355**, 2018. Citado na página 56.

TRACY, K. **The International Encyclopedia of Language and Social Interaction, 3 Volume Set**. [S.l.]: John Wiley & Sons, 2015. Citado na página 52.

UPPAL, A.; SACHDEVA, V.; SHARMA, S. Fake news detection using discourse segment structure analysis. In: IEEE. **Confluence 2020: International Conference on Cloud Computing, Data Science & Engineering**. [S.l.], 2020. p. 751–756. Citado na página 58.

VAPNIK, V. **The nature of statistical learning theory**. [S.l.]: Springer science & business media, 2013. Citado na página 57.

VARGAS, F. A.; PARDO, T. A. S. Studying dishonest intentions in brazilian portuguese texts. **arXiv e-prints**, 2020. Citado na página 110.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, Ł.; POLOSUKHIN, I. Attention is all you need. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2017. p. 5998–6008. Citado nas páginas 45, 46 e 51.

VELICKOVIC, P.; CUCURULL, G.; CASANOVA, A.; ROMERO, A.; LIO, P.; BENGIO, Y. Graph attention networks. **stat**, v. 1050, p. 20, 2017. Citado nas páginas 48, 50, 73 e 123.

VLACHOS, A.; RIEDEL, S. Fact checking: Task definition and dataset construction. In: **ACL 2014: Proceedings of the Workshop on Language Technologies and Computational Social Science**. [S.l.: s.n.], 2014. p. 18–22. Citado nas páginas 54 e 55.

VOLKOVA, S.; JANG, J. Y. Misleading or falsification: Inferring deceptive strategies and types in online news and social media. In: **WWW 2018: Companion Proceedings of the The Web Conference**. [S.l.: s.n.], 2018. p. 575–583. Citado nas páginas 51 e 52.

WANG, H.; BAH, M. J.; HAMMAD, M. Progress in outlier detection techniques: A survey. **IEEE Access**, IEEE, v. 7, p. 107964–108000, 2019. Citado nas páginas 25, 26, 65, 68 e 70.

WANG, W. Y. "liar, liar pants on fire": A new benchmark dataset for fake news detection. **arXiv preprint arXiv:1705.00648**, 2017. Citado na página 56.

WANG, X.; JIN, B.; DU, Y.; CUI, P.; TAN, Y.; YANG, Y. One-class graph neural networks for anomaly detection in attributed networks. **Neural computing and applications**, Springer, v. 33, p. 12073–12085, 2021. Citado nas páginas 27, 72 e 124.

WANG, Y.; MA, F.; JIN, Z.; YUAN, Y.; XUN, G.; JHA, K.; SU, L.; GAO, J. Eann: Event adversarial neural networks for multi-modal fake news detection. In: ACM. **SIGKDD 2018: Proceedings of the International Conference on Knowledge Discovery & Data Mining**. [S.l.], 2018. p. 849–857. Citado na página 52.

WANG, Y.; QIAN, S.; HU, J.; FANG, Q.; XU, C. Fake news detection via knowledge-driven multimodal graph convolutional networks. In: **ICMR 2020: Proceedings of the International Conference on Multimedia Retrieval**. [S.l.: s.n.], 2020. p. 540–547. Citado nas páginas 58 e 60.

WANI, A.; JOSHI, I.; KHANDVE, S.; WAGH, V.; JOSHI, R. Evaluating deep learning approaches for covid19 fake news detection. In: SPRINGER. **CONSTRAINT 2021: Combating Online Hostile Posts in Regional Languages during Emergency Situation: International Workshop**. [S.l.], 2021. p. 153–163. Citado na página 59.

WEISS, S. M.; INDURKHYA, N.; ZHANG, T.; DAMERAU, F. J. **Text Mining - Predictive Methods for Analizing Unstructured Information**. [S.l.]: Springer Science+Business Media, Inc., 2005. Citado na página 36.

WILSON, T.; WIEBE, J.; HOFFMANN, P. Recognizing contextual polarity in phrase-level sentiment analysis. In: **EMNLP 2005: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing**. [S.l.: s.n.], 2005. p. 347–354. Citado na página 52.

WU, L.; RAO, Y.; NAZIR, A.; JIN, H. Discovering differential features: Adversarial learning for information credibility evaluation. **Information Sciences**, Elsevier, v. 516, p. 453–473, 2020. Citado na página 58.

WU, Z.; PAN, S.; CHEN, F.; LONG, G.; ZHANG, C.; PHILIP, S. Y. A comprehensive survey on graph neural networks. **IEEE Transactions on Neural Networks and Learning Systems**, IEEE, v. 32, n. 1, p. 4–24, 2020. Citado nas páginas 48, 49 e 50.

XU, W.; WU, J.; LIU, Q.; WU, S.; WANG, L. Evidence-aware fake news detection with graph neural networks. In: ACM. **WWW 2022: Web Conference**. [S.l.], 2022. p. 2501–2510. Citado na página 61.

XU, Y.; LI, L.; HUANG, J.; YIN, Y.; SHAO, W.; MAI, Z.; HEI, L. Positive-unlabeled learning for sentiment analysis with adversarial training. In: SPRINGER. **CollaborateCom 2019: International Conference on Collaborative Computing: Networking, Applications and Worksharing**. [S.l.], 2019. p. 364–379. Citado nas páginas 75, 76 e 84.

YAN, D.; LI, K.; GU, S.; YANG, L. Network-based bag-of-words model for text classification. **IEEE Access**, IEEE, v. 8, p. 82641–82652, 2020. Citado nas páginas 27, 86 e 90.

YANG, C.; XIAO, Y.; ZHANG, Y.; SUN, Y.; HAN, J. Heterogeneous network representation learning: A unified framework with survey and benchmark. **TKDE 2020: Transactions on Knowledge and Data Engineering**, IEEE, 2020. Citado nas páginas 26, 27, 47, 64, 76 e 85.

YANG, S.; SHU, K.; WANG, S.; GU, R.; WU, F.; LIU, H. Unsupervised fake news detection on social media: A generative approach. In: **AAAI 2019: Proceedings of the Conference on Artificial Intelligence**. [S.l.: s.n.], 2019. v. 33, p. 5644–5651. Citado na página 60.

YAVARY, A.; SAJEDI, H.; ABADEH, M. S. Information verification in social networks based on user feedback and news agencies. **Social Network Analysis and Mining**, Springer, v. 10, n. 1, p. 2, 2020. Citado nas páginas 57, 58 e 60.

YU, J.; HUANG, Q.; ZHOU, X.; SHA, Y. Iarnet: An information aggregating and reasoning network over heterogeneous graph for fake news detection. In: IEEE. **IJCNN 2020: International Joint Conference on Neural Networks**. [S.l.], 2020. p. 1–9. Citado na página 60.

YU, S.; LI, C. PE-PUC: A graph based PU-Learning approach for text classification. In: SPRINGER. **mldm 2007: International Workshop on Machine Learning and Data Mining in Pattern Recognition**. [S.l.], 2007. p. 574–584. Citado nas páginas 76 e 84.

ZHANG, C.; REN, D.; LIU, T.; YANG, J.; GONG, C. Positive and unlabeled learning with label disambiguation. In: **IJCAI 2019: International Joint Conference on Artificial Intelligence**. [S.l.: s.n.], 2019. p. 1–7. Citado na página 73.

ZHANG, J.; DONG, B.; PHILIP, S. Y. Deep diffusive neural network based fake news detection from heterogeneous social networks. In: IEEE. **Big Data 2019: International Conference on Big Data**. [S.l.], 2019. p. 1259–1266. Citado nas páginas 57, 58 e 59.

ZHANG, X.; GHORBANI, A. A. An overview of online fake news: Characterization, detection, and discussion. **Information Processing & Management**, Elsevier, v. 57, n. 2, p. 102025, 2020. Citado nas páginas 28, 57 e 63.

ZHAO, J.; CAO, N.; WEN, Z.; SONG, Y.; LIN, Y.-R.; COLLINS, C. # fluxflow: Visual analysis of anomalous information spreading on social media. **Transactions on Visualization and Computer Graphics**, IEEE, v. 20, n. 12, p. 1773–1782, 2014. Citado na página 62.

ZHAO, Z.; ZHAO, J.; SANO, Y.; LEVY, O.; TAKAYASU, H.; TAKAYASU, M.; LI, D.; WU, J.; HAVLIN, S. Fake news propagates differently from real news even at early stages of spreading. **EPJ Data Science**, Springer Berlin Heidelberg, v. 9, n. 1, p. 7, 2020. Citado na página 53.

ZHOU, D.; BOUSQUET, O.; LAL, T. N.; WESTON, J.; SCHÖLKOPF, B. Learning with local and global consistency. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2004. p. 321–328. Citado na página 80.

ZHOU, J.; CUI, G.; HU, S.; ZHANG, Z.; YANG, C.; LIU, Z.; WANG, L.; LI, C.; SUN, M. Graph neural networks: A review of methods and applications. **AI open**, Elsevier, v. 1, p. 57–81, 2020. Citado nas páginas 48, 49 e 50.

ZHOU, L.; BURGOON, J. K.; TWITCHELL, D. P.; QIN, T.; JR, J. F. N. A comparison of classification methods for predicting deception in computer-mediated communication. **Journal of Management Information Systems**, Taylor & Francis, v. 20, n. 4, p. 139–166, 2004. Citado nas páginas 52 e 92.

ZHOU, X.; ZAFARANI, R.; SHU, K.; LIU, H. Fake news: Fundamental theories, detection strategies and challenges. In: ACM. **WSDM 2019: Proceedings of the International Conference on Web Search and Data Mining**. [S.l.], 2019. p. 836–837. Citado na página 28.

ZHU, X.; GHAHRAMANI, Z.; LAFFERTY, J. D. Semi-supervised learning using gaussian fields and harmonic functions. In: **ICML 2003: Proceedings of the International Conference on Machine Learning**. [S.l.: s.n.], 2003. p. 912–919. Citado nas páginas 80, 92, 122 e 123.

ZHU, X.; GOLDBERG, A. B. **Introduction to Semi-supervised Learning**. [S.l.]: Morgan and Claypool Publishers, 2009. ISBN 1598295470, 9781598295474. Citado nas páginas 64 e 73.


ICMC
USP
SÃO CARLOS