Classificação de notícias digitais utilizando Processamento de Linguagem Natural

Guilherme da Silva Lima

Universidade Federal de Uberlândia
Faculdade de Computação
Bacharelado em Sistemas de Informação

Monte Carmelo - MG
2023

Guilherme da Silva Lima

Classificação de notícias digitais utilizando Processamento de Linguagem Natural

Trabalho de Conclusão de Curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Área de concentração: Sistemas de Informação

Orientador: Profa. Dra. Fernanda Maria da Cunha Santos

Monte Carmelo - MG
2023

Dedico este trabalho a Deus e a minha mãe por estar comigo nesta jornada e ter possibilitado que meu sonho fosse realizado.

Agradecimentos

Agradeço a Deus por esta comigo durante todos os momentos de dificuldade me dando forças para seguir em frente. Agradeço a minha mãe Maria Lucia da Silva pelo incentivo e apoio, aos meus amigos que fiz durante este período Bruna Helana, Vitor Soares, Murilo Villas Boas, Kelo Fortunato e Larissa Campos por compartilhar comigo os momentos mais especiais de toda minha vida. A minha orientadora Fernanda Maria da Cunha Santos, por me acompanhar no desenvolvimento deste trabalho, e a professora Ana Claudia Martinez por aceitar o convite para compor a banca e pela presença em minha vida acadêmica. Muito obrigado a todos vocês!

"Grandes coisas fez o Senhor por nós, pelas quais estamos alegres."
(Salmos 126:3)

Resumo

Nos últimos anos, acompanha-se a popularidade das redes sociais como meio de comunicação para divulgação de notícias importantes para a sociedade, como as relacionadas à saúde, à política, à economia e outros. No entanto, as *fake news* estão incluídas nas notícias que também circulam pelas redes sociais e, notoriamente, se propagam a uma velocidade superior as verdadeiras. Pensando nos possíveis problemas morais, sociais e econômicos que as fake news podem atingir a população, ferramentas computacionais e estudos promissores estão surgindo com o objetivo de identificar quais notícias são fake e quais não são, por meio de aplicativos constituídos pelas técnicas de Processamento de Linguagem Natural e pelas de Aprendizado de Máquinas. Assim, o objetivo deste trabalho é construir um modelo computacional destinado à classificação de notícias digitais falsas para o português do Brasil usando técnicas de Processamento de Linguagem Natural (PLN) juntamente com a árvore de decisão como algoritmo de Aprendizado de Máquinas (AM). Os resultados foram satisfatórios, porém ideias complementares e mais amostras para corpus aperfeiçoarão o modelo proposto.

Palavras-chave: Processamento de Linguagem Natural, Fake News, Aprendizado de Máquina, Word2Vec.

Lista de ilustrações

Lista de tabelas

Lista de siglas

AM Aprendizado de Máquinas

BOW Bag-of-Words

CBOW Continuos Bag-of-Words

HAN Hierarchical Attention Networks

PLN Processamento de Linguagem Natural

TF-IDF Term Frequency-inverse Document Frequency

Sumário

Capítulo 1

Introdução

1.1 Motivação e Contextualização

Fake News são notícias falsas propositalmente espalhadas com o objetivo de enganar as pessoas e favorecer um determinado grupo (PARALELO, 2022). Antigamente, as fake news eram divulgadas "boca a boca" ou através de papéis informais impressos. Atualmente, as redes sociais e aplicativos da Internet são os meios de divulgação destas que alcançam as pessoas numa rapidez inimaginável. Estudos apontam que notícias falsas possuem uma probabilidade maior de ser compartilhada do que conteúdo verdadeiro. Experimentos realizados na plataforma Twitter mostram que as informações falsas têm 70% mais chances de serem compartilhadas e que se propagam mais rápida e profundamente pela web (SHARIATMADARI, 2019).

Os principais assuntos envolvidos nas fake news são de ordem política, econômica, temas da saúde coletiva e sobre a vida pessoal de celebridades. As estratégias das fake news são alcançar as pessoas pela desinformação ou pela manipulação emocional. A linha da desinformação é uma estratégia muito utilizada por serviços de inteligência para enfraquecer os seus inimigos e atingir moralmente um grupo de pessoas. Já a manipulação emocional ocorre por uma linguagem diferente daquela usada em textos reais, cuja escrita tem a intenção de provocar fortes reações emocionais no leitor, sentimentos como raiva, indignação e frustração. Estes sentimentos dificultam a capacidade do leitor em analisar racionalmente o conteúdo e provocam a urgência de compartilhar a informação (SHARIATMADARI, 2019).

Portanto, pode-se considerar que as fake news são escritas de forma diferente do conteúdo puramente jornalístico onde foca-se somente nos fatos (SHARIATMADARI, 2019). A partir disso, acadêmicos e cientistas de dados estão explorando e aperfeiçoando técnicas de Processamento de Linguagem Natural (PLN), juntamente com os algoritmos de Aprendizado de Máquina (AM) para que possam identificar as diferenças de padrões de escrita entre as notícias digitais falsas e verdadeiras e usá-los para classificar corretamente uma notícia.

Assim, para o conjunto PLN e AM, faz-se necessário um corpus de textos escritos na linguagem português para treinar o modelo computacional, o qual classificará os textos entre verdadeiros ou fake news. Destaca-se o corpus Fake.br, que contém notícias previamente classificadas em conteúdo verdadeiro ou falso, desenvolvido por pesquisadores do Núcleo Interinstitucional de Linguística Computacional (MONTEIRO et al., 2018).

1.2 Objetivos da Pesquisa

Este trabalho tem como objetivo principal criar um modelo computacinal para classificar notícias digitais em português brasileiro e ser capaz de identificar notícias falsas utilizando de técnicas de PLN e da Àrvore de Decisão, como algoritmo de AM.

Sigularizando o objetivo principal em objetivos específicos, após a escolha do corpus, segue com a análise e a implementação das técnicas de PLN na fase de pré-processamentos dos textos, ou seja, a tokenização, a limpeza dos dados, a simplificação das formas léxicas para extrair informações relevantes e, por fim, a tradução de palavras em números. Na sequência, objetiva a implementação de uma Árvore de Decisão como parte final do modelo classificador.

Os resultados alcançados pelo modelo computacional serão avaliados pelas medidas de avaliação: acurácia, *recall*, precisão e *f1 score*.

1.3 Contribuições

Espera-se ao final do trabalho um modelo computacional capaz de classificar textos digitais em fake news ou não, usando um simplificado conjunto de técnicas de PLN na fase de pré-processamento.

1.4 Organização do Trabalho

O trabalho foi estruturado em 4 Capítulos. O Capítulo 2 apresenta o estado da arte, no qual o tema é fortalecido por uma explanação sucinta sobre fake news, PLN e Árvore de Decisão. O Capítulo 3 descreve, detalhadamente, as etapas do modelo computacional para classificação de textos digitais, os resultados alcançados e a avaliação dos mesmos. E, por fim, o Capítulo 4 contém as conclusões e trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Este capítulo apresentará, resumidamente, a fundamentação teórica que foi necessária para o desenvolvimento deste trabalho, destacando os conceitos sobre fake news, as principais técnicas de PLN implementadas no modelo computacional, Àrvore de Decisão e uma descrição dos trabalhos correlatos.

2.1 Fake News

Fake news é uma expressão popularmente usada para designar fatos e notícias falsas. As fake news vem sendo relatadas como artimanha contra o verdadeiro jornalismo, porém, os boatos, as notícias fraudulentas e as informações ludibriosas não são aspectos somente do novo mundo da comunicação, e tais distorções sempre estiveram na história de cada sociedade por meio de narrações inverídicas. Isso ocorre também nos dias atuais, porém com uma velocidade vertigionosa devido o uso da Web como o principal meio de divulgação.

Atualmente, as fake news são notícias que aparentam ser verdadeiras, que em algum grau poderiam ser verdade ou que remontam situações para tentar se mostrar confiáveis. As notícias falsas não são apenas aquelas extremamente irônicas, que têm o intuito de serem engraçadas e provocar o leitor. As notícias falsas atualmente buscam disseminar boatos e inverdades com informações que não estão 100% corretas, por isso o cuidado ao analisar uma notícia (MERELES C.; MORAES, 2017). Um profissional em jornalismo faz a identificação de notícias falsas manualmente, decorrente de seu conhecimento, porém o foco deste estudo é a identificação automática por meio de um modelo computacional.

Sharma et al. (2019) relataram três características relevantes para a identificação de notícias falsas: as fontes da notícia; o conteúdo da informação; e a resposta do usuário ao receber a notícia em redes sociais (SHARMA et al., 2019). A característica mais relevante para um modelo computacional é o conteúdo da informação. Isso se deve ao comportamento do somatório das frequências relativas das palavras usadas nas notícias reais serem superiores ao das notícias falsas (OLIVEIRA et al., 2020). Além disso, Sharma et al.

(2019) destacou que notícias falsas tendem apresentarem menor complexidade cognitiva, menos palavras exclusivas, mais palavras de emoção negativa e mais palavras de ação.

2.2 Processamento de Linguagem Natural

O PLN é um campo de estudo que envolve modelos e processos computacionais para a solução de problemas de compreensão e manipulação de linguagens humanas. O PLN pode ser decomposto em cinco estágios primários (OLIVEIRA et al., 2020):

- ❏ Tokenização: tem como finalidade extrair unidades mínimas de texto, chamada de token e que, corresponde a uma palavra do texto, símbolo ou caractere de pontuação. Por exemplo: "Zico foi o maior jogador da história!"

  [Zico] [foi] [o] [maior] [jogador] [da] [história] [!]

- ❏ Análise léxica: relaciona as variantes morfológicas aos seus lemas.

- ❏ Análise sintática: foca no relacionamento das palavras entre si, cada uma assumindo seu papel estrutural nas frases, e de como as frases podem ser partes de outras, constituindo sentenças.

- ❏ Análise semântica: destinado ao significado de palavras, expressões fixas e sentenças inteiras.

- ❏ Análise pragmática: busca compreender uma determinada frase, observando referências pronominais e a coerência textual da estrutura das frases adjacentes.

2.2.1 *Word Embeddings*

Word Embedding é uma das principais formas de representação textual, considerando os tokens (unidades linguísticas) no contexto e as frequências de ocorrência. No *Word Embedding* as palavras são representadas de forma matemática em um vetor. Cada palavra é representada por um ponto em um espaço multidimensional, que chamamos de *embedding space*. Um vetor de tamanho fixo é utilizado para representar cada palavra, ou, a **palavra central**, e neste vetor será armazenado os valores de semelhança entre a **palavra central** e as palavras que estão próximas. A Figura 1 ilustra a equação usada para calcular os valores do vetor e a Figura 2 esboça a dinâmica usada por um dos algoritmos do *Word Embedding*.

As *Word Embeddings* são aprendidas a partir de um grande corpus por meio de métodos de contagem ou pelas redes neurais, cujo algoritmo foi denominado Word2Vec (MIKOLOV et al., 2013). Por meio dos vetores de palavras é possível realizar cálculos espaciais para encontrar a similaridade entre palavras, pois palavras que compartilham contextos

$$J(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{\substack{-m\leq j\leq m;\\ i\neq 0}} logP(t_{i+j}|t_i)$$

Figura 1 – Equação para calcular o valor de similaridade da palavra central com as demais.

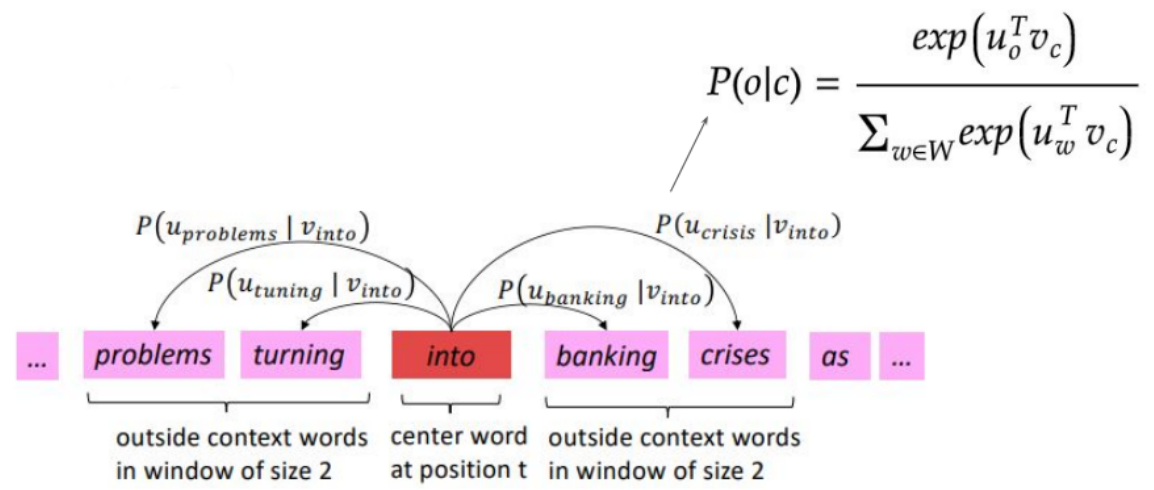

Figura 2 – Exemplo do cálculo do método CBOW.

semelhantes tendem a ter significados semelhantes. A Figura 2 exemplifica os cálculos utilizados pelo Word2Vec, usando o modelo CBOW.

O Word2Vec tornou-se um dos método mais utilizados no pré-processamento de dados em formato de textos, para realizar tarefas como a análise de sentimento, tradução de textos, reconhecimento de entidades nomeadas, ou até mesmo geração de textos carácter a carácter ou palavra a palavra.

2.3 Aprendizado de Máquina

Aprendizado de Máquina são técnicas que abordam como tornar as máquinas aptas a aprender, dada a partir de um conjunto de exemplos, e realizar métodos de inferência indutiva para obter resultados a partir deste aprendizado. Dentre essas técnicas, será analisado as árvores de decisão.

Uma árvore de decisão toma como entrada um conjunto de atributos e retorna um valor de saída que significa uma decisão. Os atributos de entrada se forem valores discretos, a árvore fará uma aprendizagem de uma função destinada à classificação, e se forem valores contínuos a aprendizagem será denominada regressão.

A árvore de decisão é constituída por nós (*decision nodes*) que se relacionam entre si por uma hierarquia. Existe o nó-raiz (*root node*), que é o mais importante, os nós intermediários e os nós-folhas (*leaf nodes*), que são os resultados. O nó-raiz e os nós

intermediários são os atributos da base de dados e o nó-folha é a classe ou o valor que será gerado como resposta.

A Figura 3 ilustra a representação em forma de uma tabela de uma base de dados. A Figura 4 demonstra a representação dos dados da Figura 3 após a execução de um algoritmo de árvore de decisão.

Dia	Sol?	Vento?	Vou para praia?
1	Sim	Sim	Não
2	Sim	Sim	Não
3	Sim	Não	Sim
4	Não	Não	Não
5	Não	Sim	Não
6	Não	Sim	Não

Figura 3 – Exemplo de uma base de dados.

Fonte: https://didatica.tech/como-funciona-o-algoritmo-arvore-de-decisao

.

Figura 4 – Representação de uma árvore de decisão do exemplo da Figura 3.

Fonte: https://didatica.tech/como-funciona-o-algoritmo-arvore-de-decisao

.

Segundo os autores Russell e Norvig (2004), *"uma árvore de decisão alcança sua decisão executando uma sequência de testes. Cada nó interno na árvore corresponde a um*

teste do valor de uma das propriedades, e as ramificações a partir do nó são identificadas com os valores possívies do teste. Cada nó de folha na árvore especifica o valor a ser retornado se aquela folha for alcançada."

2.4 Trabalhos Correlatos

O tema de PLN está sendo muito abordado nos últimos anos, sendo destaque de muitos trabalhos científicos e de ferramentas comerciais, como os Chatbots. No entanto, será descrito nesta seção apenas os trabalhos fundamentais que serviram de base e de direcionamento para o desenvolvimento deste trabalho de TCC.

Os artigos Silva et al. (2020) e Monteiro et al. (2018) descrevem o corpus Fake.Br composto por notícias na liguagem Português do Brasil. Este corpus é composto por notícias digitais verdadeiras e falsas, e que foram analisadas minuciosamente com o intuito de descobrir características linguísticas das notícias. Este cuidado deve ao objetivo deste artigo que é fazer o corpus Fake.Br referência nacional para futuras pesquisas na área.

O artigo Monteiro et al. (2018), também usou técnicas tradicionais de aprendizado de máquina para obter a detecção automática de notícias falsas sob o novo corpus, alcançando bons resultados, onde o classificador obteve taxa de erro de 11,6% para textos políticos, 10,4% para TV celebridades, 18,1% para economia e 20% para religião.. As características exploradas dos textos formaram diferentes conjuntos, os quais serviram de dados de entrada para os classificadores de AM: Naive-Bayes, *Random Forest* e rede neural *Multilayer Perceptron*. Entre as técnicas de AM testadas, a rede neural *Multilayer Perceptron* obteve 90% de acurácia.

No artigo Silva et al. (2020) relata experimentos com técnicas de aprendizado de máquina, como *support vector machine*, regressão logística, árvore de decisão, *random forest*, *bootstrap aggregating (bagging)* e *adaptive boosting (Ad-aBoost)* em diferentes conjuntos de características de base linguística e de representações de texto distributiva e distribuída. As características da base linguística utilizadas foram : a pausa, emotividade, número de verbos da notícia, não imediatismo, diversidade, tamanho médio das frases, tamanho médio das palavras e número de erros ortográficos. As técnicas de representação de texto foram a tradicional Bag-of-Words (BOW) e as técnicas distribuída Word2Vec e FastText. No experimento com BOW usou o algoritmo Term Frequency-inverse Document Frequency (TF-IDF) para ajustar o peso dos tokens em cada documento, e para os algoritmos Word2Vec e FastText, foram utilizados os vetores pré-treinados proposto por Hartmann et al. (2017). Para avaliar os resultados obtidos, foi calculado os valores das medidas de avaliação: f1-score, taxa de falso positivo, *recall* e taxa de verdadeiro positivo.

O TCC Guarise (2019) propôs montar um classificador de notícias em português brasileiro capaz de identificar notícias falsas utilizando redes neurais de aprendizado profundo. O modelo implementado foi o Hierarchical Attention Networks (HAN), pois permite a

visualização dos resultados destacando as palavras e sentenças mais determinantes para a classificação através de um mapa de calor com os valores dos pesos de atenção gerados no treinamento do modelo. Inicialmente, foram realizadas alterações nos textos com objetivo de adequá-los para que um número menor de tokens não reconhecidos pelo vocabulário pré-treinado de word embeddings fosse utilizado, pois palavras não reconhecidas não adicionam informação a rede. Na sequência, os vetores gerados serviram como dados de entrada para a rede neural HAN, na qual foram realizados o treinamento com 80% da base de dados, 5760 textos, sendo 2880 notícias verdadeiras e 2880 notícias falsas.

Capítulo 3

Experimentos e Análise dos Resultados

As finalidades deste capítulo são descrever a base de dados, as etapas do modelo computacional de classificação de notícias digitais, detalhando toda a metodologia, e, principalmente, os resultados atingidos pelo modelo. A metodologia do classificador, assim como a organização deste capítulo, está esboçado na Figura 5.

A Figura 5, ilustra a metodologia aplicada neste classificador, partindo do ponto da escolha da base dados. É necessário que a mesma possua uma quantidade de notícias relevantes para o classificador e esteja organizada de modo a facilitar a aplicação das técnicas de pré-processamento e utilização do Word2Vec. O Word2Vec irá gerar os dados a serem utilizados no treinamento e teste do classificador, sendo separados através do algoritmo K-fold. Com esses resultados serão possíveis aplicar a Árvore de Decisão e, como consequência, gerar os resultados.

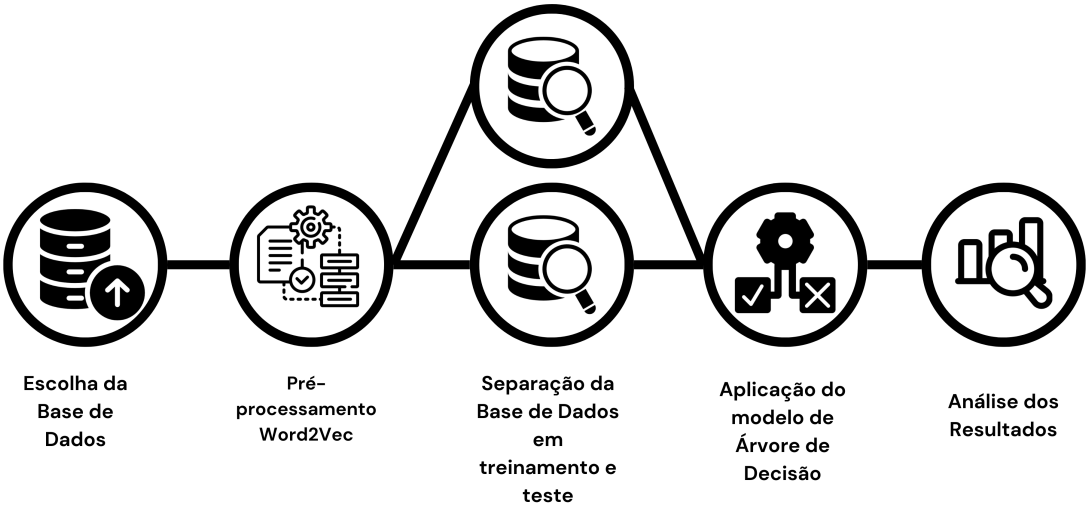

Figura 5 – Fluxograma das etapas do modelo para classificação de notícias digitais.

Fonte: Autoria própria.

3.1 Base de Dados

A base de dados Fake.Br foi um trabalho desenvolvido por pesquisadores do Núcleo Interinstitucional de Linguística Computacional (MONTEIRO et al., 2018). O corpus Fake.br contém notícias previamente classificadas em conteúdo verdadeiro ou falso. A classificação das notícias falsas ocorreu de forma manual e das notícias verdadeiras de forma semi-automática. De maneira geral, observa-se que as notícias totalizaram em 7.200, sendo 3.600 verdadeiras e 3.600 falsas, embora as notícias verdadeiras sejam mais extensas que as falsas. Assim sendo, o corpus é composto por textos simples, separadas em arquivos distintos, e organizado nos diretórios:

- pasta full_texts, que contém os textos completos, conforme coletados em seus sites. Dentro desta pasta, existem mais 4 pastas:

  pasta fake: contém as fake news coletadas;

  pasta true: contém as notícias verdadeiras coletadas;

  pasta fake-meta-information: contém as informações de metadados de cada fake news;

  pasta true-meta-information: contém as informações de metadados de cada true news;

Os temas das notícias que compõem o corpus Fake.Br foram divididos em categorias, segundo a Tabela 1.

Tabela 1 – Quantidade de notícias por categoria no corpus Fake.Br.

Categoria	**Quantidade**	**Percentual (%)**
Política	4.180	58.0
TV e Celebridades	1.544	21.4
Sociedade e Atualidades	1.276	17.7
Sociedade e Tecnologia	112	1.5
Economia	44	0.7
Religião	44	0.7

Fonte: Monteiro et al. (2018).

3.2 Linguagem de Programação usada no Modelo

Diferente de muitas linguagens o Python possui particularidades que facilitam o desenvolvimento de algoritmos, como por exemplo ser uma linguagem de alto nível com tipagem forte e dinâmica, e também ser multi-paradigma, permitindo programação orientada a objetos, procedural e funcional. A linguagem de programação Python foi a

linguagem escolhida para implementar o modelo de classificação devido a grande quantidade de bibliotecas para algoritmos de AM (PYTHON, 2022). São elas:

- Natural Language Toolkit (NLTK): para todas as funções de PLN (PEDREGOSA et al., 2011);
- Gensim: para a execução do algoritmo word2vec.
- Scikit-learn: para a definição das funções de validação cruzada e da árvore de decisão (PEDREGOSA et al., 2011).

Com as facilidades de se desenvolver algoritmos em Python combinado com um conjunto de bibliotecas destinadas ao PLN, o Python torna-se ideal para a implementação deste modelo.

3.3 Pré-processamento

3.3.1 Etapas do Processamento de Linguagem Natural

Antes qualquer etapa possa ser iniciada é necessário fazer o pré-processamento do texto que será usado. Nesta etapa, o objetivo é limpar o texto, removendo os ruídos (pontos, caracteres especiais, etc), palavras repetidas e aquelas com pouco valor para a classificação do texto.

As funções de PLN implementadas no modelo computacional de classificação de notícias foram:

- Normalização

  A normalização abrange tratativas como a tokenização, transformação de letras maiúsculas para minúsculas e limitação de tamanho entre as amostras textuais analisadas.

- Tokenização

  O processo de tokenização tem como objetivo separar palavras ou sentenças em unidades.

- Remoção de Stopwords

  Esse método consiste em remover palavras muito frequentes, tais como “a”, “de”, “o”, “da”, “que”, “e”, “do” entre outras, pois na maioria das vezes não são informações relevantes para a construção do modelo. Além disso, foram removidos todos os acentos.

- ❏ Remoção de numerais

  Outra remoção necessária é dos numerais presentes no texto, pois não agregam informação relevante por não trazerem carga semântica. Também, foi removido símbolos que os acompanham, como “R$”, “$”, “US$”, “kg”, “km”, “milhões”, “bilhões” dentre outros.

- ❏ Lematização A lematização reduz a palavra ao seu lema, que é a forma no masculino e singular. No caso de verbos, o lema é o infinitivo. Por exemplo, as palavras “gato”, “gata”, “gatos” e “gatas” são todas formas do mesmo lema: “gato”. Igualmente, as palavras “tiver”, “tenho”, “tinha”, “tem” são formas do mesmo lema “ter”.

Na sequência, foi executado o algoritmo word2vec da biblioteca Gensim da linguagem de programação do Python. Nos bastidores dessa biblioteca, o que está acontecendo é o treinamento de uma rede neural com uma única camada oculta, para prever a palavra atual com base no contexto. No entanto, não será usado a rede neural após o treinamento. O objetivo é aprender os pesos da camada oculta, os quais são essencialmente os vetores de palavras que se está tentando aprender. O vetor aprendido resultante também é conhecido como *embeddings*.

Para cada notícia do corpus Fake.Br foi selecionado a palavra mais frequente, para torná-la a palavra alvo na execução do word2vec. Assim, os valores resultantes do algoritmo geraram os vetores numéricos para a próxima etapa do modelo de classificação.

Como exemplo teremos o seguinte texto:
“MP pede proibição de filmagens no MAM e acaba praticando atentado contra a humanidade.”

- ❏ Normalização
  “mp pede proibição de filmagens no mam e acaba praticando atentado contra a humanidade.”

- ❏ Remoção de stopwords
  “mp pede proibição filmagens mam acaba praticando atentado contra humanidade.”

- ❏ Lematização
  “mp pede proibi filmar mam acaba praticar atentado contra humanidade”

- ❏ Tokenização

“mp”,”pede”,” proibi”,” filmar”,”mam”,”acaba”,”praticar”,”atentado”,”contra”,”humanidade”

3.3.2 Validação Cruzada K-fold

Validação Cruzada (*Cross Validation*) é uma técnica muito utilizada para a avaliação de desempenho de modelos de AM. A Validação Cruzada k-fold consiste em particionar os dados em conjuntos, onde um conjunto é utilizado para treino e outro conjunto é utilizado para teste, ambos são escolhidos aleatoriamente. O "k" significa a quantidade de conjuntos (*fold*) a técnica irá subdividir a base de dados. A Figura 6 exemplifica e ilustra a subdivisão em 10-fold, o valor definido para o modelo proposto neste trabalho.

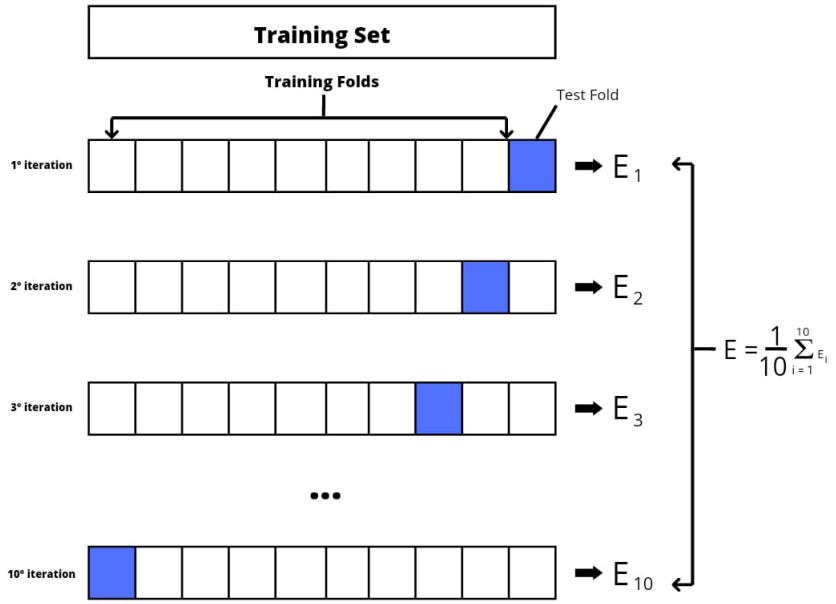

Figura 6 – Imagem ilustrando a técnica da Validação Cruzada 10-fold.

Fonte: http://karlrosaen.com/ml/learning-log/2016-06-20/

Como pode ser observado na Figura 6, na primeira iteração, o último subconjunto é usado como dados de teste, enquanto todos os outros subconjuntos são considerados como dados de treinamento. O modelo é treinado com os dados de treinamento e avaliado com o subconjunto de teste. Em cada nova iteração, um subconjunto diferente é escolhido como o conjunto de dados de teste e todo os subconjuntos restantes se tornam conjunto de dados de treinamento.

A utilização desta metodologia tem altas chances de detectar se o seu modelo está sobreajustado aos seus dados de treinamento, ou seja, sofrendo overfitting. Caso contrário, a média de resultados obtidos com o conjunto de teste define o quão bom o algoritmo de

AM está aprendendo o modelo.

3.4 Árvore de Decisão

A árvore de decisão implementada no modelo de classificação em estudo foi implementada pela biblioteca do Python scikit-learn (PEDREGOSA et al., 2011). A função escolhida para medir a qualidade das subdivisões da árvore foi *entropy*, com o número mínimo de amostras necessárias igual a cinco para dividir um nó interno e o controle da aleatoriedade do estimador igual a zero.

3.5 Resultados

3.5.1 Medidas de Avaliação

Após desenvolver a etapa de aprendizagem do classificador, é importante verificar se ele apresenta um bom desempenho executando o conjunto de dados de testes no modelo computacional treinado. Assim, para analisar o desempenho gerou a matriz de confusão, para que, consequentemente, calculasse as medidas de avaliação quantitativas de desempenho. São elas:

- *Acurácia*: é a razão do total de acertos que a árvore de decisão alcançou sobre o número total de notícias.
- *Recall*: é a razão entre o total de fake news classificadas corretamente sobre a soma da quantidade de fake news classificadas corretamente com a quantidade de fake news que foram consideradas verdadeiras;
- *Precisão*: é a razão entre o total de fake news classificadas corretamente sobre a soma da quantidade de fake news classificadas corretamente com a quantidade de notícias verdadeiras que foram consideradas fake news;
- *F1-score*: média harmônica entre precisão e o *recall*.

3.5.2 Análise dos Resultados

Os experimentos foram formados por 500 notícias do corpus Fake.Br escolhidos aleatoriamente, porém divididos em 50% para notícias do tipo fake news e 50% para notícias verdadeiras.

A Figura 7 mostra os valores para a medida acurácia resultante da execução do modelo computacional para classificação de notícias digitais. A média da acurácia para o conjunto de treinamento foi 98.75%, enquanto que a média da acurácia para o conjunto de teste

foi de 55.59%. Nota-se que apenas um subconjunto da validação cruzada teve acurácia inferior a 50% de acertos e três outros subconjuntos obtiveram acima de 60% de acertos. Essas diferenças justificam-se pelas diferentes notícias utilizadas para o treinamento e para o teste da árvore de decisão.

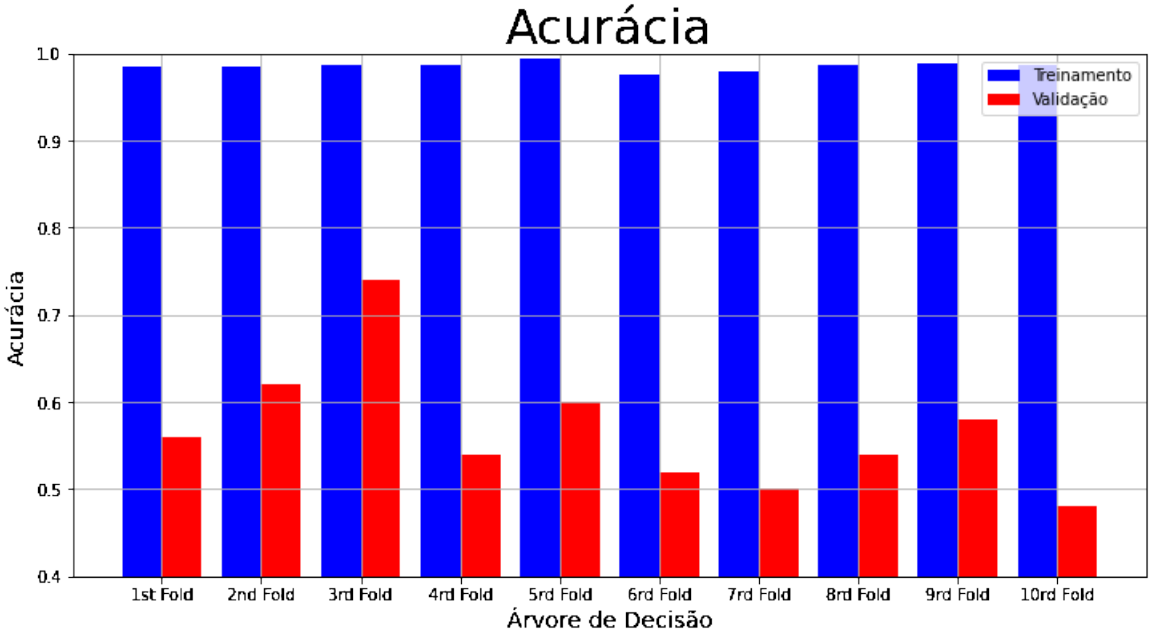

Figura 7 – Valores gerados pelo modelo computacional para a medida acurácia.

A Figura 8 mostra os valores da medida *recall* resultante da execução do modelo computacional. A média do *recall* para o conjunto de treinamento foi 98.49%, já para o conjunto de validação foi 55,37%. O *fold* de número 3, cujo subconjunto teve a maior acurácia, também se destaca pelo maior valor de *recall*, acima de 70%. Ou seja, o modelo computacional para esse subconjunto classificou corretamente uma maior quantidade de fake news.

A Figura 8 mostra os valores da medida *recall* resultante da execução do modelo computacional. A média do *recall* para o conjunto de treinamento foi 98.49%, já para o conjunto de validação foi 55,37%. O *fold* de número 3, cujo subconjunto teve a maior acurácia, também se destaca pelo maior valor de *recall*, acima de 70%. Ou seja, o modelo computacional para esse subconjunto classificou corretamente uma maior quantidade de fake news.

No entanto, dois subconjuntos presentes na Figura 8 tiveram o valor do *recall* para o conjunto de validação inferior à 40%. Em outras palavras, o modelo computacional executado para esses dois subconjuntos obtiveram uma significativa quantidade de classificação de fake news como sendo notícias verdadeiras.

A Figura 9 mostra os valores para a medida precisão resultante da execução do modelo computacional da classificação de notícias digitais. A média da precisão para o conjunto

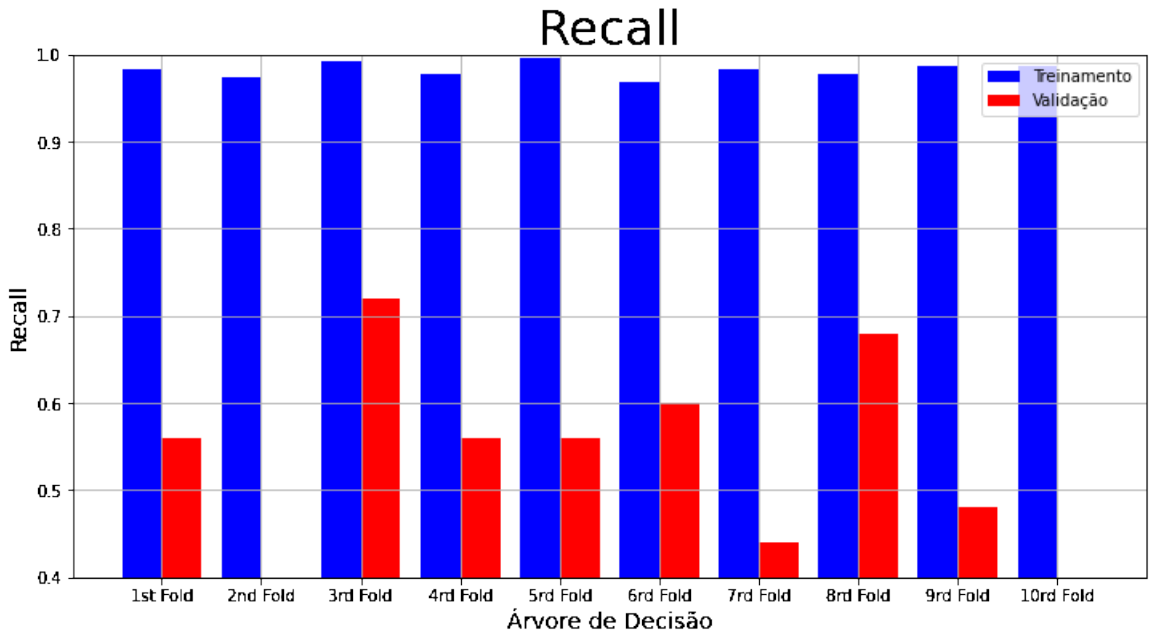

Figura 8 – Valores gerados pelo modelo computacional para a medida recall.

de treinamento foi 99.02%, enquanto que a média da precisão para o conjunto de teste foi de 56.83%.

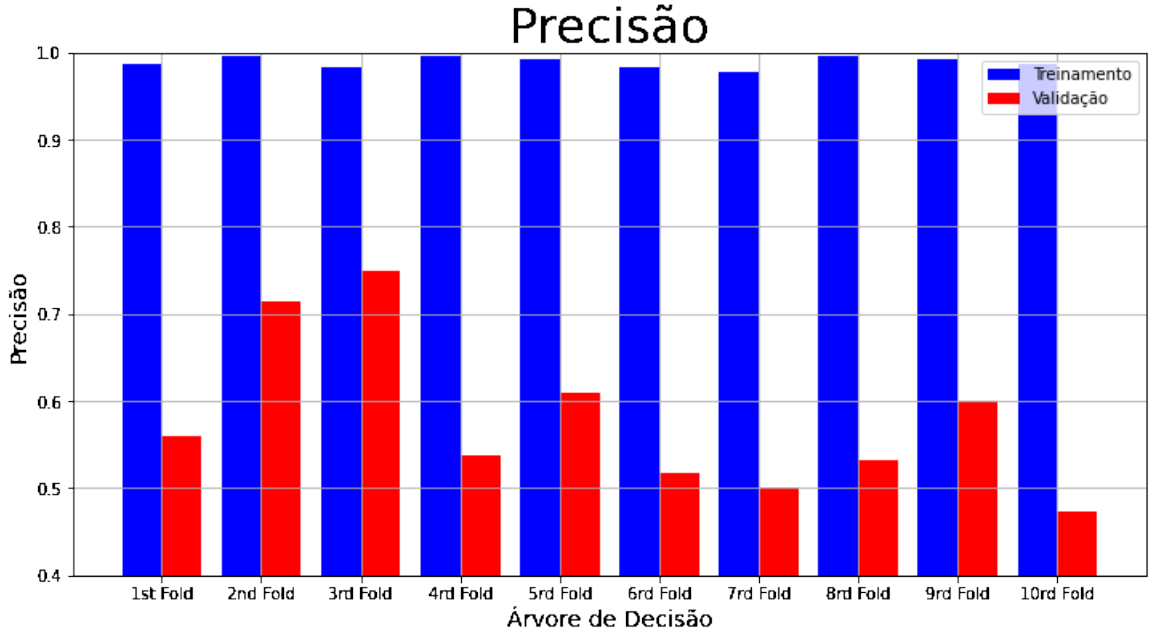

Figura 9 – Valores gerados pelo modelo computacional para a medida precisão.

Ademais, a Figura 9 mostra que apenas um subconjunto obteve o valor da precisão para o conjuto de validação inferior a 50%, o que significa que para todos os outros subconjuntos o modelo computacional obteve maiores acertos em classificar notícias verdadeiras.

A Figura 10 mostra os valores para a medida f1 score resultante da execução do modelo

computacional. A média do f1 score para o conjunto de treinamento foi 98.75%, e para o conjunto de validação foi 52,44%.

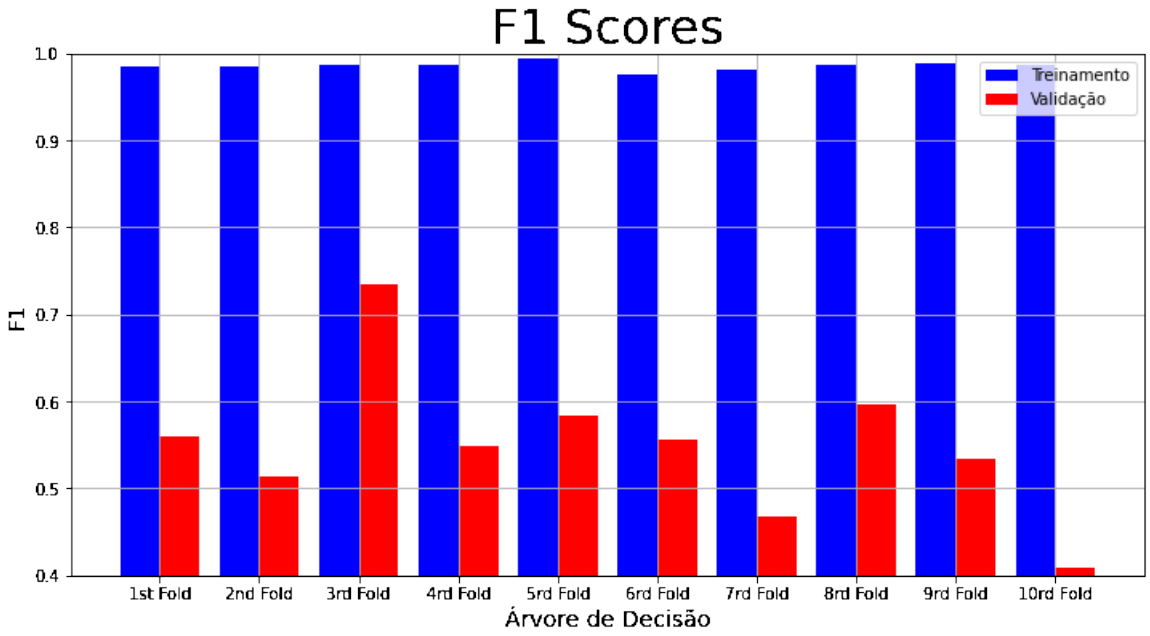

Figura 10 – Valores gerados pelo modelo computacional para a medida f1 score.

Capítulo 4

Considerações Finais

Este trabalho foi desenvolvido com o objetivo de construir um modelo computacional destinado à classificação de notícias falsas para o português do Brasil usando técnicas de PLN juntamente com a árvore de decisão como algoritmo de AM. Após o treinamento e a validação do classificador proposto pela metodologia da validação cruzada k-fold, os resultados foram apresentados no capítulo anterior. Conclui-se que os resultados gerados mostraram-se satisfatórios, devido a quantidade de notícias utilizadas no treinamento e pelo uso de apenas uma palavra alvo na execução do word2vec.

4.1 Principais Contribuições e Trabalhos Futuros

As metodologias empregadas no PLN devem ser executadas imediatamente após a coleta do corpus e promove uma formatação e representação da massa textual. Elas são bastante onerosas, pois os algoritmos consomem boa parte do tempo do processo de extração de conhecimento. Entretanto, o sucesso do classificador depende signficativamente das etapas do PLN. Assim, sugestiona que o modelo computacional proposto neste estudo poderia ser aperfeiçoado com o uso de diferentes técnicas de normalizações, lematização e, principalmente, de algoritmos de transformação de texto em informação numérica.

As Word Embeddings apresentam limitações. A principal delas é a confluência de significados. Isto é, uma vez que uma única representação é gerada para a mesma forma superficial de uma palavra, os diversos significados dessa forma superficial são "misturados" na mesma embedding perdendo-se, assim, informação linguística valiosa. Como proposta de trabalho futuro, uso de vetores dinâmicos, como já foi sugerido em bibliografias recentes.

Em suma, para trabalho futuro, expandiria os experimentos do classificador proposto para mais amostras do corpus Fake.Br, além de selecionar mais quatro palavras alvos em cada notícia durante a execução do algoritmo word2vec. Além disso, acrescentaria as redes neurais deep learning na fase final do classificador, em substituição às árvores de decisão.

Referências

GUARISE, L. **Detecção de notícias falsas usando técnicas de deep learning**. 51 p. Monografia (Graduação) — Instituto de Ciências Matemáticas e de Computação – ICMC-USP, São Carlos - SP, 2019. Citado na página 16.

HARTMANN, N. S. et al. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. In: **Anais do XI Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana**. Porto Alegre, RS, Brasil: SBC, 2017. p. 122–131. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/stil/article/view/4008>. Citado na página 16.

MERELES C.; MORAES, I. **Notícias falsas e pós-verdade: o mundo das fake news e da (des)informação.** 2017. Disponível em: <https://www.politize.com.br/noticias-falsas-pos-verdade/>. Citado na página 12.

MIKOLOV, T. et al. Efficient estimation of word representations in vector space. **Proceedings of Workshop at ICLR**, v. 2013, 01 2013. Citado na página 13.

MONTEIRO, R. et al. **Contributions to the Study of Fake News in Portuguese: New Corpus and Automatic Detection Results: 13th International Conference, PROPOR 2018, Canela, Brazil, September 24–26, 2018, Proceedings**. [S.l.: s.n.], 2018. 324-334 p. ISBN 978-3-319-99721-6. Citado 3 vezes nas páginas 11, 16 e 19.

OLIVEIRA, N. et al. **Processamento de Linguagem Natural para Identificação de Notícias Falsas em Redes Sociais: Ferramentas, Tendências e Desafios**. [S.l.: s.n.], 2020. 51-100 p. ISBN 9786587003856. Citado 2 vezes nas páginas 12 e 13.

PARALELO, R. B. **O que é Fake News? Entenda o significado desse conceito e como ele vem sendo usado**. 2022. Disponível em: <https://www.brasilparalelo.com.br/artigos/fake-news>. Citado na página 10.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011. Citado 2 vezes nas páginas 20 e 23.

PYTHON. 2022. Disponível em: <https://www.python.org/>. Citado na página 20.

RUSSELL, S.; NORVIG, P. **Inteligência Artificial: Referência Completa para os Cursos de Computação**. [S.l.: s.n.], 2004. 1021 p. ISBN 9788535211771. Citado na página 15.

SHARIATMADARI, D. **Could language be the key to detecting fake news?** 2019. Disponível em: <https://www.theguardian.com/commentisfree/2019/sep/02/language-fake-news-linguistic-research>. Citado na página 10.

SHARMA, K. et al. Combating fake news: A survey on identification and mitigation techniques. **ACM Trans. Intell. Syst. Technol.**, Association for Computing Machinery, New York, NY, USA, v. 10, n. 3, apr 2019. ISSN 2157-6904. Disponível em: <https://doi.org/10.1145/3305260>. Citado 2 vezes nas páginas 12 e 13.

SILVA, R. M. et al. Towards automatically filtering fake news in portuguese. **Expert Systems with Applications**, v. 146, p. 113199, 2020. ISSN 0957-4174. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0957417420300257>. Citado na página 16.