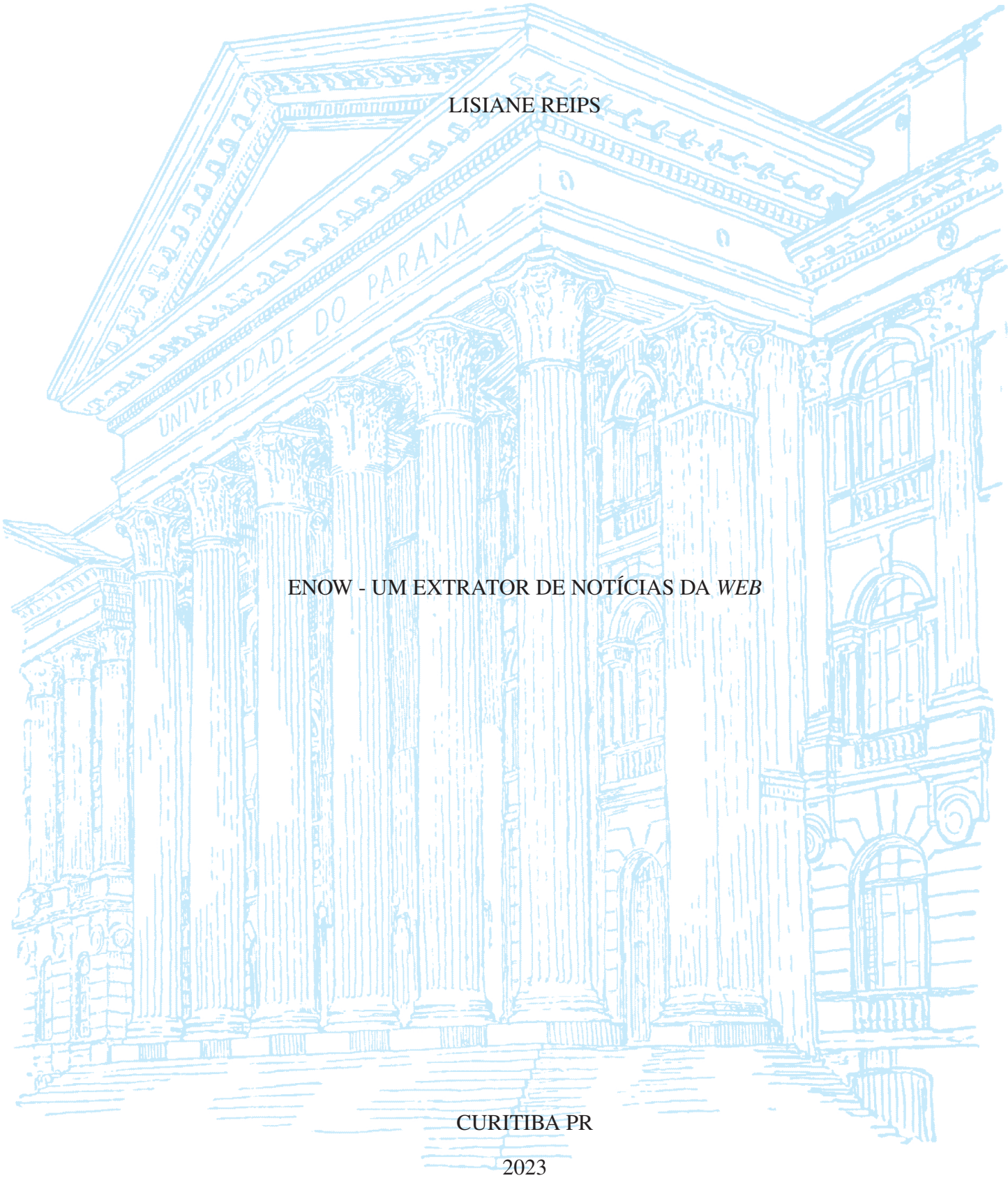
UNIVERSIDADE FEDERAL DO PARANÁ

LISIANE REIPS



ENOW - UM EXTRATOR DE NOTÍCIAS DA *WEB*

CURITIBA PR

2023

LISIANE REIPS

ENOW - UM EXTRATOR DE NOTÍCIAS DA *WEB*

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciência da Computação no PPGINF (Programa de Pós-Graduação em Informática), Setor de Ciências Exatas, da UFPR (Universidade Federal do Paraná).

Área de concentração: *Ciência da Computação*.

Orientador: Carmem Satie Hara.

CURITIBA PR

2023

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)
UNIVERSIDADE FEDERAL DO PARANÁ
SISTEMA DE BIBLIOTECAS – BIBLIOTECA DE CIÊNCIA E TECNOLOGIA

Reips, Lisiane
ENoW - um extrator de notícias da Web / Lisiane Reips. – Curitiba, 2023.
1 recurso on-line : PDF.

Dissertação (Mestrado) - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática.

Orientador: Carmem Satie Hara

1. Jornalismo. 2. Serviços de notícias. 3. Jornais eletrônicos. 4. World Wide Web (Sistema de recuperação da informação) - Acesso por assunto. 5. Extrator de Notícias da Web. I. Universidade Federal do Paraná. II. Programa de Pós-Graduação em Informática. III. Hara, Carmem Satie. IV. Título.

Bibliotecário: Elias Barbosa da Silva CRB-9/1894

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da dissertação de Mestrado de **LISIANE REIPS** intitulada: **ENOW - Um Extrator de Notícias da Web**, sob orientação da Profa. Dra. CARMEM SATIE HARA, que após terem inquirido a aluna e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestra está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 18 de Outubro de 2023.

Assinatura Eletrônica
19/10/2023 08:35:36.0
CARMEM SATIE HARA
Presidente da Banca Examinadora

Assinatura Eletrônica
23/10/2023 07:24:57.0
MARTIN ALEJANDRO MUSICANTE
Avaliador Externo (UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE)

Assinatura Eletrônica
19/10/2023 10:45:52.0
AURORA TRINIDAD RAMIREZ POZO
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

A tod@s que sempre me apoiam e torcem por mim.

AGRADECIMENTOS

Gostaria de agradecer imensamente a todos que contribuíram para a realização deste trabalho de mestrado. Este é um momento especial que não teria sido possível sem o apoio e a inspiração de pessoas importantes ao meu redor.

Agradeço a minha família, que sempre foi meu alicerce e fonte de apoio. Cada conquista aqui é um reflexo do apoio que recebi ao longo desta jornada.

Agradeço a minha orientadora, Carmem Satie Hara, pela orientação valiosa, pelos conselhos, pela inspiração e paciência de sempre.

Também agradeço aos amigos e colegas que estiveram ao meu lado durante este percurso. Principalmente ao meu amigo Gian Paulo Vieira, que sempre me auxiliou no desenvolvimento e implementação do ENoW.

Agradeço a comunidade acadêmica que proporcionou um ambiente propício ao aprendizado.

Agradeço a todos que, de alguma forma, contribuíram para este trabalho, seja por meio de discussões construtivas, suporte técnico ou incentivo moral. Cada contribuição foi valiosa e apreciada.

RESUMO

A transição dos meios de comunicação tradicionais para o ambiente digital abrange diversas áreas. Dentre estes meios, destacam-se os jornais, que têm disponibilizado seu conteúdo *online*, permitindo o acesso a uma diversidade de dados na *Web*. Para explorar estes dados, eles precisam ser extraídos, armazenados, organizados e filtrados de acordo com os interesses da aplicação. Entretanto, os sistemas que viabilizam esses processos nem sempre dão suporte a todas as funcionalidades. Alguns focam somente em extração e armazenamento, enquanto outros englobam extração, processamento e transformação. Há ainda aqueles que abrangem somente a transformação e filtragem. Nesse contexto, surge o **Extrator de Notícias da *Web*** [1] (**ENoW**), um sistema de coleta de dados de jornais *online* que pré-processa os dados coletados, com o intuito de filtrar apenas as notícias de interesse do usuário. O *ENoW* aceita como entrada *strings* de busca, realiza a coleta de notícias relacionadas àquela *string* e armazena as notícias coletadas em uma base de dados relacional. O sistema mantém a proveniência dos dados, bem como um log com histórico de extrações. Ele foi implementado na linguagem de programação *Python*, utilizando técnicas de *Web Scraping*. A avaliação do ENoW foi realizada por meio de uma análise experimental. O processo envolve a coleta de dados de notícias de um conjunto de URLs, seguido do pré-processamento destes dados. Além disso, são empregados algoritmos de aprendizado de máquina e cálculos de semelhança de textos para a filtragem das notícias. Um estudo de caso sobre notícias referentes a caravelas-portuguesas (cnidário *Physalia physalis*) mostra o desempenho do processo de filtragem.

Palavras-chave: Extração de Notícias. Armazenamento Relacional. Busca por palavra-chave.

[1] Disponível em https://github.com/LReips/ENoW

ABSTRACT

The transition from traditional media to the digital environment covers many areas. Newspapers around the world have made their content available online, allowing access to a variety of data on the Web. To exploit this data, they need to be extracted, stored, organized and filtered according to the interests of the application. However, the systems that enable these processes don't always support all the functionalities. Some focus only on extraction and storage, while others encompass extraction, processing and transformation. There are also those that only cover transformation and filtering. In this context, we have developed the **Extractor de Notícias da *Web*** (**ENoW**). It is a system for collecting data from *online* newspapers and for processing the collected data in order to filter out only the news of interest to the user. *ENoW* accepts a set of *strings* as input, collects news related to that *string*, and stores the collected news in a relational database. The system maintains the provenance of the data, as well as a log with the history of extractions. It was implemented in *Python* using *Web Scraping* techniques. We have conducted an experimental analysis involving the collection of news data from a set of URLs. The system pre-processes this data and uses machine learning algorithms and text similarity calculations to filter the news. A case study of news involving Portuguese man-of-war (cnidarian *Physalia physalis*) shows the effectiveness of the filtering process.

Keywords: News Extraction. Relational Storage. Keyword search.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ACRÔNIMOS

AM	Aprendizado de Máquina
API	*Application Programming Interface*
BERT	*Bidirectional Encoder Representations from Transformers*
BoW	*Bag of Words*
CSV	*Comma-Separated Values*
ENoW	Extrator de dados de Notícias *Web*
FN	Falsos Negativos
FP	Falsos Positivos
GloVe	Global Vectors for Word Representation
HTML	*HyperText Markup Language*
IA	Inteligência Artificial
IDF	*Inverse Document Frequency*
JSON	*JavaScript Object Notation*
KNN	*K-Nearest Neighbor*
LSTM	*Long Short Term Memory*
MLP	*Multi-Layer Perceptron*
NB	*Naive Bayes*
PLN	Processamento de Linguagem Natural
PPGINF	Programa de Pós-Graduação em Informática
RBC	*Rule-Based Classifier*
REN	Reconhecimento de Entidades Nomeadas
RF	*Random Forest*
RI	Recuperação de Informações
RL	Regressão Logística
RNN	Rede Neural Recorrente
RSS	*Rich Site Summary*
SBERT	*Sentence*-BERT
TF	*Term Frequency*
TF-IDF	*Term Frequency-Inverse Document Frequency*
UFPR	Universidade Federal do Paraná
URL	*Uniform Resource Locator*
VN	Verdadeiros Negativos
VP	Verdadeiros Positivos
XML	*Extensible Markup Language*

SUMÁRIO

1 INTRODUÇÃO

Atualmente, dados disseminados por meio das plataformas *online* são amplamente compartilhados e utilizados para uma variedade de propósitos (Ferrara et al., 2014). Tarefas que variam desde previsões de mercado até tradução de linguagem, diagnósticos médicos e expressões artísticas (Mitchell, 2018) recorrem a esses dados, requerendo processos de extração, depuração e armazenamento (Johnson, 2014). Diante dessa realidade, a seleção de fontes confiáveis emerge como um passo relevante na extração de dados, principalmente levando em conta a natureza diversificada e abrangente das fontes de informação. Ainda que as mídias e redes sociais tenham se estabelecido como as principais fontes de informação para os usuários, é notável que tais plataformas também se converteram em veículos para a propagação de notícias falsas (Moser et al., 2022). Contudo, as notícias de jornais permanecem como uma fonte confiável de informações (Jr et al., 2018), sendo amplamente empregadas em diversas aplicações, a exemplo da previsão de demandas turísticas e da identificação de áreas afetadas por desastres naturais, como deslizamentos de terra (Park et al., 2021; Franceschini et al., 2022). A extração desses dados pode ser executada quer por métodos manuais ou por abordagens automatizadas.

A extração manual de dados *online* se caracteriza pelo processo de coleta de informações provenientes de várias fontes na *Web*, prescindindo de ferramentas ou *software* automatizados. Apesar da morosidade e complexidade associadas à extração manual, esta estratégia encontra aplicação relevante em campos diversos, a exemplo de estudos sobre a prevalência de hipertensão no Paquistão, que demandaram a coleta manual de dados de prontuários (Riaz et al., 2021). Apesar das dificuldades e dos desafios inerentes, a extração manual mantém seu lugar em pesquisas e âmbitos acadêmicos (Vargas-Solar et al., 2021; do Nascimento et al., 2022). Em contrapartida, a automação da extração de dados *online* emerge como uma aliada na redução do esforço humano empregado na coleta e análise de dados, encontrando no processo de *Web Scraping* uma abordagem amplamente utilizada (Mitchell, 2018).

A técnica de *Web Scraping* possibilita a extração de dados específicos de páginas *online*. No entanto, para viabilizar a exploração desses dados, como em contextos de classificação utilizando modelos de Aprendizado de Máquina (AM), é necessária a prévia adequação desses dados brutos. O conjunto de etapas realizadas sobre os dados brutos antes de sua aplicação em tarefas de análise ou modelagem é conhecido como pré-processamento, desempenhando papel fundamental no Processamento de Linguagem Natural (PLN). Por meio do PLN, é almejado que os computadores compreendam e interpretem textos à semelhança dos seres humanos. Nesse cenário, surge a relevância de um sistema que englobe a sequência de extração de dados confiáveis e subsequente pré-processamento, elevando-se ainda mais a utilidade desse sistema mediante a capacidade de apresentar dados categorizados conforme as necessidades do usuário.

Todavia, os sistemas existentes frequentemente exibem limitações em alguma de suas funcionalidades. Alguns se restringem à extração sem pré-processamento, enquanto outros se concentram unicamente na etapa de pré-processamento. Além disso, mesmo nos casos em que tais sistemas oferecem uma variedade de funcionalidades, muitas vezes elas não estão disponíveis de maneira integralmente gratuita, o que pode dificultar o acesso para usuários que não dispõem dos recursos necessários para pagar. Adicionalmente, a limitação no acesso a metadados do sistema acrescenta complexidade à gestão das informações.

No intuito de superar essas lacunas, a presente dissertação propõe o desenvolvimento do **Extrator de Notícias da *Web*** (ENoW). O ENoW é desenvolvido como um sistema extrator capaz de, a partir de *strings* de interesse definidas pelo usuário, extrair notícias de um conjunto

de jornais *online*. Adicionalmente, o ENoW realiza o pré-processamento dos dados extraídos e emprega técnicas de similaridade de textos e de Aprendizado de Máquina (AM) para filtrar as notícias de interesse. O ENoW está disponível na plataforma GitHub.

A fim de avaliar a viabilidade dessa proposta, foi conduzido um experimento iniciado com a coleta de dados provenientes de jornais *online* brasileiros, os quais foram obtidos mediante uma *string* de busca especificada pelo usuário. Na sequência, o usuário identifica uma ou mais notícias de seu interesse, que passam pela etapa de pré-processamento, seguida pela fase de filtragem, para categorizar as notícias com base em características relevantes. A intervenção humana (*human-in-the-loop*) visa garantir a adequação das categorizações, através da sua revisão e refinamento. A avaliação das abordagens empregadas se ampara em métricas de desempenho. Dessa maneira, a dissertação propõe uma solução abrangente e integrada para a extração, pré-processamento e filtragem de dados de notícias *online*. A avaliação empírica da proposta busca não apenas validar sua eficácia, mas também oferecer *insights* relevantes para a melhoria contínua dessa abordagem.

1.1 MOTIVAÇÃO

A principal motivação para o desenvolvimento do ENoW está na demanda por um processo de extração de informações de notícias *online* que seja caracterizado pela entrega de dados depurados e pré-processados, aliado à filtragem desses dados, sendo um sistema inteiramente gratuito. Com essa perspectiva, a intenção é dotar os usuários com uma ferramenta que, além de coletar informações úteis para diversas áreas, prepara essas informações para análise e interpretação. A importância dessa motivação se manifesta na capacidade do ENoW de contribuir para o processamento de informações, aprimorando a experiência do usuário. Assim, o objetivo central do ENoW é dar suporte a procedimentos de classificação de dados extraídos de notícias *online*. Para atingir esse fim, o sistema se propõe a realizar tanto o processo de extração quanto o pré-processamento desses dados. Além do objetivo principal, desdobram-se objetivos secundários que abrangem a fase inicial do fluxo de trabalho, enfatizando a obtenção de dados de notícias, bem como sua pronta utilização para fins analíticos. Para atender a esses objetivos, a seguinte hipótese foi definida.

Hipótese: a hipótese contida a este estudo sugere que é possível extrair dados de notícias *online*, correspondentes ao tema selecionado pelo usuário, realizar um pré-processamento e executar uma classificação desses dados, em um único sistema, com disponibilização gratuita de uso.

1.2 CONTRIBUIÇÕES

As contribuições deste estudo aplicam-se diretamente na esfera da extração automatizada de dados, conferindo-lhes a capacidade de serem preparados para análises de classificação. As principais contribuições abrangem:

• A idealização e o desenvolvimento de uma ferramenta de extração de dados de notícias *online*, com flexibilidade para aplicação em diversas situações, juntamente com o armazenamento de metadados e informações em um banco de dados;

• O desenvolvimento de um processo que envolva o usuário ativamente na filtragem das notícias relevantes, interagindo de forma iterativa e influenciando no resultado;

• A aplicação de modelos de AM para realizar a classificação dos dados, empregando dados já rotulados para orientar essa classificação;

• A condução de análises experimentais com o intuito de validar e verificar o desempenho da ferramenta desenvolvida, assegurando sua aplicabilidade em outros cenários.

1.3 ORGANIZAÇÃO DO DOCUMENTO

O restante do documento está organizado da seguinte forma. O Capítulo 2 apresenta fundamentos teóricos sobre curadoria dos dados, envolvendo fontes de informação, extração, pré-processamento de dados, vetorização, AM e o processo *human-in-the-loop*. O Capítulo 3 apresenta e compara as abordagens relacionadas ao tema dessa dissertação, presentes em outros trabalhos. O Capítulo 4 apresenta o ENoW, bem como a sua arquitetura, o seu modelo de dados e o seu funcionamento. O Capítulo 5 descreve a implementação do ENoW. O Capítulo 6 apresenta os experimentos realizados. O Capítulo 7 apresenta as considerações finais dessa dissertação.

2 FUNDAMENTAÇÃO TEÓRICA

Este Capítulo apresenta os principais fundamentos sobre os processos de extração, transformação e carregamento (ETL - Extraction, Transformation, Load). Na Seção 2.1, são descritas as fontes de dados escolhidas para este trabalho. Na Seção 2.2 são apresentados os conceitos sobre a extração dos dados dessas fontes. Os conceitos sobre o pré-processamento dos dados são vistos na Seção 2.3. A Seção 2.4 aborda as definições de vetorização. Na Seção 2.5 são apresentados fundamentos sobre Aprendizado de Máquina (AM). A Seção 2.6 aborda conceitos de *human-in-the-loop*. Na Seção 2.7 são apresentadas as considerações.

2.1 FONTES DE INFORMAÇÃO

A esfera jornalística tem experimentado notáveis transformações (Moura et al., 2023), resultando na produção crescente de conteúdos digitais. Nesse cenário, as informações jornalísticas alcançam diretamente o usuário, sem a necessidade de exploração ativa em um portal de notícias *online*. Muitas vezes, as notícias são recebidas de forma passiva pelo usuário, seja por meio de compartilhamentos efetuados por outros usuários ou ainda mediante fontes previamente seguidas por estes (Bentley et al., 2019).

Outro fator considerado é a celeridade com que os jornais *online* apresentam informações atualizadas. Isto é perceptível enquanto o usuário navega pelo *site* e este atualiza-se automaticamente, incorporando novas notícias em tempo real. Além disso, os portais de notícias *online* proporcionam uma ampla variedade de assuntos, viabilizando a obtenção de perspectivas diversas sobre algum tema específico.

2.2 EXTRAÇÃO DE DADOS DA WEB

A extração de dados provenientes da *Internet* pode ser realizada de forma manual ou por meio de *software* específicos para essa finalidade. A extração manual de dados da *Web* refere-se ao processo de coletar informações e conteúdos de *sites* por meio de interações humanas diretas (Lawson, 2015). Nesse método, os indivíduos visitam os *sites* desejados (Lawson, 2015), analisam artigos, examinam imagens e, então, registram manualmente as informações relevantes. Diante deste contexto, observa-se que o processo de extração manual é uma abordagem morosa e que demanda tempo. No entanto, essa abordagem apresenta algumas vantagens, tais como o controle completo por parte dos coletores de dados sobre as informações que estão sendo coletadas. Além disso, possibilita a filtragem e seleção de dados considerados mais pertinentes, resultando em maior precisão na seleção das informações coletadas. Adicionalmente, essa abordagem proporciona um entendimento imediato da contextualização dos textos a serem coletados. Contudo, juntamente com as vantagens, tornam-se visíveis também as desvantagens: a extração manual requer investimento de tempo e recursos humanos, o que pode ser dispendioso, particularmente para projetos que envolvam coleta de volumes significativos de dados (Upadhyay et al., 2017). Além disso, a coleta de grandes quantidades de dados pode ser inviável ou impraticável. Outra desvantagem é a repetitividade inerente ao processo manual, o que pode levar à fadiga e a erros humanos, incluindo a omissão de informações relevantes ou interpretação equivocada dos dados.

Com o intuito de minimizar esses desafios inerentes ao processo manual, mecanismos automatizados de extração de dados se apresentam como alternativas auxiliares disponíveis na

Web. A extração automatizada de dados através do *Web Scraping* é uma alternativa para a coleta de informações, notadamente no contexto de notícias (Barbosa e Cavalcanti, 2020). O *Web Scraping* envolve a utilização de *scripts* e ferramentas para extrair dados de *sites*, eliminando a necessidade de intervenção manual. Esse método oferece vantagens como celeridade em relação à extração manual e utilização eficiente de recursos por meio da automação. Além disso, os processos automatizados com *Web Scraping* são ideais para a coleta de grandes volumes de dados textuais e imagens (Lawson, 2015).

As Interfaces de Programação de Aplicativos (APIs) também desempenham um papel importante na reutilização de *software* (Rauf et al., 2019), sendo capazes de realizar extrações de dados. No entanto, nem sempre uma API se adequa ao propósito desejado. Adicionalmente, existem fóruns[1] que compartilham códigos utilizando *Web Scraping*, o que auxilia e simplifica o processo de desenvolvimento de extratores de dados. Após a etapa de extração, frequentemente é necessário realizar o pré-processamento dos dados (Kang et al., 2020).

2.3 PRÉ-PROCESSAMENTO DOS DADOS

Após a obtenção dos dados, o pré-processamento, frequentemente, constitui a segunda etapa em um projeto de Processamento de Linguagem Natural (PLN) (Santos et al., 2022). O PLN é um domínio da Inteligência Artificial (IA) que se concentra na interação entre computadores e a linguagem humana (Ribeiro et al., 2020). O escopo do PLN consiste em capacitar computadores para compreender, interpretar e gerar linguagem de modo análogo aos seres humanos. Isso engloba diversas tarefas, desde a compreensão de palavras e frases até a análise avançada de significados, contexto e sentimentos expressos na linguagem. A etapa de pré-processamento assume um papel essencial no PLN, aprimorando a qualidade dos dados. O pré-processamento visa tornar os dados mais adequados para utilização em várias tarefas, como:

• Análise de Sentimento: os dados pré-processados podem ser usados para determinar o sentimento expresso em um texto, identificando se o conteúdo é positivo, negativo ou neutro. Isso serve para avaliar opiniões de clientes (Tereso, 2019) e análise de mídias sociais (Xavier, 2019), por exemplo.

• Classificação de Texto: com base no conteúdo do texto, ele pode ser categorizado em diferentes classes ou categorias. Isso é comum em tarefas como classificação de artigos de notícias em tópicos específicos (da S. Lima, 2023) e categorização de e-mails (Silva, 2021).

• Extração de Informações: os dados pré-processados podem ser usados para extrair informações específicas de um texto, como identificar nomes de pessoas, datas, locais e outros tipos de Reconhecimento de Entidades Nomeadas (REN).

• Sumarização Automática: com base no texto pré-processado, é possível criar resumos automáticos que capturam as informações essenciais do texto original. Isso é útil para processar grandes volumes de informações rapidamente (Rodrigues, 2020).

• Tradução Automática: os dados pré-processados podem ser usados em sistemas de tradução automática para converter texto de um idioma para outro (Moro, 2019).

• Pergunta e Resposta Automatizadas: usando os dados pré-processados, é possível criar sistemas de perguntas e respostas que respondam a perguntas feitas pelos usuários com base no conteúdo do texto (Zucchi e dos Reis, 2021).

• Mineração de Opiniões: consiste na identificação de opiniões e avaliações expressas no texto (dos Santos, 2021).

Mas para utilizar os dados nas tarefas supracitadas, algumas etapas do pré-processamento, precisam ser realizadas (Malley et al., 2019):

[1] https://forum.scriptbrasil.com.br/topic/202045-web-scraping/

• Limpeza de dados: o processo de limpeza é realizado para remover elementos indesejados, ruídos e informações irrelevantes ou redundantes dos dados textuais brutos, a fim de prepará-los para análises mais eficazes. A limpeza dos dados é uma etapa crítica do pré-processamento, pois dados "sujos" e não processados podem afetar negativamente a qualidade e os resultados das análises subsequentes. A limpeza envolve a remoção de caracteres especiais, pontuações e símbolos desnecessários, além de converter todo o texto para letras minúsculas ou maiúsculas, a fim de evitar duplicações de palavras.

Além disso, ela pode envolver os seguintes processos:

• *Tokenização*: o processo de *tokenização* é uma etapa fundamental no PLN, pois divide o texto em unidades menores que podem ser analisadas, processadas e compreendidas de maneira mais eficaz por algoritmos de PLN. Isso permite que uma ampla variedade de tarefas de análise textual seja realizada com precisão e eficiência.

• Remoção de *stop words*: as *stop words* são palavras comuns que geralmente não contribuem muito para a compreensão do texto e podem acabar prejudicando o processamento do texto. As palavras "a", "e", "o", "de" são alguns dos exemplos de *stop words*. A remoção delas reduz a dimensionalidade dos dados.

• *Stemming* e *Lemmatization*: esses processos reduzem as palavras às suas formas raiz (*lemmas*) para lidar com variações de palavras, como "correr", "correndo" e "corre" sendo transformadas em "correr";

2.4 VETORIZAÇÃO DOS DADOS

Esta Seção apresenta fundamentos sobre a vetorização dos dados. Ela está estruturada da seguinte forma. A Seção 2.4.1 apresenta conceitos sobre vetorização dos dados. A Seção 2.4.2 descreve conceitos sobre similaridade de texto.

2.4.1 Vetorização

O processo de vetorização converte o texto em uma representação numérica. Isso pode ser feito usando técnicas como *Bag of Words* (BoW) (Hardeniya et al., 2016) ou *Term Frequency-Inverse Document Frequency* (TF-IDF) (Jurafsky, 2000) para criar uma matriz numérica.

A técnica BoW representa o texto em uma coleção de palavras, que formam um vetor (ignorando a estrutura gramatical), juntamente com a contagem de quantas vezes cada palavra aparece no documento. Por exemplo: denomina-se como D um documento, que é um fragmento de texto. Também denomina-se como V, o conjunto de todas as palavras únicas encontradas no vocabulário do conjunto desses documentos. O vetor BoW para o documento D é uma representação numérica onde cada elemento *BoW(D, w)* corresponde à contagem de quantas vezes a palavra w (pertencente ao vocabulário V) aparece no documento D.

Assim, o BoW de uma palavra w no documento D é definido como:

$$\text{BoW}(D, w) = \text{contagem de ocorrências de } w \text{ em } D, \tag{2.1}$$

Considerando como exemplo um texto de uma notícia real, tem-se:

"Uma caravela-portuguesa chamou a atenção de banhistas em Praia Grande, no litoral de São Paulo"[2].

Para representar esse texto usando a técnica BoW, tem-se o vocabulário V considerado:

[2]https://g1.globo.com/sp/santos-regiao/noticia/2023/01/18/caravela-portuguesa-chama-atencao-de-banhistas-no-litoral-de-sp-e-especialista-explica-os-riscos.ghtml

V = {caravela-portuguesa, chamou, atenção, banhistas, praia, grande, litoral, são, paulo}.

A contagem de ocorrências de palavras no documento, usando a técnica BoW seria:
BoW(D, caravela-portuguesa) = 1
BoW(D, chamou) = 1
BoW(D, atenção) = 1
BoW(D, banhistas) = 1
BoW(D, praia) = 1
BoW(D, grande) = 1
BoW(D, litoral) = 1
BoW(D, são) = 1
BoW(D, paulo) = 1
O vetor BoW para o documento resulta em:

[1, 1, 1, 1, 1, 1, 1, 1].

A técnica TF-IDF também transforma cada documento em um vetor, mas ao invés de usar apenas a contagem de ocorrências, ele leva em conta a importância relativa das palavras dentro do documento e em toda a coleção de documentos. Isso é feito usando um cálculo que leva em consideração a frequência do termo (TF) no documento e a frequência inversa do documento (IDF) na coleção. Por exemplo: Seja D um documento, V o conjunto de todas as palavras no vocabulário, e N o número total de documentos na coleção. O valor TF-IDF para uma palavra w no documento D é dado pela fórmula (Aizawa, 2003):

$$\text{TF-IDF}(D, w) = \text{TF}(D, w) \times \text{IDF}(w), \tag{2.2}$$

em que

$$\text{TF}(D, w) = \frac{\text{Número de vezes que } w \text{ aparece em } D}{\text{Número total de palavras em } D}, \tag{2.3}$$

e

$$\text{IDF}(w) = \log\left(\frac{N}{\text{Número de documentos em que } w \text{ aparece}}\right). \tag{2.4}$$

Considera-se *d1* e *d2* como sentenças:

d1 = "Caravelas-portuguesas são bonitas e perigosas.", e

d2 = "Caravelas-portuguesas podem ser perigosas e causar queimaduras.",

Supondo que o número total de documentos N é 5000, e em 2 destes documentos há o termo caravelas-portuguesas, verifica-se então a frequência de cada termo (palavra) na sentença, conforme demonstrado na Tabela 2.1.

Termo	***d1***	***d2***
caravelas-portuguesas	1	1
ser	1	1
bonitas	1	0
perigosas	1	1
podem	0	1
causar	0	1
queimaduras	0	1

Tabela 2.1: Frequência de termos nas sentenças *d1* e *d2*.

Calculando o TF-IDF para o termo caravelas-portuguesas, tem-se:

$$\text{TF}(D, caravela - portuguesa) = \left(\frac{1}{5}\right) \approx 0.2, \tag{2.5}$$

e

$$\text{IDF}(D, caravela - portuguesa) = log\left(\frac{5000}{2}\right) \approx 3.39, \tag{2.6}$$

obtêm-se

$$\text{TF-IDF}(D, caravela - portuguesa) = 0.2 \times 3.39 \approx 0.67. \tag{2.7}$$

Assim, os valores de TF-IDF para cada termo podem ser visualizados na Tabela 2.2.

Termo	**TF-IDF(*d1*)**	**TF-IDF(*d2*)**
caravelas-portuguesas	0,67	0,54
ser	0,33	0,47
bonitas	0,60	0
perigosas	0,71	0,64
podem	0	0,63
causar	0	0,49
queimaduras	0	0,31

Tabela 2.2: TF-IDF de cada termo nas sentenças *d1* e *d2*.

Ressalta-se que cada cálculo leva em consideração a frequência do termo (TF) no documento e, quando possível, a frequência inversa do documento (IDF) na coleção de documentos. O valor TF-IDF pondera a importância relativa de cada palavra no documento e na coleção, levando em consideração tanto a frequência do termo quanto sua raridade na coleção. Assim, se um termo não está presente em uma sentença, sua contagem (TF) será zero, pois ele não aparece nessa sentença. Se um termo está presente em todas as sentenças, significa que é muito comum e não fornece informações distintivas sobre uma sentença específica. Assim, seu IDF será zero, indicando a falta de relevância. Quando ambos TF e IDF são 0, o TF-IDF para esse termo também será zero. Isso implica que o termo não contribui para a representação diferenciada das sentenças, pois é comum a todas as sentenças e não traz informações relevantes. Além do BoW e do TF-IDF, *Word2Vec* é outra técnica utilizada para vetorização (Sitikhu et al., 2019).

A conversão de sentenças em vetores, com o propósito de calcular a similaridade entre pares de sentenças, é conhecida como *embedding* (Lage e Cunha, 2022). Os vetores resultantes representam uma codificação numérica de palavras ou frases em um espaço vetorial, no qual a proximidade espacial reflete a semelhança entre essas palavras ou frases. Esses vetores são empregados para explorar a similaridade das palavras, por exemplo. A verificação da similaridade nos textos é relevante para avaliação de categorização de textos, Recuperação de Informações (RI), detecção de plágio e tradução de documentos (Wang e Kuo, 2020).

2.4.2 Similaridade de Texto

Existem diversas técnicas para calcular a distância entre dois vetores de *embeddings*. Uma delas é a distância de cosseno A distância de cosseno é calculada mediante a fórmula:

$$\text{distância de cosseno} = 1 - \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \cdot \|\vec{B}\|}, \tag{2.8}$$

em que $\vec{A}$ e $\vec{B}$ são os vetores a serem comparados, e $\cdot$ denota o produto escalar desses vetores. Para calcular o produto escalar, somam-se os produtos dos componentes dos dois vetores. Esse cálculo verifica o quão alinhados estão os vetores na mesma direção. $\|\vec{A}\| \cdot \|\vec{B}\|$ representam as normas (magnitudes) dos vetores $\vec{A}$ e $\vec{B}$. A norma de um vetor é a medida de seu comprimento no espaço vetorial. Para a distância de cosseno, as normas são usadas para normalizar os vetores. Para ilustrar o cálculo de distância, aqui é considerado o TF-IDF das palavras como métrica de entrada. O cálculo de similaridade entre os vetores TF-IDF das sentenças *d1* e *d2* apresentadas anteriormente, utilizando a fórmula do cosseno:

$$\text{similaridade de cosseno} = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \cdot \|\vec{B}\|}, \tag{2.9}$$

em que $\vec{A}$ é o vetor TF-IDF da Sentença 1 e $\vec{B}$ é o vetor TF-IDF da Sentença 2.

$$\begin{aligned} \vec{A} \cdot \vec{B} &= 0.67 \cdot 0.54 + 0.33 \cdot 0.47 + 0.6 \cdot 0 + 0.71 \cdot 0.64 \\ &\quad + 0 \cdot 0.63 + 0 \cdot 0.49 + 0 \cdot 0.31 \\ &= 0.9713. \end{aligned} \tag{2.10}$$

Além disso, tem-se

$$\begin{aligned} \|\vec{A}\| &= \sqrt{0.67^2 + 0.33^2 + 0.60^2 + 0.71^2 + 0^2 + 0^2 + 0^2} \\ &= \sqrt{1.4219} \end{aligned} \tag{2.11}$$

e

$$\begin{aligned} \|\vec{B}\| &= \sqrt{0.54^2 + 0.47^2 + 0^2 + 0.64^2 + 0.63^2 + 0.49^2 + 0.31^2} \\ &= \sqrt{1.6552}. \end{aligned} \tag{2.12}$$

Fazendo as respectivas substituições de (2.10), (2.11) e (2.12) em (2.9), obtém-se

$$\text{similaridade de cosseno} = \frac{0.9713}{\sqrt{1.4219} \cdot \sqrt{1.6552}} \approx 0.6331. \tag{2.13}$$

Isso significa que as duas sentenças têm uma similaridade de cosseno de aproximadamente 0.6331, indicando que elas estão relativamente próximas em termos de conteúdo, conforme a técnica TF-IDF. Para uma categorização e análise de dados textuais, mede-se a similaridade de cosseno com uma variação entre 0 e 1, onde valores mais próximos de 1 indicam maior similaridade entre os vetores (Park et al., 2020). No caso do valor calculado de aproximadamente 0.6331, isso sugere uma semelhança relativa entre as duas sentenças. Isso ocorre porque as palavras usadas nas sentenças possuem uma certa semelhança, indicando que ambas compartilham informações e características relacionadas. Uma variação da similaridade de cosseno é a similaridade de cosseno suave, que ajusta a métrica de similaridade de cosseno para atenuar o impacto de palavras raras ou pouco frequentes nos cálculos de similaridade. Isso é feito adicionando um componente de suavização que penaliza palavras com baixas frequências.

Para a classificação de textos, é preciso definir um limiar, que é um valor numérico que determina o quão próximo dois vetores de sentenças precisam estar para serem considerados semelhantes. Esse valor é definido com base nas necessidades e nos objetivos específicos de cada tarefa. Se o objetivo é identificar pares de sentenças que compartilham um significado

muito semelhante, o ponto limiar pode ser ajustado de forma mais restrita. Por outro lado, se a tarefa exige uma abordagem mais abrangente, onde até mesmo sentenças com algum grau de semelhança sejam consideradas relevantes, o ponto limiar pode ser ajustado de maneira mais flexível. Assim, um ponto limiar muito baixo pode resultar em sentenças consideradas semelhantes quando, na verdade, possuem diferenças significativas. Por outro lado, um ponto limiar muito alto pode levar a falsos negativos, onde sentenças genuinamente semelhantes não são identificadas como tal (Alpaydin, 2020). O ajuste do ponto limiar muitas vezes é realizado por meio de experimentação e validação, utilizando conjuntos de dados de teste ou validação. Métricas de avaliação, como precisão, *recall* e *F1-Score*, podem ser utilizadas para encontrar o ponto limiar que melhor equilibra a capacidade de identificar similaridade de textos precisa e abrangente.

A similaridade de textos também pode ser verificada pela distância de Levenshtein. Porém, a distância de Levenshtein mede a diferença entre duas *strings*, através do número mínimo de operações como inserções, exclusões ou substituições de um único caractere necessárias para transformar uma *string* em outra (Berger et al., 2020). Assim, quanto menos operações realizadas, menor é a distância de Levenshtein e mais semelhantes são as strings em termos de estrutura e conteúdo.

2.5 APRENDIZADO DE MÁQUINA

O AM aborda a questão de aprimoramento de computadores por meio de experiências (Jordan e Mitchell, 2015). Os algoritmos de AM identificam padrões e correlações nos dados, viabilizando a realização de previsões ou a tomadas de decisão em relação a dados inéditos. Sua aplicabilidade abrange, com frequência, o reconhecimento de imagens, PLN e análise preditiva (Kang et al., 2020). A aplicação de técnica de AM, em geral, implica a coleta e o pré-processamento dos dados, a seleção do modelo (algoritmo) adequado para o problema específico, bem como o treinamento, avaliação, ajuste e implementação desse modelo (Kang et al., 2020). Os modelos de AM são elaborados para cumprir tarefas específicas. Dentre exemplos de tais tarefas, destacam-se: previsões de eventos futuros; classificações de dados em diferentes classes ou categorias; regressão para antever valores contínuos, com base em várias características (Kang et al., 2020). Além disso, os algoritmos de AM podem ser de diferentes tipos, como aprendizado supervisionado, não supervisionado e por reforço.

O aprendizado supervisionado representa uma abordagem em que um algoritmo é treinado por meio de um conjunto de dados que incorpora exemplos rotulados (Fontana, 2020). Cada exemplo engloba um conjunto de atributos e o rótulo correspondente à classe ou valor desejado. O algoritmo, por meio do treinamento, aprende a associar os atributos aos rótulos, tornando-se apto a realizar previsões em dados não rotulados. Dados rotulados são aqueles que contêm rótulos ou etiquetas indicativas da classe, ou categoria à qual cada amostra pertence. Os rótulos são fornecidos por especialistas humanos, contendo a resposta correta ou desejada para cada exemplo. Nesse contexto, a aprendizagem ocorre quando o algoritmo é treinado com base nesses dados rotulados, ajustando seus parâmetros para identificar padrões e relações entre os atributos dos dados e os rótulos correspondentes. O propósito é permitir que o algoritmo efetue previsões precisas para novos exemplos sem rotulação humana. Além dos dados rotulados, existem os dados sem a rotulação de um especialista, que carecem de rótulos ou etiquetas correspondentes, assim como uma categoria intermediária denominada de dados semi-rotulados, os quais combinam exemplos rotulados e não rotulados por humanos. Esses dados semi-rotulados são úteis em situações em que a rotulação completa de todos os exemplos é dispendiosa ou demorada, mas um pequeno conjunto de exemplos rotulados pode orientar a aprendizagem

(Fontana, 2020). Por sua vez, o aprendizado não supervisionado configura uma abordagem em que o algoritmo identifica padrões ou estruturas fundamentais em um conjunto de dados não rotulado por especialistas (Fontana, 2020). O intuito principal não é realizar previsões, mas sim descobrir agrupamentos naturais, redução de dimensionalidade ou outras características relevantes dos dados. O aprendizado por reforço é inspirado na psicologia comportamental, em que um *software* interage com um ambiente e aprende a tomar decisões. Essa abordagem é adotada em situações nas quais não há exemplos rotulados disponíveis e na qual o aprendizado acontece por meio de tentativa e erro. A escolha da abordagem e do modelo de classificação são determinados pela complexidade do problema a ser solucionado. No âmbito dos modelos, destacam-se, conforme Raschka et al. (2020):

• Regressão Logística (RL): a RL é um algoritmo de classificação utilizado em AM, útil quando a variável que se pretende prever é categórica, ou seja, possui diferentes categorias ou classes. A RL estima a probabilidade de uma instância (dado analisado) pertencer a uma classe específica, aplicando uma função de regressão à combinação linear das características da instância.

• *K-Nearest Neighbor* (KNN): O algoritmo KNN classifica os dados baseando-se na proximidade das instâncias. Para classificar uma nova instância, o algoritmo procura os K vizinhos mais próximos dessa instância no conjunto de treinamento. A classe da nova instância é determinada pelas classes das instâncias vizinhas. A escolha do valor K afeta a sensibilidade do algoritmo a pequenas variações nos dados.

• *Rule-Based Classifier* (RBC): O RBC é um classificador baseado em regras, que toma decisões de classificação com base em um conjunto de regras pré-definidas. Cada regra é uma combinação de condições (valores das características) que levam à classificação de uma instância em uma classe específica. Quando várias regras são aplicadas a uma instância, o classificador escolhe a classe que atende à maior quantidade de regras.

• *Long Short Term Memory* (LSTM): O LSTM é um tipo de Rede Neural Recorrente (RNN) projetado para lidar com sequências de dados, como séries temporais e texto. Ele é usado em tarefas que envolvem previsão ou classificação de sequências. No contexto de PLN, as LSTMs são frequentemente usadas para modelar dependências de longo alcance em textos e podem ser aplicadas a tarefas como tradução automática, resumo de texto e análise de sentimento.

• *Random Forest* (RF): este algoritmo se baseia na criação de um conjunto de árvores de decisão. Cada árvore é gerada a partir de uma seleção aleatória de atributos do conjunto de dados. A abordagem visa mitigar o risco de sobreajuste e aprimorar a generalização do modelo. A classificação é alcançada pela agregação das previsões de todas as árvores do conjunto. O algoritmo RF é capaz de lidar com um grande número de atributos, o que o torna uma boa opção para a classificação de texto, onde há múltiplos atributos envolvidos.

• *Naive Bayes* (NB): este modelo fundamenta-se no teorema de Bayes para efetuar a tarefa de classificação. O teorema é uma fórmula probabilística que descreve como ajustar uma probabilidade com base em novas evidências. Tal abordagem viabiliza o cálculo da probabilidade de um determinado rótulo (Berrar, 2018). O NB estabelece uma relação entre as probabilidades condicionais, mostrando-se eficaz quando empregado em situações que envolvem um amplo conjunto de atributos dos dados, como em palavras presentes em artigos de jornais.

• *Multi-Layer Perceptron* (MLP): o MLP, um tipo de rede neural artificial utilizado em tarefas de classificação, que compreende múltiplas camadas de neurônios, incluindo uma camada de entrada, uma ou mais camadas intermediárias (denominadas camadas ocultas) e uma camada de saída. Cada neurônio em uma camada se conecta a todos os neurônios na camada precedente e subsequente. Por meio do ajuste dos pesos das conexões neurais durante o processo de treinamento, o MLP é capaz de reconhecer padrões complexos nos dados de entrada e realizar

tarefas de classificação. Sua habilidade em lidar com relações não-lineares o torna uma escolha adequada para desafios de classificação de texto.

Após a aplicação de um modelo de AM, torna-se necessário avaliar sua eficácia. Diversas métricas e técnicas podem ser empregadas para tal avaliação, destacando-se a matriz de confusão, acurácia, precisão, *recall* e *F1-Score* (Kang et al., 2020). A matriz de confusão é uma tabela que expõe as classificações reais em relação às previsões efetuadas pelo modelo. Ela proporciona *insights* sobre o número de Verdadeiros Positivos (VP), que são amostras corretamente classificadas como positivas, Verdadeiros Negativos (VN), que são amostras corretamente classificadas como negativas, Falsos Positivos (FP), que são amostras erroneamente classificadas como positivas e Falsos Negativos (FN), que são amostras erroneamente classificadas como negativas, gerados pelo modelo. A acurácia quantifica a proporção de previsões corretas em relação ao total de previsões realizadas pelo modelo. Isso é calculado da seguinte maneira:

$$\text{Acurácia} = \frac{\text{VP} + \text{VN}}{\text{Total de Predições}}. \tag{2.14}$$

A precisão mede a proporção de VP em relação a todas as amostras classificadas como positivas pelo modelo, podendo ser vista em:

$$\text{Precisão} = \frac{\text{VP}}{\text{VP} + \text{FP}}. \tag{2.15}$$

A *recall* é a taxa de VP, medindo a proporção de VP em relação a todas as amostras que realmente pertencem à classe positiva. A equação que representa a *recall* é a seguinte.

$$\textit{Recall} = \frac{\text{VP}}{\text{VP} + \text{FN}}. \tag{2.16}$$

O *F1-Score* combina precisão e *recall* em uma única medida, calculando a média harmônica entre essas duas métricas. Assim, ela traz um equilíbrio entre precisão e *recall*. Essa pontuação é representada por:

$$\textit{F1-Score} = 2 \times \frac{\text{precisão} \times \textit{recall}}{\text{precisão} + \textit{recall}}. \tag{2.17}$$

A escolha das métricas depende do contexto do problema. A acurácia é útil para classes balanceadas, enquanto a precisão, *recall* e *F1-Score* são mais apropriados para classes desbalanceadas. A acurácia pode ser enganosa em situações de desequilíbrio, uma vez que um modelo pode obter alta acurácia apenas prevendo a classe majoritária. Isso pode levar a um aparente alto desempenho, que não é real em classes minoritárias. Problemas de desbalanceamento podem ser mitigados com técnicas como *oversampling* e *data augmentation*. A técnica de *oversampling* cria cópias artificiais das instâncias da classe minoritária para equilibrar as proporções entre as classes. Essas instâncias sintéticas são baseadas nas características das instâncias existentes (Tang e He, 2015). A técnica *data augmentation* também pode ser usada para criar variações artificiais nos exemplos da classe minoritária, aumentando a quantidade de dados disponíveis para treinamento e, assim, ajudando a melhorar o desempenho da representação dessas classes (Shorten e Khoshgoftaar, 2019).

2.6 *HUMAN-IN-THE-LOOP*

O processo *human-in-the-loop* refere-se à integração do conhecimento e da experiência de humanos nas modelagens de sistemas, visando aprimorar sua precisão e a eficiência (Wu et al., 2022). Ou seja, o conceito de *human-in-the-loop* refere-se a um modelo ou sistema em que os seres humanos estão ativamente envolvidos em alguma parte do processo de tomadas de decisão, validação ou controle. Esse processo é comumente aplicado a sistemas de IA e automação. O *loop* representa um ciclo de interação entre o algoritmo e o ser humano, no qual a máquina faz uma parte do trabalho e, em seguida, envolve o ser humano para revisão, orientação ou validação. O envolvimento do ser humano nessas tarefas é importante porque permite que eles forneçam conhecimento e experiência a modelos de aprendizado de máquina ou a outras etapas do processo. Os humanos podem fornecer dados de treinamento e avaliar a precisão do modelo para melhorar o desempenho do sistema (Wu et al., 2022). A pesquisa de Wu et al. (2022) relata que os humanos podem interagir diretamente com o modelo para realizar tarefas que às vezes são complexas para os computadores, como reconhecimento de imagens ou PLN. A Figura 2.1 apresenta a interação entre o usuário e o treinamento do modelo e a inferência do modelo de *human-in-the-loop* no PLN. Assim, o objetivo do processo de *human-in-the-loop* é combinar a inteligência humana e de máquina para melhorar o desempenho em modelos de sistemas de forma mais rápida (Monarch, 2021). Outro conceito usado na literatura é o *active learning* que envolve a participação ativa de seres humanos na melhoria de modelos de AM. Isto é, mesmo o sistema permanecendo no controle do processo de aprendizagem, há a intervenção dos humanos para anotação de poucos dados não rotulados (Mosqueira-Rey et al., 2023).

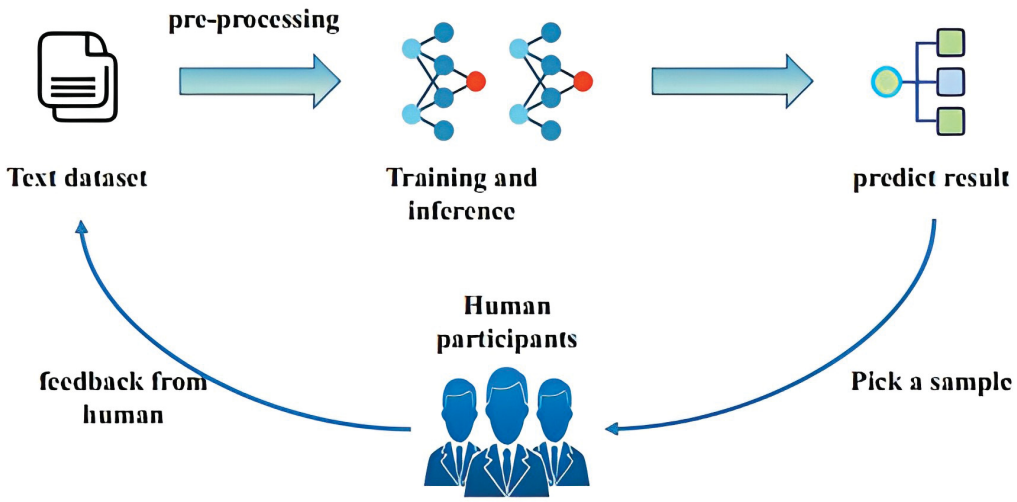


Figura 2.1: Interação entre o usuário e a inferência *human-in-the-loop* em PLN, visto em Wu et al. (2022).

2.7 CONSIDERAÇÕES

A exploração das diversas fontes de informação disponíveis desempenha um papel importante na compreensão e diversidade dos dados a serem coletados e empregados em projetos envolvendo PLN e AM. A decisão entre a extração manual e automatizada de dados é influenciada por diversos fatores, incluindo os objetivos específicos do projeto e a complexidade inerente à realização da coleta. Além disso, a etapa subsequente de pré-processamento desempenha um papel importante, pois prepara os dados para análises mais profundas e a aplicação de algoritmos

de AM voltados para a categorização, classificação e análise de texto. O pré-processamento destes dados não apenas assegura a qualidade dos resultados, mas também possibilita a extração de *insights* significativos por meio dos algoritmos de AM, aprimorando a tomada de decisões informadas. Da mesma forma, a intervenção humana nos processos de tomada de decisão podem contribuir para melhorar a precisão do sistema.

3 TRABALHOS RELACIONADOS

No presente Capítulo, são expostos os estudos correlatos que empregam a extração de dados de notícias *online*, o pré-processamento desses dados, bem como a sua classificação. Esses estudos são categorizados em seis seções distintas. A Seção 3.1 detalha trabalhos que exploram as abordagens existentes para a automação da extração de notícias como fonte de dados. A Seção 3.2 discute as pesquisas referentes ao pré-processamento dos dados extraídos. As abordagens relativas à classificação dos dados são apresentados na Seção 3.3. A Seção 3.4 conduz uma análise comparativa entre os trabalhos relacionados e o próprio ENoW. Por fim, as considerações são fornecidas na Seção 3.5.

3.1 SISTEMAS AUTOMATIZADOS DE EXTRAÇÃO DE DADOS *ONLINE*

Nesta Seção, são discutidos sistemas de extração de dados *online*. A Seção 3.1.1 aborda o Sistema de Extração e Visualização de Informações sobre Vítimas. A Seção 3.1.2 descreve a ferramenta *ParseHub*. A Seção 3.1.3 apresenta a ferramenta *80legs*. A Seção 3.1.4 explora a ferramenta *Octoparse*. A Seção 3.1.5 discute o trabalho *FactExtract*. Finalmente, a Seção 3.1.6 realiza uma análise comparativa entre os trabalhos sobre extração de dados.

3.1.1 Sistema de Extração e Visualização de Informações sobre Vítimas

O Sistema de Extração e Visualização de Informações sobre Vítimas (Chaulagain et al., 2019) surgiu como uma resposta à necessidade de se obter dados estruturados de forma ágil durante situações de crise. Esses dados são frequentemente difundidos por agências de notícias locais. A proposta do sistema abrange o desenvolvimento de uma plataforma capaz de extrair entidades relevantes relacionadas a incidentes, como fatalidades, lesões, localização do evento, veículos envolvidos, categorias de veículos, data e hora da ocorrência, a partir de fluxos de notícias. Tais dados extraídos são consolidados em um repositório e, posteriormente, apresentados ao público por meio de uma plataforma *online* dotada de interfaces de análise e visualização. Metodologicamente, o sistema combina técnicas de Reconhecimento de Entidades Nomeadas (REN), marcação de funções semânticas e expressões regulares. O sistema é dividido em três componentes principais: coleta de dados, extração de informações e análise e visualização. O componente de coleta de dados reúne artigos de notícias provenientes de diversas fontes, inclusive entradas no formato *feeds Rich Site Summary* (RSS), e os armazena de forma organizada em um banco de dados. O componente de extração de informações emprega abordagens como REN, marcação de funções semânticas e expressões regulares. O componente de análise e visualização oferece uma interface gráfica que permite a análise e visualização das informações extraídas. Através dessa interface, os usuários têm a capacidade de aplicar filtros e realizar ordenações dos dados com base em parâmetros diversos, tais como localização, tipos de veículos e datas dos incidentes. A plataforma ainda apresenta uma interface baseada em mapas, que permite a visualização geográfica da localização dos eventos. Os autores idealizam melhorias contínuas e futuras no sistema, como classificação automática de notícias vinculadas às vítimas, agregação de notícias de fontes múltiplas e validação e correção das informações extraídas, a fim de manter uma base de dados precisa e atualizada. A Figura 3.1 ilustra a arquitetura do sistema.

A avaliação e validação do desempenho do sistema foram conduzidas através da extração de informações sobre vítimas de artigos jornalísticos relacionados a acidentes rodoviários no

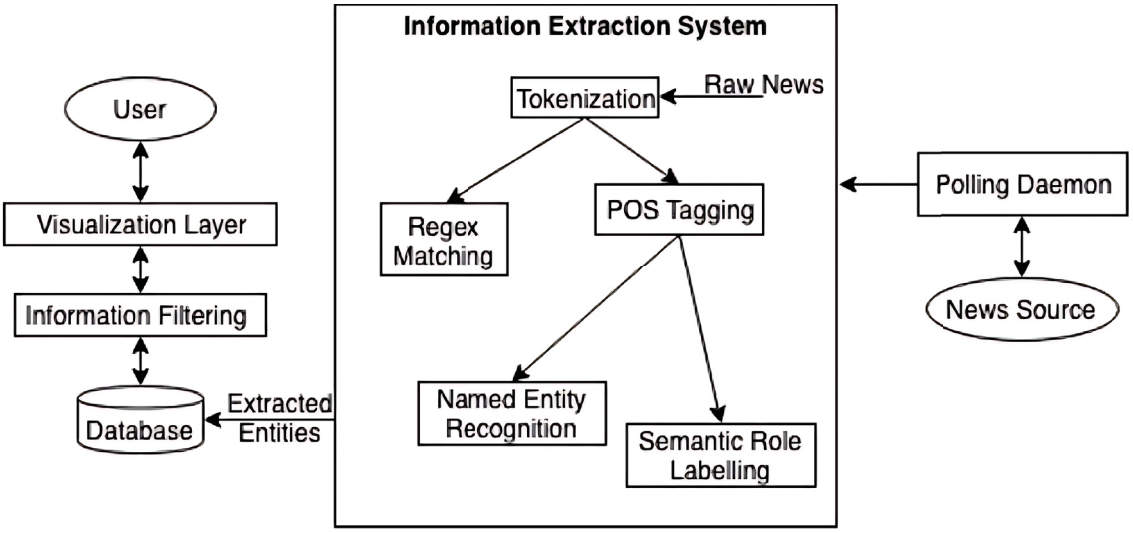


Figura 3.1: Arquitetura do Sistema de Extração e Visualização de Informações sobre Vítimas, visto em Chaulagain et al. (2019).

Nepal. As informações extraídas foram posteriormente comparadas com dados manualmente coletados pela polícia de trânsito. Os resultados obtidos evidenciam a precisão do sistema, atingindo uma acurácia de 85,70% na extração do número de óbitos e 92,30% na extração do número de feridos. O estudo também reconhece limitações, como a necessidade de contínuas atualizações na base de conhecimento do sistema e o desafio de extrair informações de artigos jornalísticos com estruturas de frases complexas. Em síntese, o trabalho proporciona uma contribuição ao tratar da extração e visualização automatizadas de informações estruturadas sobre vítimas a partir de artigos jornalísticos, aprimorando as capacidades de gestão de crises. O Sistema de Extração e Visualização de Informações sobre Vítimas usa uma combinação de técnicas de PLN e algoritmos de AM para extrair informações relacionadas a vítimas de notícias. O sistema não depende de palavras-chave específicas para pesquisar artigos de notícias, diferentemente do ENoW, que usa *strings* para a realização da coleta. O sistema também pode extrair informações de qualquer artigo de notícias que relate incidentes envolvendo vítimas. Já o ENoW realiza coleta de notícias sobre temas relacionados à *string* escolhida pelo usuário. Para o armazenamento dos dados coletados pelo sistema, um banco de dados é utilizado, da mesma forma que ocorre no ENoW. Também não é informado se o sistema armazena e disponibiliza os metadados das páginas de artigos jornalísticos.

3.1.2 *ParseHub*

O *ParseHub*[1] é uma ferramenta de extração de dados da *Web* que permite coletar informações de *sites*, estruturando essas informações em planilhas, bancos de dados ou JSON. As funcionalidades dessa ferramenta estão disponíveis em modalidades gratuita e paga. As funcionalidades gratuitas abrangem a criação e configuração de projetos de extração, com regras e treinamento do programa de extração; um número limitado de páginas coletadas por projeto por mês; a exportação dos dados coletados em formatos CSV e JSON. As funcionalidades pagas abrangem a criação e configuração de projetos de extração, com regras e treinamento do programa de extração; um número ilimitado de páginas coletadas por projeto por mês; o agendamento para coletas automáticas. O *ParseHub* é uma ferramenta de *Web Scraping* que concentra-se principalmente na coleta automatizada de dados de *sites*. No entanto, o *ParseHub* não é projetado

[1]https://www.parsehub.com/

para realizar pré-processamento avançado dos dados coletados. Ele fornece funcionalidades básicas de limpeza de dados, como remoção de tags HTML, eliminação de espaços em branco extras, entre outros ajustes simples. Caso haja a necessidade de se obter dados extraídos e com maiores pré-processamentos, como análise de linguagem natural, normalização de texto, detecção de entidades nomeadas, remoção de ruídos ou transformações mais sofisticadas, é necessário utilizar outras ferramentas ou bibliotecas de PLN que ofereçam técnicas de análise de dados após a coleta de dados com o *ParseHub*. Em resumo, o *ParseHub* é mais voltado para a coleta de dados da *Web*. O *software ParseHub* demonstra semelhanças com a abordagem ENoW no que concerne ao procedimento de extração de dados, usando *strings* como entrada. Os dados coletados são em formato de texto, assim como ocorre no ENoW, porém, esses dados são disponibilizados em formatos CSV e JSON, enquanto o ENoW os armazena em um banco de dados. Além disso, diferentemente do ENoW, o *ParseHub* não disponibiliza metadados das páginas coletadas.

3.1.3 *80legs*

A ferramenta *80legs*[2] oferece um serviço de extração de dados que permite criar tarefas personalizadas de extração, definindo padrões de URLs e regras de execução de coleta. Os dados coletados podem ser exportados em diferentes formatos, como CSV, JSON ou XML. As funcionalidades pagas abrangem suporte e personalizações avançadas da ferramenta. No entanto, assim como qualquer ferramenta, é importante considerar a natureza dos requisitos da aplicação e a complexidade das análises que se pretende fazer após a coleta dos dados. Para processos mais sofisticados, como análise de linguagem natural, detecção de padrões complexos ou análises estatísticas avançadas, pode ser necessário usar ferramentas ou bibliotecas especializadas adicionais após a coleta de dados com o *80legs*. O sistema *80legs* apresenta similaridades com a abordagem ENoW no âmbito do processo de extração de dados usando *strings* como entrada. Os dados extraídos também são em formato de textos, assim como ocorre no ENoW. Porém, os dados são disponibilizados nos formatos CSV, JSON ou XML, no *80legs*, diferentemente do armazenamento do ENoW, que usa um banco de dados. Outra diferença é que o *80legs* não deixa acessível ao usuário os metadados dos *sites* onde a coleta é realizada.

3.1.4 *Octoparse*

O *Octoparse*[3] é uma ferramenta que permite extrair dados de *sites*, mas com limitações, em sua forma gratuita, no que diz respeito ao número de páginas que podem ser coletadas. Os dados coletados podem ser exportados em formatos como CSV, Excel e HTML. Na modalidade paga, o número de páginas que podem ser extraídas é ilimitado. Também há a permissão de agendamento de tarefas de extração para serem executadas automaticamente em intervalos definidos. No entanto, assim como acontece com outras ferramentas de *Web Scraping*, caso haja necessidade de processamento dos dados, após a extração, como análise de linguagem natural e normalização de texto, outras ferramentas precisam ser consideradas. O programa *Octoparse* exibe semelhanças com a abordagem ENoW no procedimento de extração de dados com uso de *strings*, retornando dados textuais ao usuário. Mas os dados são disponibilizados nos formatos CSV, Excel e HTML, o que não ocorre no ENoW, que os armazena e disponibiliza em um banco de dados. O *Octoparse* também não disponibiliza o acesso a metadados das páginas em que os dados foram extraídos.

[2] https://80legs.com/

[3] https://www.octoparse.com/

3.1.5 *FactExtract*

O *FactExtract* (Sarr et al., 2018) é um extrator automático de artigos, implementado em quinze *sites* de notícias senegaleses. O *FactExtract* possui a proposta de tornar o processo de checagem de dados jornalísticos mais eficiente e menos trabalhoso. A metodologia envolve o uso de *Web Scraping* para recuperar dados específicos e estruturados com esforço humano reduzido. Sua implementação é composta por três módulos principais: o módulo de coleta, o módulo de extração de fatos e o módulo de apresentação de resultados. O módulo de coleta é responsável por coletar artigos de jornais *online* de várias fontes. O módulo de extração de fatos é responsável por extrair informações específicas e estruturadas de cada artigo coletado. O módulo de apresentação de resultados é responsável por apresentar os resultados da extração. A Figura 3.2 apresenta a arquitetura do sistema *FactExtract*.

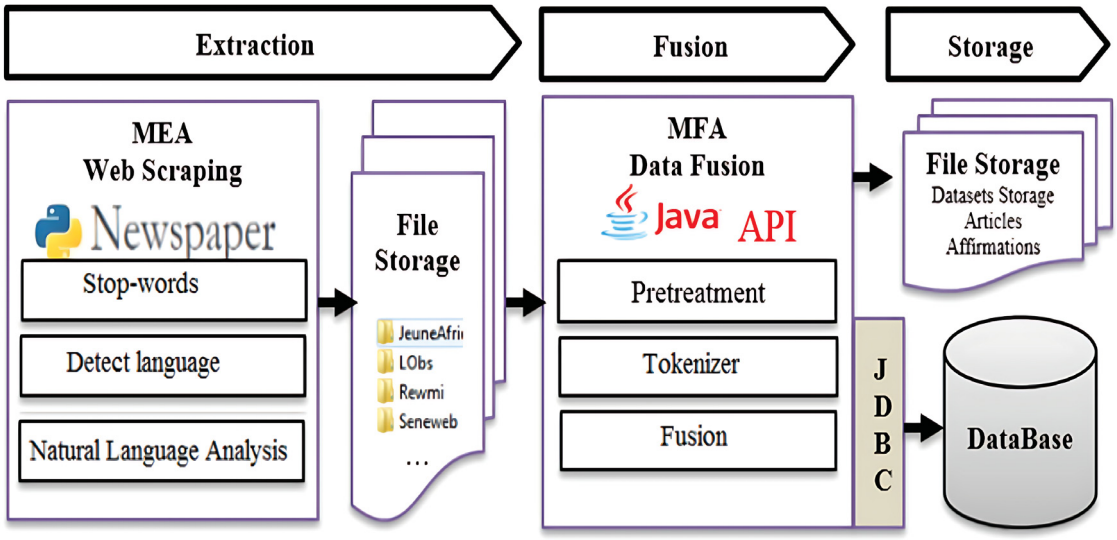


Figura 3.2: Arquitetura do Sistema *FactExtract*, visto em Sarr et al. (2018).

Os resultados de extração de dados específicos foram considerados bons e com esforço humano reduzido. Eles também idealizam trabalhos futuros envolvendo o aumento do tamanho dos conjuntos de dados. A coleta realizada pelo sistema *FactExtract* é feita a partir de uma lista de quinze *sites* selecionados e não por meio de uma *string*, como ocorre no ENoW. Os dados coletados são textuais e são armazenados em um banco de dados, o que também ocorre no ENoW. Além disso, o *FactExtract* não informa se os metadados das páginas coletadas são disponibilizados aos usuários.

3.1.6 Análise Comparativa entre os Trabalhos sobre Extração de Dados

A análise comparativa dos trabalhos que envolvem a extração de dados está apresentada na Tabela 3.1. A comparação é realizada com cinco trabalhos relacionados à extração de dados. Cada trabalho possui diferenças em termos de entrada para a coleta, saída da coleta, limitação de URLs e acesso a metadados.

Trabalhos	**Entrada para a coleta**	**Saída da coleta**	**Limitação de URLs**	**Acesso a metadados**
(Chaulagain et al., 2019)	artigos de notícias sobre incidentes envolvendo vítimas	banco de dados	não possui	não informado
ParseHub	*string* de busca	arquivos	não possui	não possui
80legs	*string* de busca	arquivos	não possui	não possui
Octoparse	*string* de busca	arquivos	não possui	não possui
(Sarr et al., 2018)	lista de URLs	banco de dados	quinze URLs	não informado
ENoW	*string* de busca	banco de dados	não possui	possui

Tabela 3.1: Comparação de trabalhos relacionados à extração de dados.

O trabalho de Chaulagain et al. (2019), utiliza artigos de notícias sobre incidentes envolvendo vítimas como entrada para a coleta dos dados. A saída da coleta é armazenada em um banco de dados. Este trabalho não possui limitações de URLs especificadas e não informa se há acesso a metadados. As ferramentas *ParseHub*, *80legs* e *Octoparse* utilizam uma *string* de busca como entrada para a coleta e a saída é obtida em forma de arquivos. Não há limitações de URLs mencionadas e nenhum deles possui acesso a metadados. O trabalho de Sarr et al. (2018) usa uma lista de URLs como entrada para a coleta, e os dados são armazenados em um banco de dados como saída. Há uma limitação de quinze URLs para a coleta de dados. No entanto, não é especificado se há acesso a metadados. Por fim, o ENoW também utiliza uma *string* de busca como entrada para a coleta, e a saída é armazenada em um banco de dados. Ele não possui limitações de URLs e possui acesso a metadados.

3.2 PRÉ-PROCESSAMENTO DE DADOS

Neste segmento, são expostas as pesquisas que incorporam PLN em suas análises. A Seção 3.2.1 descreve um sistema voltado à detecção de notícias falsas. A Seção 3.2.2 relata estas funcionalidades do FactExtract, que emprega PLN em seus dados. A Seção 3.2.3 apresenta uma análise comparativa entre os trabalhos sobre o pré-processamento dos dados.

3.2.1 Sistema de Detecção de Notícias Falsas

O sistema para detecção de notícias falsas em jornais na língua tailandesa (Meesad, 2021) usa RI, PLN e AM. O estudo inclui fases de coleta de dados e construção de modelos de AM, e compara vários modelos de classificação quanto à precisão. A coleta de dados é realizada a partir de *sites* de notícias *online* tailandeses usando RI. As informações são recuperadas com o uso de um extrator de *links* para páginas de notícias. O extrator verifica se a página não foi visitada, realiza a busca e extrai o conteúdo da notícia. Esse conteúdo é armazenado em um banco de dados para análise posterior. O sistema extrator não é informado se foi desenvolvido pelos autores. Após coletados, os dados são pré-processados com etapas de limpeza, remoção de *stop words*, entre outras. A Figura 3.3 apresenta o fluxo do PLN no sistema.

Para avaliar as etapas do PLN, modelos de AM foram usados para comparação de desempenho. A avaliação demonstrou que as etapas do PLN foram eficazes na extração de características relevantes para classificação das notícias em categorias *(i)* reais, *(ii)* suspeitas ou

(iii) falsas. As técnicas de PLN usadas no sistema assemelham-se às usadas pelo ENoW, porém, o ENoW também usa REN, que não é citado pelos autores.

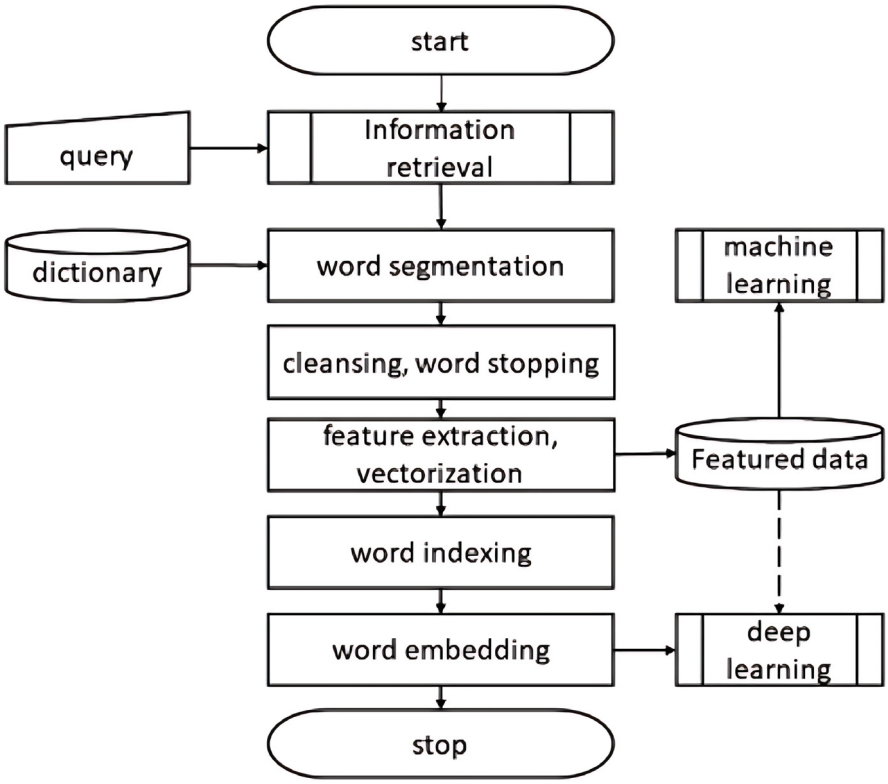


Figura 3.3: PLN no sistema para detecção de notícias falsas em jornais na língua tailandesa, visto em Meesad (2021).

3.2.2 *FactExtract*

O *FactExtract* (Sarr et al., 2018) utiliza técnicas de PLN para extrair informações específicas e estruturadas de cada artigo jornalístico coletado. Os autores mencionam que o módulo que executa o PLN do *FactExtract* utiliza técnicas de *tokenização*, análise morfológica, análise sintática e análise semântica para identificar e extrair informações relevantes de cada artigo. Além disso, a ferramenta utiliza bibliotecas de programação, como o *Newspaper* para realizar tarefas de PLN. O sistema *FactExtract* assemelha-se à abordagem ENoW em relação às técnicas de PLN empregadas. Também se assemelha na extração de informações dos textos, usando a técnica de REN.

3.2.3 Análise Comparativa entre os Trabalhos sobre Pré-processamento

Os três trabalhos comparados envolvem as etapas de PLN e extração de informações. Meesad (2021) concentram-se em etapas de PLN, incluindo limpeza, indexação e incorporação de palavras, mas não especifica sobre técnica de extração de informações. Por outro lado, os trabalhos (Sarr et al., 2018) e ENoW aplicam etapas de limpeza, *stemming* e *lemmatization*. Ambos utilizam o método de extração de informações REN. A análise comparativa dos trabalhos está apresentada na Tabela 3.3.

Trabalhos	Etapas de PLN	Extração de informações
(Meesad, 2021)	limpeza, indexação e incorporação de palavras	não informado
(Sarr et al., 2018)	limpeza, *stemming* e *lemmatization*	REN
ENoW	limpeza, *stemming* e *lemmatization*	REN

Tabela 3.2: Comparação de trabalhos relacionados ao processamento dos dados.

3.3 CLASSIFICAÇÃO DE NOTÍCIAS

As pesquisas que envolvem a classificação de notícias são apresentadas nesta seção. A Seção 3.3.1 apresenta a categorização de artigos curtos de notícias. A Seção 3.3.2 relata a classificação de notícias usando AM. A Seção 3.3.3 apresenta a análise comparativa entre os trabalhos sobre classificação de notícias.

3.3.1 Categorização de Artigos de Notícias

Métodos de cálculo de similaridade entre textos foram empregados no experimento conduzido por Sitikhu et al. (2019), que propõe a categorização de artigos de notícias curtos de um conjunto de dados. O trabalho foi realizado utilizando o conjunto de dados *AG's news topic classification*[4], o qual consiste em artigos curtos de notícias. Cada artigo de notícias foi categorizado em quatro classes: Mundo, Esportes, Negócios e Ciência/Tecnologia. As categorias específicas ou tópicos dentro dessas classes não são mencionados por Sitikhu et al. (2019). Para o experimento, uma comparação entre três abordagens diferentes para medir a similaridade semântica entre duas notícias de texto curto foi realizada. Os conjuntos de dados relacionados com notícias da AG foram utilizados para verificar a experiência. As três abordagens são similaridade de cosseno com vetores TF-IDF, similaridade de cosseno com vetores *Word2Vec*, similaridade de cosseno suave com vetores *Word2Vec*. Todos os métodos utilizados obtiveram bons resultados. Na validação, o cosseno com TF-IDF teve a maior precisão. O ENoW também usa a similaridade de cosseno com vetores TF-IDF, além de utilizar os classificadores NB, RF e MLP em AM.

3.3.2 Classificação de Notícias Utilizando Aprendizado de Máquina

A estrutura proposta para a detecção de notícias falsas na Tailândia (Meesad, 2021) é apresentada na Figura 3.4 e se fundamenta em três módulos centrais: RI, PLN e AM.

[4]https://www.kaggle.com/datasets/amananandrai/ag-news-classification-dataset

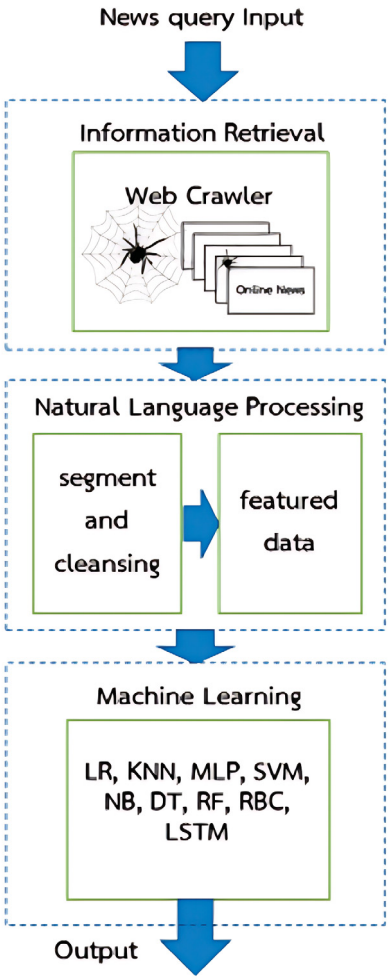


Figura 3.4: Processo de detecção de notícias falsas com a estrutura de aprendizado de máquina, visto em Meesad (2021).

Na fase de coleta de dados, os pesquisadores adquiriram informações de *sites* de notícias *online* no idioma tailandês por meio de um rastreador *Web* utilizado na RI. Esses dados foram submetidos à segunda fase, usando técnicas de PLN para a extração de características relevantes. Essas técnicas incluíram segmentação de palavras, limpeza dos dados, remoção de *stop words*, extração de características, indexação e *word embedding*. As notícias foram classificadas em três categorias: *(i)* reais, *(ii)* suspeitas e *(iii)* falsas. O processo de vetorização foi realizado, mas não tem especificação sobre qual técnica foi utilizada. Diversos modelos de classificação de AM foram explorados no estudo, tais como RL, KNN, NB, MLP, RF, RBC e LSTM. O desempenho desses modelos foi comparado usando métricas como acurácia, precisão, *recall* e *F1-Score*. O modelo LSTM obteve o melhor desempenho. Em comparação com o ENoW, os modelos NB, RF e MLP também foram utilizados, porém, o ENoW utilizou também a distância de cosseno para a classificação das notícias. A vetorização do ENoW é realizada com a técnica TF-IDF.

3.3.3 Análise Comparativa entre os Trabalhos sobre Classificação de Dados

Sitikhu et al. (2019) exploram técnicas de vetorização como TF-IDF e *Word2Vec*, mas não aplica métodos de classificação baseados em AM. Em vez disso, utiliza a distância de cosseno para classificação. Meesad (2021) utiliza TF-IDF para vetorização e uma variedade de métodos de classificação baseados em AM, incluindo RL, KNN, NB, MLP, RF, RBC e LSTM. O ENoW

também utiliza TF-IDF para vetorização e aplica métodos de classificação com AM, incluindo NB, RF e MLP, além de usar a distância de cosseno para classificação sem AM. Essa análise comparativa destaca as abordagens de vetorização e classificação de dados entre os trabalhos listados, incluindo as técnicas de vetorização, métodos de classificação com e sem AM, podendo ser visualizada na Tabela 3.3.

Trabalhos	**Vetorização**	**Classificação com AM**	**Classificação sem AM**
(Sitikhu et al., 2019)	TF-IDF *Word2Vec*	não se aplica	distância de cosseno
(Meesad, 2021)	TF-IDF	RL, KNN, NB MLP, RF RBC, LSTM	não se aplica
ENoW	TF-IDF	NB, RF, MLP	distância de cosseno

Tabela 3.3: Comparação de trabalhos relacionados à classificação dos dados.

3.4 ANÁLISE COMPARATIVA

Os estudos foram categorizados de acordo com as atividades associadas ao desenvolvimento de um sistema de extração de dados, englobando etapas de pré-processamento e classificação dos dados. Foram submetidos à análise comparativa um total de sete trabalhos, dos quais três constituem ferramentas acessíveis *online*. Os trabalhos de Chaulagain et al. (2019) e Sarr et al. (2018), além das ferramentas *ParseHub*, *80legs* e *Octoparse*, dedicam-se ao processo de extração de dados provenientes da *Web*. Os estudos de Sitikhu et al. (2019) e (Meesad, 2021) não indicam explicitamente se contemplam essa funcionalidade em seus contextos.

Entre os trabalhos que empregam o PLN estão os de Chaulagain et al. (2019), Sarr et al. (2018), Sitikhu et al. (2019) e (Meesad, 2021). No âmbito da classificação dos dados, os autores Sitikhu et al. (2019) e (Meesad, 2021) desempenham esses papéis. Porém, o primeiro classifica os dados com a similaridade dos textos, usando o cálculo da distância de cosseno, e o segundo classifica os dados usando algoritmos classificadores em AM. A Tabela 3.4 apresenta os trabalhos relacionados, conforme as funcionalidades oferecidas. O ENoW contempla todas as funcionalidades de extração, processamento, vetorização e classificação dos dados, mantendo a proveniência dos dados coletados. Além disso, o ENoW possui extensibilidade em termos de fontes de notícias e projeto, permitindo coletas de diferentes assuntos, conforme a escolha do usuário.

Trabalhos	**Extração automatizada dos dados**	**Pré-processamento dos dados**	**Classificação dos dados**
(Chaulagain et al., 2019)	X	X	
ParseHub	X		
80legs	X		
Octoparse	X		
(Sarr et al., 2018)	X	X	
(Meesad, 2021)		X	X
(Sitikhu et al., 2019)		X	X
ENoW	X	X	X

Tabela 3.4: Comparação de trabalhos relacionados.

3.5 CONSIDERAÇÕES

Neste capítulo, foram apresentados os estudos relacionados à extração automatizada, ao pré-processamento e à classificação de dados. Uma análise das características inerentes a cada pesquisa foi conduzida, revelando as semelhanças e diferenças em relação ao ENoW. O ENoW se fundamenta em conceitos extraídos desses estudos correlatos para criar um sistema de extração de dados de notícias de jornais *online*. Ao empregar o PLN, o ENoW filtra as notícias de interesse do usuário.

O processo de filtragem é necessário porque a pesquisa por *strings* de busca pode retornar notícias que não correspondem ao assunto de interesse do usuário. O ENoW dá suporte a duas abordagens de filtragem: **Filtragem com Intervenção Humana** e **Classificação Automática**. A primeira abordagem é utilizada quando não há uma base de notícias previamente rotulada por especialistas como sendo de interesse do usuário ou não. A existência de uma base rotulada por humanos permite o treinamento de um modelo de AM e, consequentemente, a utilização da segunda abordagem.

4 ENOW - EXTRATOR DE NOTÍCIAS DA WEB

Esta Seção apresenta o ENoW, uma ferramenta de extração de dados da *Web* projetada para a obtenção de dados de páginas de jornais *online*. Essa ferramenta também permite aos usuários a personalização das coletas por meio da inserção de *strings* contendo tópicos de interesse. Após a conclusão da coleta, é realizado um processo de filtragem das notícias, para remover aquelas que não são do assunto de interesse do usuário. As informações detalhadas sobre a apresentação do ENoW estão abordadas nas próximas três Seções. A Seção 4.1 explora a arquitetura global do sistema. A Seção 4.2 apresenta a modelagem dos dados empregada na sua construção. A Seção 4.3 explica como o ENoW opera. Por fim, a Seção 4.4 aborda considerações relevantes sobre o ENoW.

4.1 ARQUITETURA GERAL DO ENOW

A arquitetura geral do ENoW desdobra-se em três módulos principais: a etapa de registro e coleta, a etapa de pré-processamento dos dados e a etapa de classificação dos dados, conforme ilustrado na Figura 4.3.

O módulo de registro e coleta inclui o processo de cadastro de *sites* de notícias, com a função de alimentar a base de *Sites*, que contém informações sobre jornais *online*, que incluem dados sobre como extrair as informações de interesse. Esta base é composta pelas tabelas *sites* de notícias, campo, início da estrutura de notícias e estrutura geral de notícias. Este módulo também contempla o cadastro de projetos e *strings*. Ele é necessário porque o ENoW considera que podem existir diversos projetos, cada um associado a um conjunto de *strings* de busca e a um conjunto de jornais dos quais se deseja extrair as notícias. Estas informações são armazenadas na base de *Projetos e Strings*. Após o cadastro do projeto, o coletor de notícias pode ser iniciado. Os dados coletados na *Web* são armazenados em uma base de dados de *Notícias*. Os dados extraídos de cada notícia incluem o título, descrição (resumo), data, localização, imagem e texto completo. Cada entrada nesta base é também associada ao jornal do qual foi extraído e a um ou mais projetos. Um histórico de coletas também é armazenado em uma base de *Logs*. Cada entrada nessa base faz associação às notícias coletadas, armazenando a data da coleta e a disponibilidade de cada página *online*.

O módulo de pré-processamento efetua operações de PLN nos conteúdos textuais das notícias. Isso inclui a identificação e extração de entidades relevantes, como localizações e datas associadas à ocorrência da notícia no texto original. Estas informações são devidamente armazenadas nos campos apropriados das tabelas no banco de dados. Caso haja mais de uma localização, o armazenamento é realizado em forma de lista, sem haver perdas de locais citados no texto da notícia completa. Posteriormente, uma vez que os dados brutos são devidamente limpos e processados, o módulo executa a etapa de vetorização, que consiste em atribuir valores numéricos às palavras mais significativas presentes no texto, facilitando sua representação para o módulo seguinte. Esse processo de vetorização é realizado com a técnica TF-IDF.

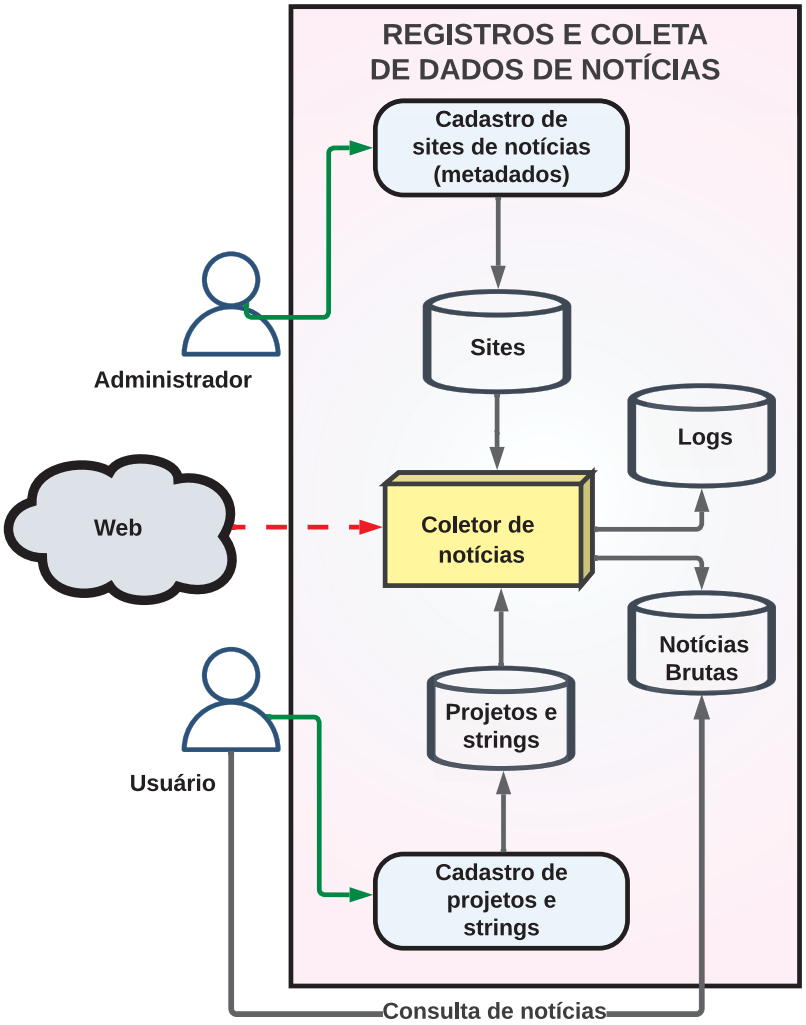


Figura 4.1: Arquitetura do módulo de registro e coleta do ENoW.

Por fim, o módulo de classificação dos dados emprega os vetores previamente gerados para classificar as notícias, alinhando-se com a seleção efetuada pelo usuário. Em outras palavras, o usuário faz a seleção de determinadas notícias consideradas relevantes, dentre dez apresentadas a ele. Os vetores das notícias selecionadas serão incluídas no grupo representativo de notícias relevantes. Os vetores das notícias não escolhidas serão incluídos no grupo representativo de notícias não relevantes. Então é desencadeado um processo de classificação. Este processo, fundamentado na similaridade de textos, recorre à medida de distância de cosseno aplicada aos vetores TF-IDF com o intuito de avaliar a semelhança entre dois textos, considerando suas respectivas estruturações de conteúdo. A distância de cosseno avalia a concordância entre os textos, utilizando o ângulo entre os vetores de representação como parâmetro: quanto menor o ângulo, maior a semelhança. É válido destacar que esse processo de classificação não é inflexível; o ENoW oferece a possibilidade ao usuário de avaliar a precisão da classificação das notícias de acordo com suas seleções iniciais. Desse modo, caso a classificação não esteja alinhada com as preferências do usuário, o ENoW permite a repetição desse processo quantas vezes forem necessárias até que as notícias estejam categorizadas de acordo com as suas preferências. Cada iteração de classificação registra o desempenho por meio das métricas de acurácia, precisão, *recall* e *F1-Score*.

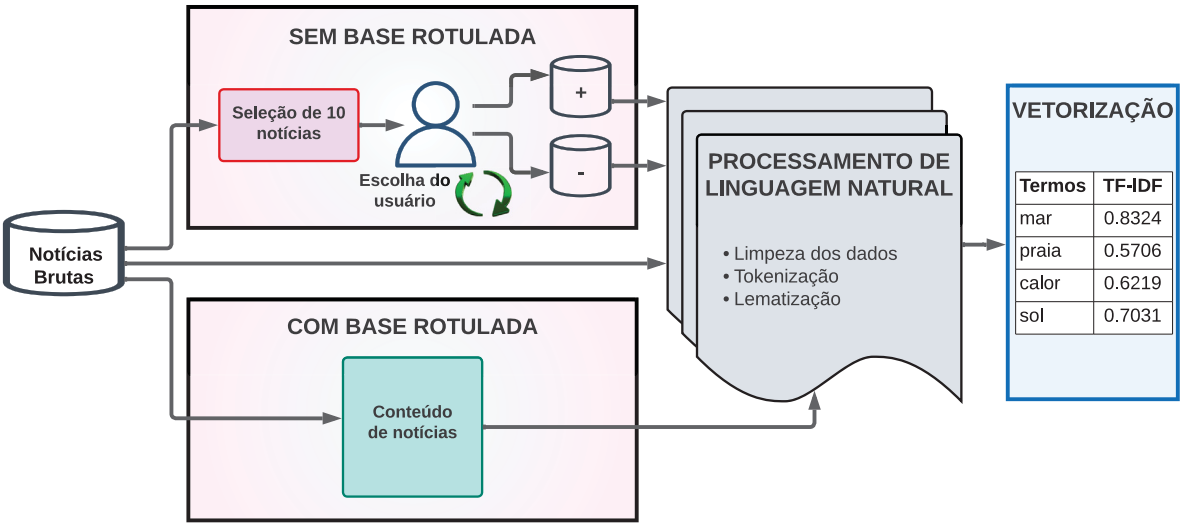


Figura 4.2: Arquitetura do módulo de pré-processamento do ENoW.

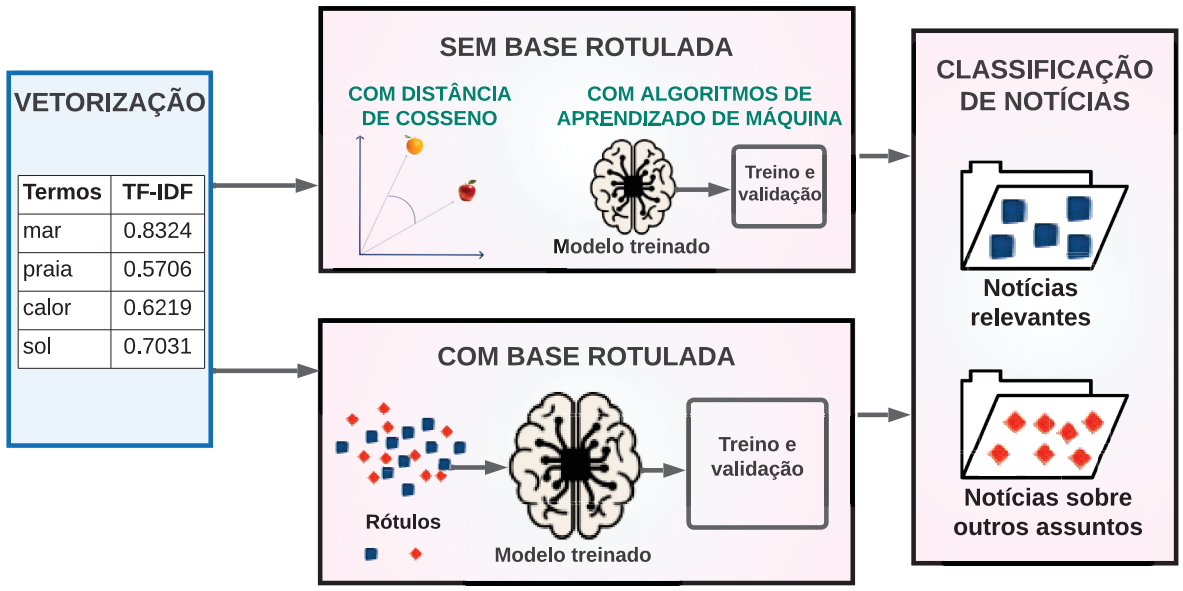


Figura 4.3: Arquitetura do módulo de classificação do ENoW.

4.2 MODELAGEM DOS DADOS

A estrutura da base de dados do ENoW é constituída por treze tabelas principais e mais cinco tabelas associativas. Essa estrutura é organizada em quatro segmentos distintos para proporcionar uma compreensão mais clara do modelo: (*i*) *sites*, (*ii*) projetos e *strings*, (*iii*) coleta de notícias e (*iv*) classificação das notícias coletadas. O detalhamento das tabelas que integram cada segmento é apresentado a seguir. A Seção 4.2.1 apresenta a estrutura de armazenamento dos *sites*. A Seção 4.2.2 detalha o armazenamento de projetos e *strings*. O armazenamento da coleta das notícias está apresentado na Seção 4.2.3. Por fim, a base de classificação está detalhada na Seção 4.2.4 Ressalta-se que a tabela denominada `conteudo_noticia` é uma presença constante em todas as Figuras que ilustram o modelo, contribuindo para a compreensão da relação global; entretanto, ela é detalhada dentro do segmento relacionado a logs.

4.2.1 Base de *Sites*

Iniciando com a seção referente aos *sites*, esta abrange um total de seis tabelas principais em sua composição. A tabela `estado` é destinada ao registro de informações referentes aos estados brasileiros. Isso abrange elementos como códigos do IBGE, nomes, siglas e regiões geográficas.

Na tabela `cidade` estão armazenados os dados relacionados às cidades brasileiras. São incluídos códigos do IBGE, nomes e coordenadas de latitude e longitude. Para manter a conexão com o respectivo estado, essa tabela conta com uma chave estrangeira que remete ao identificador da tabela `estado`.

A tabela `site_noticia` desempenha o papel de armazenar informações importantes e metadados relacionados às páginas de jornais. Seus atributos abrangem uma diversidade de detalhes. Atributos como nome, URL, estado, país e UF servem para capturar detalhes específicos do jornal, enquanto o atributo que indica o acesso à página interna aponta se a versão completa do conteúdo da notícia está disponível para acesso. O atributo `tipo_paginacao` categoriza a lógica para acessar outras páginas subsequentes da lista de notícias apresentada pelo jornal. Os atributos `json_args` e `req_response` armazenam informações relevantes para a lógica essencial à paginação. Vale mencionar que essa tabela estabelece uma relação com a tabela `estado` por meio de uma chave estrangeira, indicando a qual estado o jornal pertence.

A tabela `campo` é responsável por registrar os diversos tipos de elementos que constituem uma notícia. Isso inclui campos como título, descrição, conteúdo, informações de data (dia, mês e ano), localização, imagem e URL.

A tabela `init_estrutura_noticia` armazena informações para cada início de estrutura de notícia. Isso inclui a *tag* HTML inicial da lista de notícias, o caminho da notícia, uma chave estrangeira que conecta esse início de estrutura à página de notícia correspondente (por meio do identificador do *site* de notícias) e a data em que essas informações foram inseridas no banco de dados.

A tabela `estrutura_noticia` contém outras *tags* e caminhos relacionados a cada tipo de campo presente na notícia. As chaves estrangeiras estabelecem conexões com a tabela `init_estrutura_noticia`, representando a relação hierárquica entre as *tags* iniciais e suas *subtags*. A Figura 4.4 apresenta a estrutura em HTML de um jornal e como suas *tags* e *subtags* são armazenadas no ENoW. O esquema de base da parte de *sites* é representado na Figura 4.5.

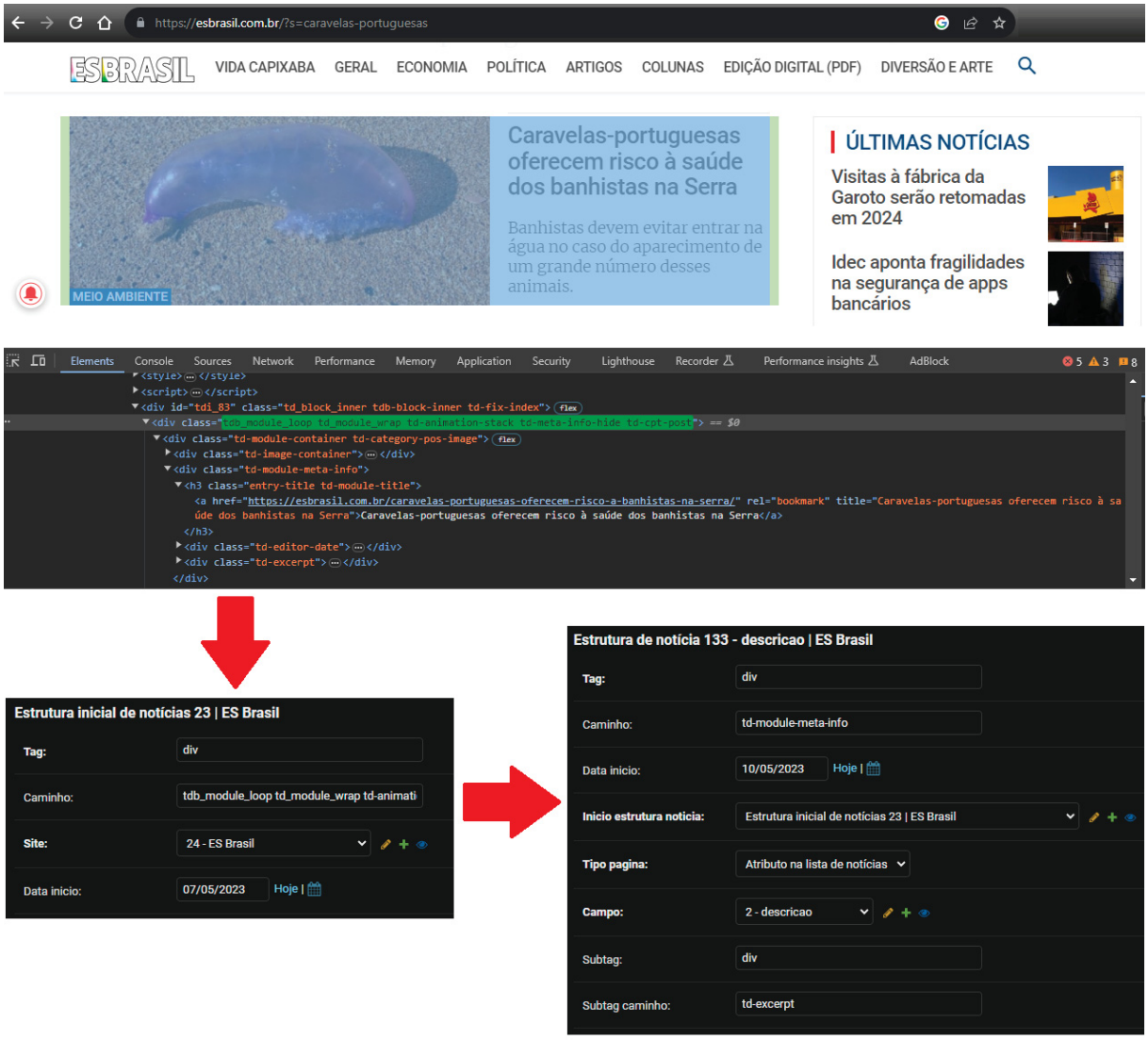


Figura 4.4: Estrutura de um jornal e armazenamento de suas *tags* e *subtags*.

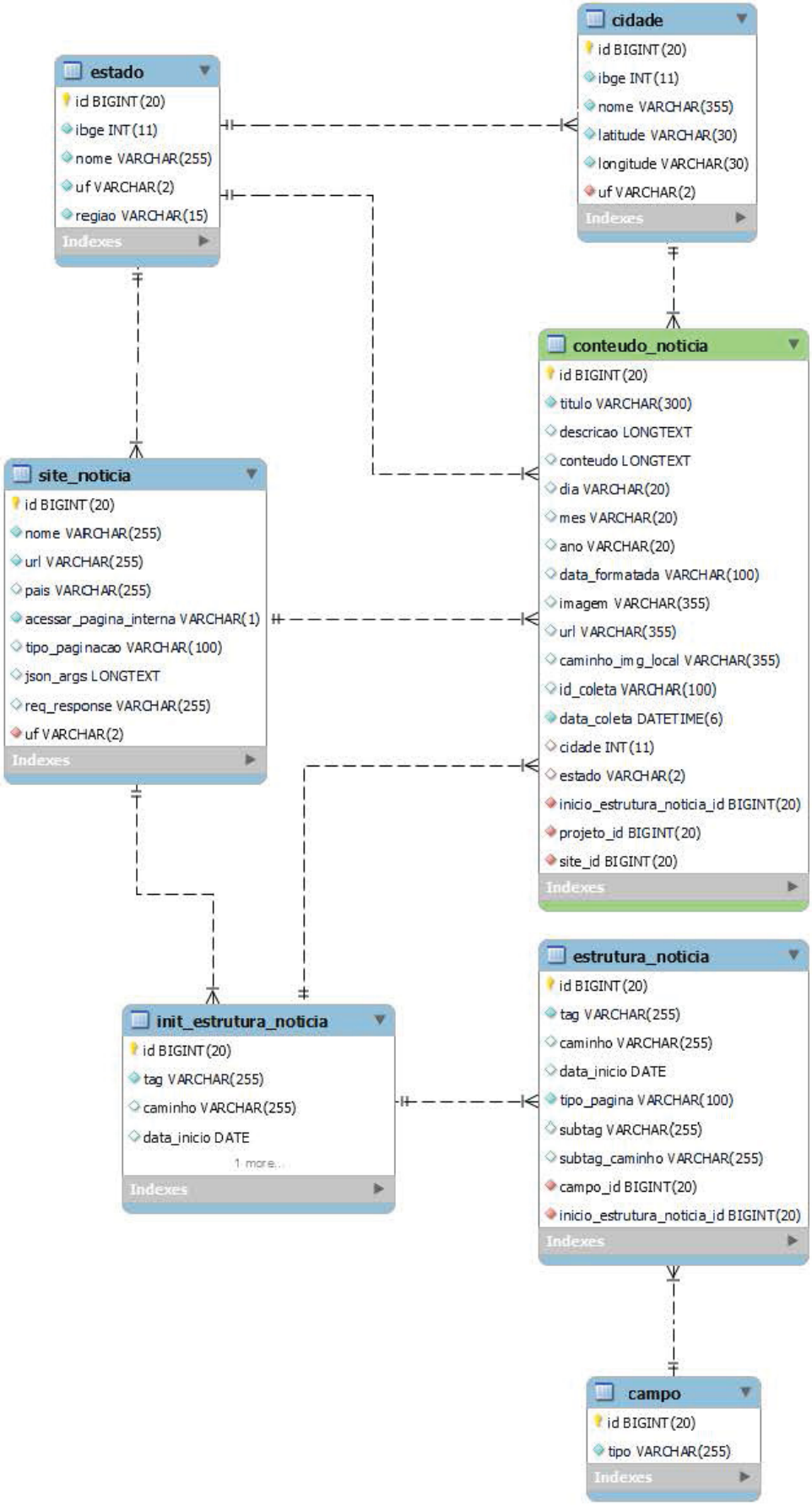


Figura 4.5: Esquema de base de *sites*.

4.2.2 Base de Projetos e *Strings*

A parte de projetos e *strings* é composta por cinco tabelas, sendo duas principais e três associativas. A tabela `projeto` armazena os projetos registrados. Esses projetos são acompanhados de informações como seus respectivos nomes, datas de início e indicações de ativação.

A tabela `palavra_chave` registra as sequências de caracteres utilizadas para buscas. Ela engloba tanto a própria sequência de caracteres quanto a data de registro. As tabelas `projeto_site`, `projeto_palavra_chave` e `conteudo_noticia_palavra_chave` são as tabelas associativas.

A tabela `projeto_sites` estabelece uma conexão entre os identificadores presentes nas tabelas `projeto` e `site`. Esse relacionamento se evidencia em situações em que há a possibilidade de existirem uma ou mais páginas de notícias para cada projeto. Similarmente, também é possível ter um ou mais projetos que empregam a mesma página de jornal para a realização de coletas. Analogamente, um cenário semelhante se aplica às tabelas `projeto` e `palavra_chave`. Essas tabelas têm seus identificadores atuando como chaves estrangeiras na tabela `projeto_palavra_chave`. Nesse contexto, um projeto tem a capacidade de utilizar mais de uma *string* para executar suas coletas, enquanto uma mesma *string* pode ser associada a mais de um projeto.

A tabela `conteudo_noticia_palavra_chave` desempenha um papel associativo semelhante às tabelas mencionadas anteriormente. Ela é composta por chaves estrangeiras provenientes das tabelas `conteudo_noticia` e `palavra_chave`. Essa estrutura reflete a natureza das notícias coletadas, onde cada notícia pode estar associada a uma ou várias *strings* de coleta. Da mesma forma, mais de uma notícia pode compartilhar a mesma *string*, criando assim um relacionamento entre as notícias e as palavras-chave correspondentes.

A Figura 4.6 representa o armazenamento dos projetos e das *strings* de busca, bem como a vinculação com os jornais de interesse para a realização da coleta. O esquema de base da parte de projetos e *strings* é representado na Figura 4.7.

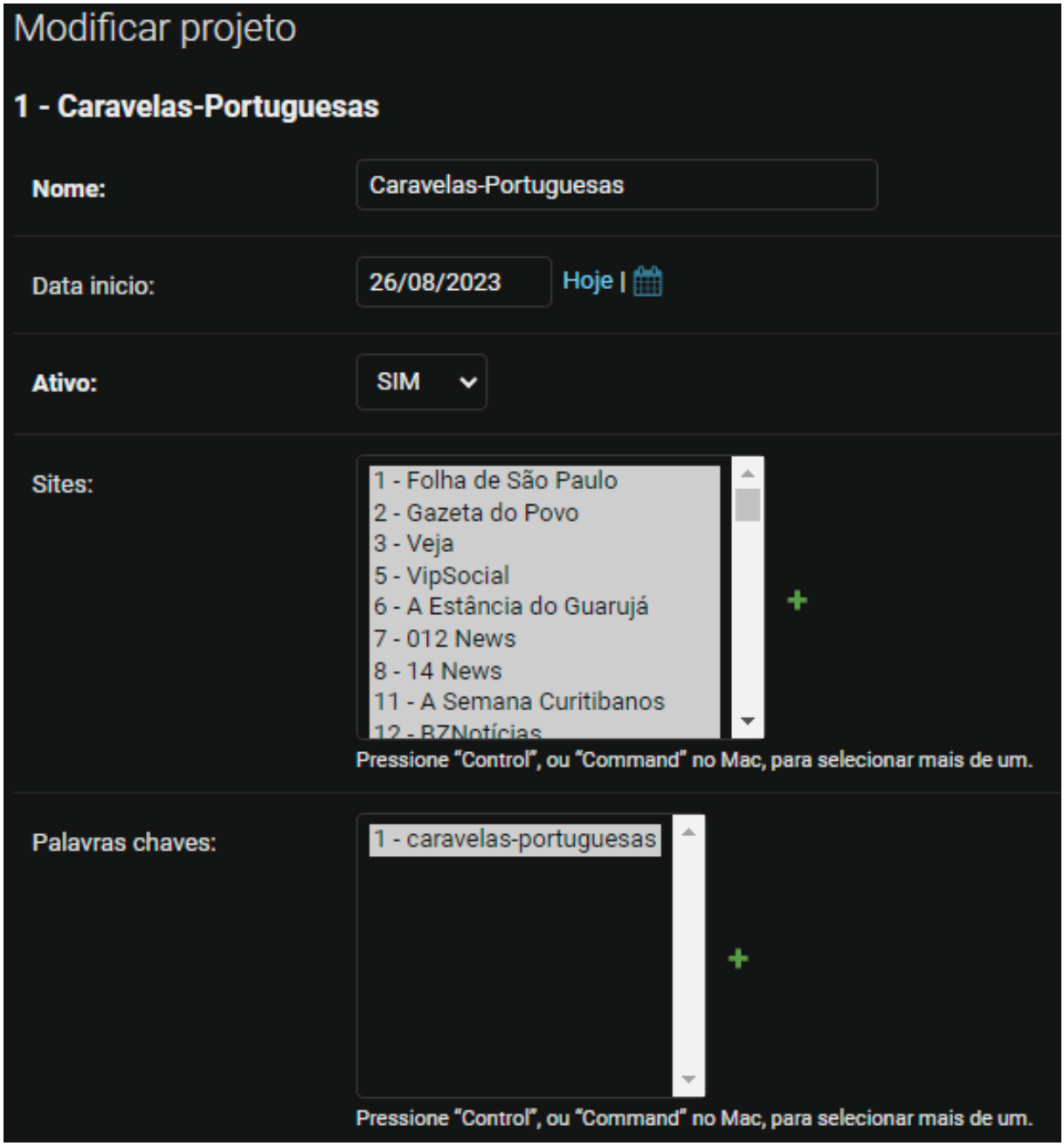


Figura 4.6: Armazenamento de projetos e *strings* e vinculação com os jornais de interesse.

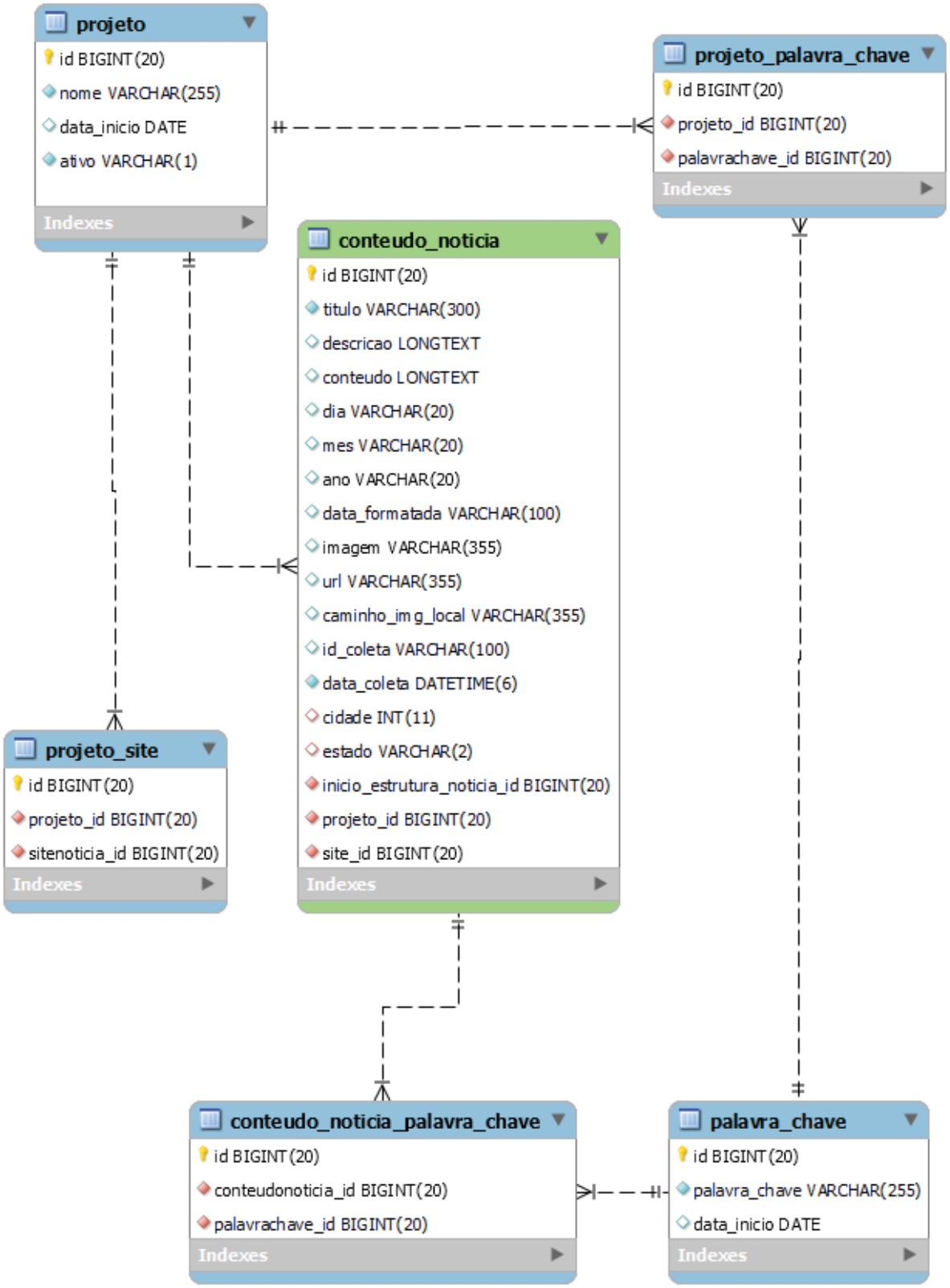


Figura 4.7: Esquema de base de projetos e *strings*.

4.2.3 Base de Coleta

A parte de coleta de notícias é composta por duas tabelas principais. A tabela `conteudo_notícia` é projetada com diversos atributos, incluindo título, descrição, conteúdo, dados temporais (dia, mês e ano), formatação temporal, URL da imagem (que armazena o *link* para a imagem da notícia na *Web*), URL (*link* para a notícia na *Web*), caminho local da imagem (as imagens são mantidas localmente, não no banco de dados), identificador de coleta, data da coleta, informações geográficas (cidade e estado) e identificadores interligados por meio de chaves estrangeiras, abrangendo a estrutura inicial da notícia, o projeto, o jornal, a cidade e o estado. Da mesma forma, a tabela abrange identificadores relativos ao projeto, provenientes da tabela `projeto`, ao jornal vinculado à notícia, derivados da tabela `site_noticia`, bem como a estrutura preliminar da notícia, extraídos da tabela `init_estrutura_noticia`. A tabela `log` armazena um histórico das coletas. Esse histórico abrange URLs, títulos, ocorrências de erro, datas de inserção e também chaves estrangeiras. Essas chaves associam as informações à notícia por meio do identificador proveniente da tabela `conteudo_noticia`, à palavra-chave por intermédio do identificador da tabela `palavra_chave`, ao projeto através do identificador da tabela `projeto` e ao jornal correspondente via o identificador da tabela `site_noticia`. As Figuras 4.8 e 4.9 apresentam uma notícia do jornal A Estância de Guarujá e o armazenamento dessa notícia coletada pelo ENoW, respectivamente. O esquema de base da parte de coleta de notícias é representado na Figura 4.10.

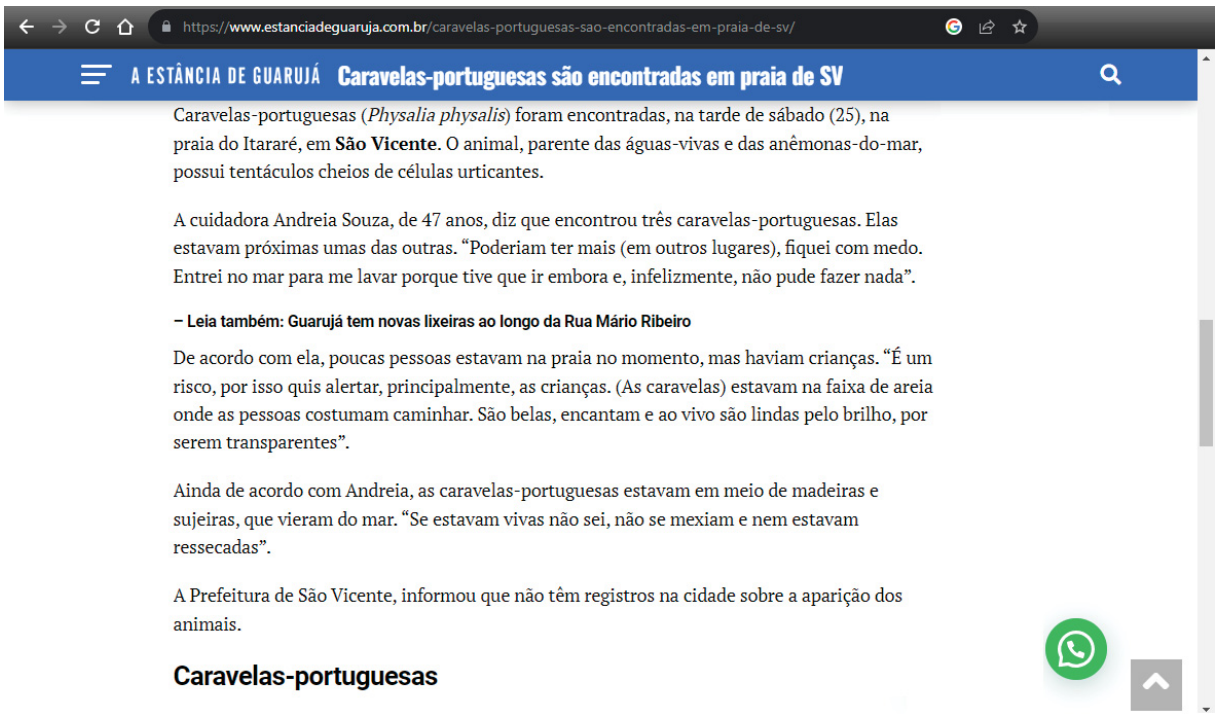


Figura 4.8: Notícia do jornal A Estância de Guarujá.

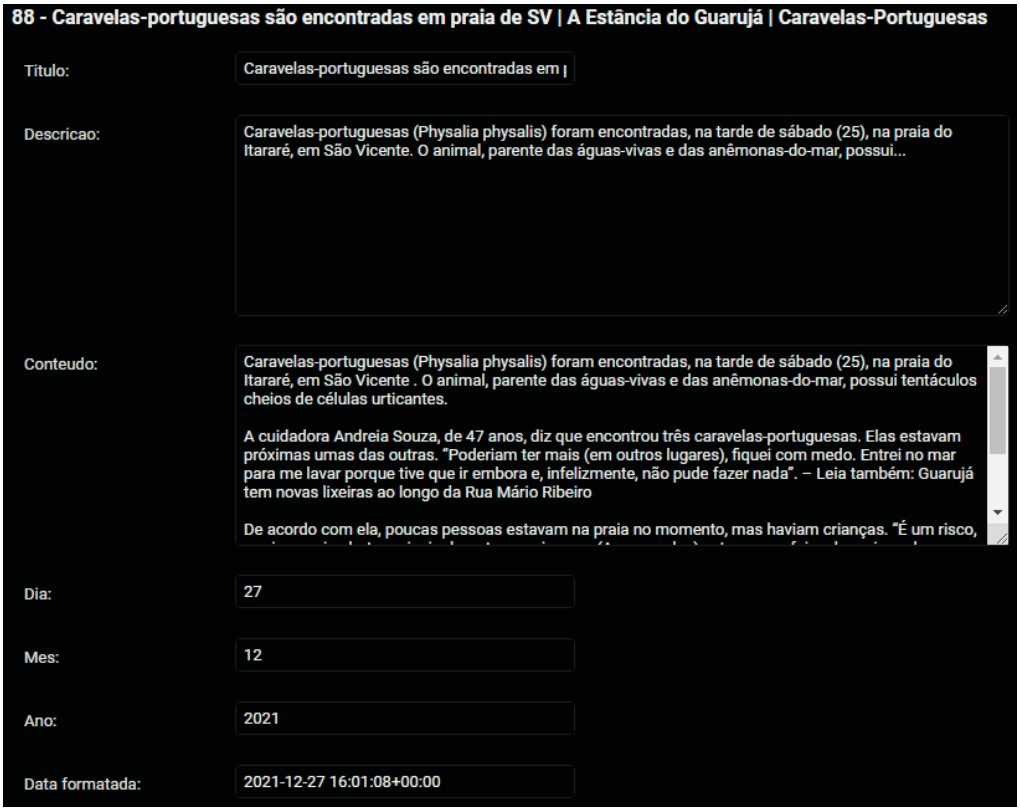


Figura 4.9: Armazenamento da notícia coletada pelo ENoW.

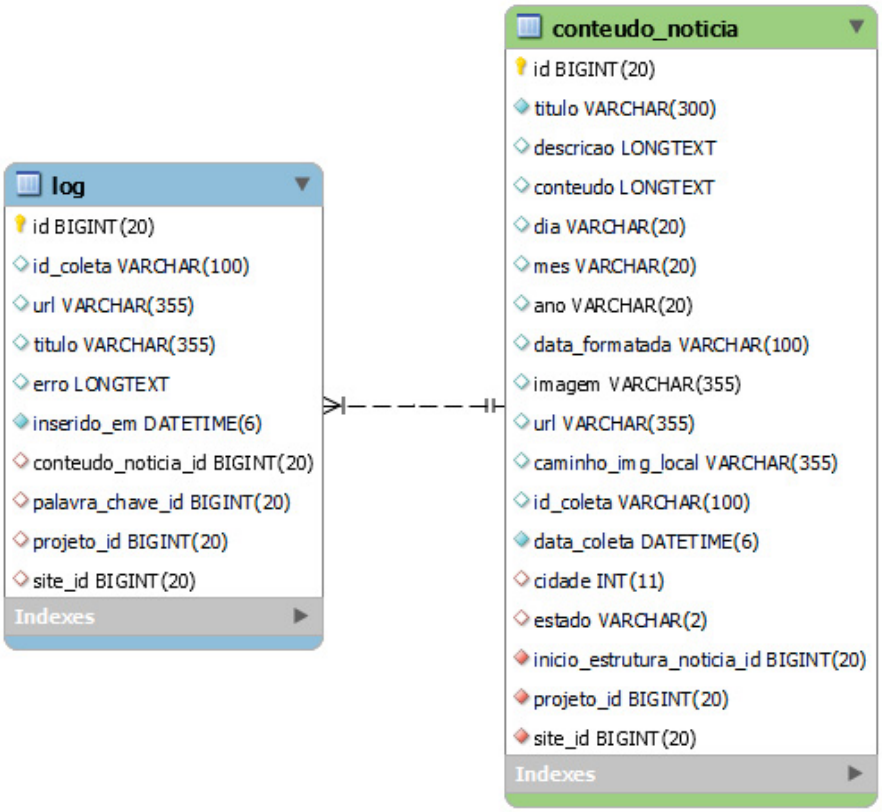


Figura 4.10: Esquema de base de coleta de notícias.

4.2.4 Base de Classificação

A etapa de processamento das notícias coletadas implica o uso de cinco tabelas, das quais três são principais e duas têm caráter associativo. As tabelas principais são denominadas `resultado_processamento`, `noticia_processada` e `classificacao_modelo`. Quanto às tabelas associativas, destacam-se `noticia_referencia_processamento` e `noticia_resultado_processamento`. Na tabela `resultado_processamento`, são registrados um identificador para cada iteração realizada e um identificador para cada processo dentro da iteração. Além disso, esta tabela contém a chave estrangeira do projeto ao qual cada processo e iteração estão vinculados. As métricas de desempenho de cada iteração são igualmente armazenadas na tabela. A tabela `noticia_processada` é responsável por armazenar o conjunto de notícias que o usuário identificou como pertinentes ao finalizar o processo de iterações. Em outras palavras, quando o usuário está satisfeito com os resultados da análise de similaridade de textos, tem a opção de salvar essas notícias na tabela `noticia_processada`. Nessa tabela, são registrados um identificador exclusivo para cada notícia processada, a chave estrangeira referente à tabela `conteudo_noticia`, além da localização e das palavras-chave associadas a cada notícia. A tabela `classificacao_modelo` armazena as métricas e seus desempenhos de cada algoritmo de AM utilizado. Ela também possui uma chave estrangeira para a tabela `resultado_processamento`. A tabela `noticia_resultado_processamento` tem como função armazenar as pontuações de similaridade de textos obtidas a cada iteração realizada. Nesse contexto, a tabela inclui uma chave estrangeira referente à tabela `resultado_processamento`. Ademais, para cada notícia processada durante as iterações, a tabela mantém uma chave estrangeira vinculada à tabela `conteudo_noticia`. Além disso, armazena os rótulos manuais e os rótulos derivados do cálculo da distância de cosseno. Por outro lado, a tabela `noticia_referencia_processamento` retém o identificador da notícia proveniente da tabela `conteudo_noticia` selecionada pelo usuário como referência, distinguindo-a entre relevante e não relevante, e também guarda o identificador do processamento correspondente a cada iteração, proveniente da tabela `resultado_processamento`. Na Figura 4.11 pode-se observar os desempenhos das métricas de classificação das notícias em um processo de iteração. A Figura 4.12 apresenta o esquema da parte de classificação de notícias.

Selecione resultado processamento para modificar — ADICIONAR RESULTADO PROCESSAMENTO +

Ação: ——— Ir 0 de 96 selecionados

ID PROCESSAMENTO	CRIADO EM	PROJETO	ACURACIA	PRECISAO	RECALL	F1 SCORE
1695142952.023194	19 de Setembro de 2023 às 14:03	1 - Caravelas-Portuguesas	0,8925	0,7647	0,3059	0,4370
1695142952.023194	19 de Setembro de 2023 às 14:13	1 - Caravelas-Portuguesas	0,9230	0,8033	0,5765	0,6712
1695142952.023194	19 de Setembro de 2023 às 14:21	1 - Caravelas-Portuguesas	0,9294	0,7971	0,6471	0,7144
1695142952.023194	19 de Setembro de 2023 às 14:34	1 - Caravelas-Portuguesas	0,9422	0,7753	0,8118	0,7932
1695142952.023194	19 de Setembro de 2023 às 14:57	1 - Caravelas-Portuguesas	0,9438	0,7660	0,8471	0,8045
1695142952.023194	23 de Setembro de 2023 às 21:41	1 - Caravelas-Portuguesas	0,9454	0,8072	0,7882	0,7976
1695142952.023194	23 de Setembro de 2023 às 21:46	1 - Caravelas-Portuguesas	0,9535	0,8415	0,8118	0,8264

Figura 4.11: Armazenamento dos desempenhos de um processo de classificação.

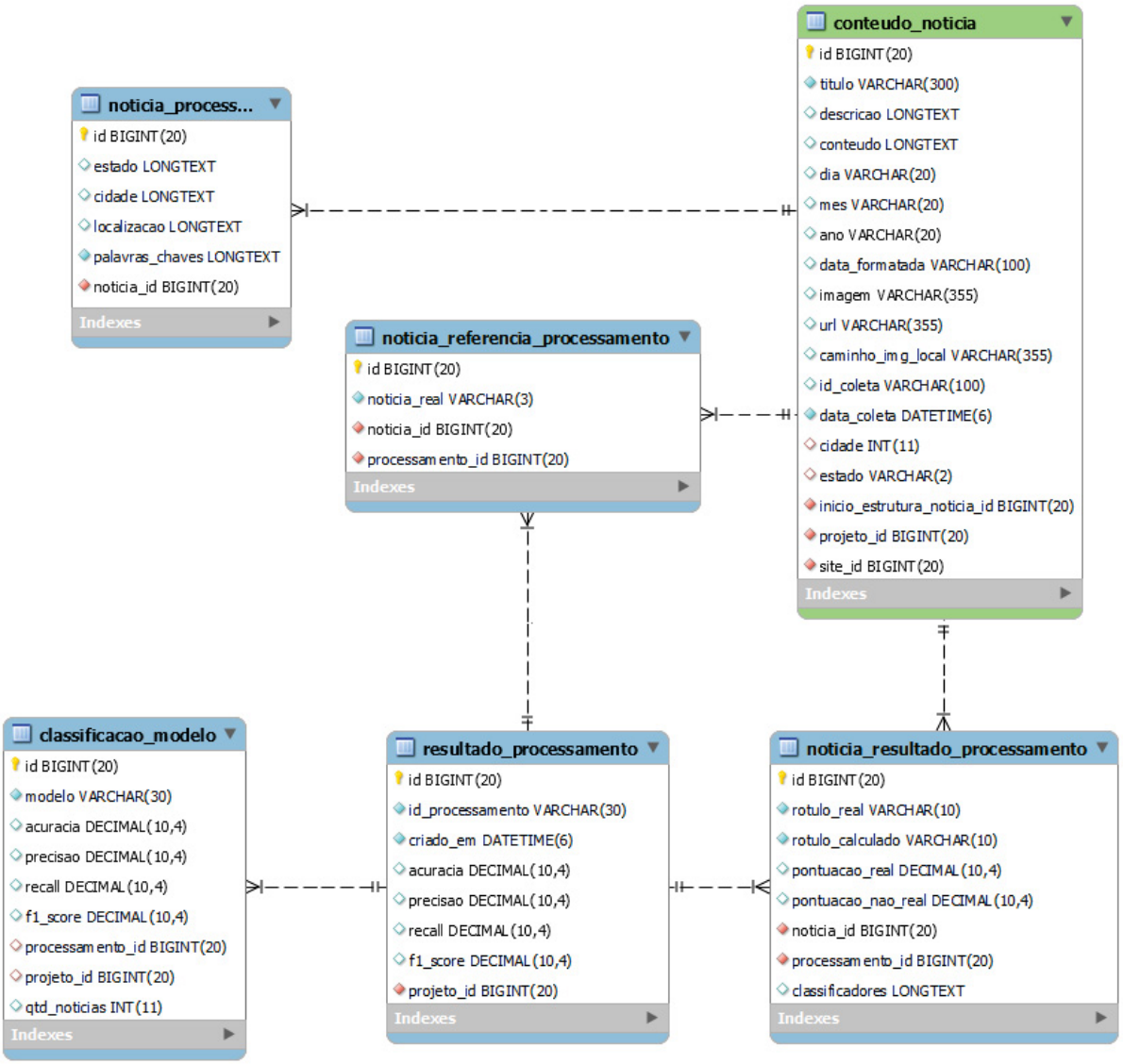


Figura 4.12: Esquema de base de classificação das notícias.

4.3 FUNCIONAMENTO DO ENOW

Para garantir o correto funcionamento do ENoW, destaca-se a relevância da etapa inicial de cadastro de *sites*. Nessa fase, o administrador do sistema registra os metadados das páginas de jornais *online*, incluindo informações inseridas em tabelas que possuem relações com essas páginas, como a estrutura da página e as localizações. Posteriormente, os usuários têm a permissão de inserir um projeto específico juntamente com uma ou mais *strings* de busca. Após os registros realizados, o usuário associa os projetos com uma ou mais *strings* de busca e seleciona as páginas de jornal relevantes para a etapa de coleta. Dessa forma, o processo de coleta é ativado e o ENoW entra em execução. Se forem adicionadas múltiplas *string* de busca a um projeto, estas são empregadas de forma conjunta durante a coleta. Por exemplo, ao adicionar as *strings* "sol"e "praia", durante a coleta, o ENoW coletará todas as notícias encontradas que contenham as palavras sol e/ou praia.

A coleta resulta na criação de tuplas (linhas) na tabela `conteudo_noticia`, para cada notícia coletada, e `log`, para cada coleta realizada. Na **Filtragem com Intervenção**

Humana e com a obtenção dos conteúdos das notícias consolidados, o usuário é convidado a destacar, dentre dez notícias oferecidas, as que possuam relevância para ele. Todas as notícias passam por um pré-processamento com técnicas de PLN e na sequência são transformadas em *word embeddings*, com a técnica TF-IDF. As dez notícias oferecidas ao usuário são representadas por vetores em dois agrupamentos: as selecionadas por ele são relevantes e as não selecionadas, são não relevantes. Esses *embeddings* são usados no processo de similaridade de textos. O processo de similaridade empregado pelo ENoW utiliza a distância de cosseno para calcular as semelhanças entre os *embeddings*. Na sequência, todas as notícias são apresentadas ao usuário, separadas em dois grupos, para que ele analise se estão de acordo com o seu interesse. Caso não sejam, outras dez notícias são oferecidas a ele e o processo é refeito. Essas novas dez notícias são agrupadas às anteriores, separadamente em dois grupos, não havendo perda das já identificadas como relevantes pelo usuário. Dessa forma, o processo torna-se iterativo e finaliza somente quando as notícias estiverem de acordo com o interesse do usuário. Essa classificação é uma alternativa quando não há uma base rotulada por especialistas para classificar com algoritmos classificadores em AM.

Na filtragem com **Classificação Automática**, o ENoW também realiza o processo de classificação de dados de notícias através da aplicação de três diferentes algoritmos de classificação: NB, RF e MLP. Esses algoritmos são supervisionados, o que requer a disponibilidade de uma base de dados rotulada manualmente para o treinamento e validação do modelo. A base de dados contém notícias previamente classificadas, manualmente, como relevantes ou não relevantes em relação à *string* usada no processo de coleta. Para avaliar o desempenho desses algoritmos de classificação, métricas são calculadas, tais como acurácia, precisão, *recall* e *F1-Score*. Essas métricas permitem uma avaliação completa do quão bem cada algoritmo é capaz de fazer previsões corretas, minimizar FPs e FNs, e encontrar um equilíbrio entre precisão e *recall*. O ENoW também utiliza as notícias resultantes da interação com o usuário para o treinamento dos mesmos algoritmos classificadores.

4.4 CONSIDERAÇÕES

Este capítulo apresentou o ENoW, uma ferramenta de extração de dados da *Web* desenvolvida visando armazenar metadados de páginas de jornais *online* e possibilitar a personalização das coletas através da inserção de uma ou mais *string* pelo usuário. Além disso, o ENoW facilita o tratamento desses dados, resultando na criação de uma base de dados filtrada utilizando abordagens de classificação. A primeira abordagem se concentra na categorização de dados em cenários onde não há disponibilidade de uma base rotulada manualmente, por meio de um processo de similaridade de textos usando o cálculo de cosseno. A segunda abordagem emprega modelos de AM alimentados com a base depurada para classificação, utilizando dados rotulados manualmente para o treinamento. Além dessas duas abordagens, o ENoW visa analisar se, através da interação do usuário, escolhendo poucos dados para um treinamento com classificadores em AM, atende o processo de classificação, sem a disponibilidade de uma base de dados rotulada manualmente. Também foi apresentada a arquitetura geral do ENoW, que consiste em três fases principais: registro e coleta, pré-processamento dos dados e processos de filtragem das notícias. O sistema utiliza o modelo relacional para o armazenamento dos dados. A modelagem dos dados abrange partes como informações de páginas de jornais, campos de notícias, estruturas de notícias, projetos, palavras-chave, cidades, estados, entre outros. As tabelas estão organizadas para dar suporte às etapas de cadastro, coleta, pré-processamento e classificação, abrangendo diversos atributos para representar os diferentes aspectos dos dados coletados.

5 IMPLEMENTAÇÃO DO ENoW

A implementação do ENoW é apresentada neste Capítulo. A Seção 5.1 descreve o ambiente de implementação. A Seção 5.2 relata as bibliotecas e ferramentas utilizadas. A Seção 5.3 apresenta os detalhes do desenvolvimento do ENoW. A Seção 5.4 descreve a implementação do PLN e a vetorização dos dados. A Seção 5.5 descreve a implementação no processo de classificação das notícias. A Seção 5.6 descreve as considerações.

5.1 AMBIENTE DE IMPLEMENTAÇÃO

O sistema ENoW foi implementado na linguagem de programação *Python*, por meio do ambiente de desenvolvimento integrado *Visual Studio Code*. O sistema de gerenciamento de banco de dados adotado foi o *MySQL*. Com o propósito de aprimorar tanto a integração quanto a visualização dos dados, o sistema ENoW foi implementado utilizando a interface do framework *Django*.

5.2 BIBLIOTECAS E FERRAMENTAS

Para a implementação do sistema ENoW, foram empregadas diversas bibliotecas. A biblioteca *BeautifulSoup*[1] foi utilizada para a extração de dados de arquivos HTML. Por meio de seu analisador, ela oferece métodos para navegação e pesquisa, proporcionando economia de tempo no desenvolvimento do ENoW. A biblioteca *Newspaper*[2] desempenhou um papel fundamental na extração e análise de artigos de jornal. Essa biblioteca possibilitou a raspagem de dados e a curadoria dos conteúdos, removendo anúncios e elementos irrelevantes presentes nas notícias. O uso da biblioteca *Pandas*[3] proporcionou ferramentas de manipulação e análise de dados, facilitando as etapas de tratamento e exploração dos dados coletados. Para a integração de técnicas de AM, foi empregada a biblioteca *Scikit-Learn*[4], que possibilitou a realização de tarefas como pré-processamento e classificação de dados. Finalmente, a biblioteca *SpaCy*[5] permitiu a realização de PLN, fazendo o REN.

5.3 DESENVOLVIMENTO DO EXTRATOR DE NOTÍCIAS

O código da ferramenta está organizado em classes e métodos que realizam tarefas específicas para realizar a coleta, processamento e armazenamento das notícias. Ele manipula os dados extraídos, formata as datas, processa as informações de notícias, imagens e faz o armazenamento em um banco de dados. No desenvolvimento do ENoW, algumas características foram levadas em conta, como, por exemplo, as diferentes estruturas de paginação dos *sites* de notícias. Alguns *sites* possuem uma paginação na qual apenas é necessário rolar a barra de rolagem, como é o caso da página de notícias do G1. Outras páginas, como a Folha de São Paulo, usam botões com numerações de páginas, mas não apresentam essa numeração em sua URL. Em contrapartida, há páginas que possuem botões para a página seguinte e apresentam essa

[1] https://www.crummy.com/software/BeautifulSoup/bs4/doc/
[2] https://newspaper.readthedocs.io/
[3] https://pandas.pydata.org/
[4] https://scikit-learn.org/stable/
[5] https://spacy.io/

numeração em sua URL, como, por exemplo, a página A Estância de Guarujá. Há também as páginas que não possuem paginação e apresentam as notícias todas numa única página. Para cada uma dessas formas de paginação, uma lógica de implementação foi codificada. Cada tipo de codificação de paginação permite que o código percorra várias páginas de um *site*, coletando informações de notícias conforme as especificações fornecidas.

5.4 IMPLEMENTAÇÃO DE ETAPAS DE PRÉ-PROCESSAMENTO E VETORIZAÇÃO NOS DADOS COLETADOS

O sistema ENoW foi desenvolvido também com o propósito de realizar o pré-processamento dos dados coletados após sua extração dos *sites* de jornais. Nesse contexto, um fluxo de etapas implementadas é seguido para preparar os dados antes de sua análise. Inicialmente, é executado um procedimento de limpeza dos dados, visando eliminar quaisquer informações indesejadas ou irrelevantes. Ou seja, na implementação do processo de limpeza, é definida uma função que recebe os textos das notícias como entrada. A função remove todas as pontuações do texto usando uma expressão regular e transforma o texto em letras minúsculas. Em seguida, a etapa de *tokenização* é aplicada, na qual o texto é dividido em unidades individuais, como palavras. Posteriormente, ocorre a remoção de palavras irrelevantes, conhecidas como *stop words*. Além disso, é implementado o processo de *Lemmatização*, que visa reduzir as palavras às suas formas básicas, a fim de diminuir o vocabulário. Na sequência, é utilizado um vetorizador TF-IDF, que transforma o texto em um formato numérico, representando a importância das palavras em relação à notícia e ao conjunto de notícias. O REN também foi implementado para reconhecer, dentro do texto, localizações e datas. Quando há mais de uma localização, o ENoW gera uma lista e armazena essa lista no devido campo no banco de dados.

5.5 DESENVOLVIMENTO DO PROCESSO DE CLASSIFICAÇÃO DE NOTÍCIAS

Os vetores representativos dos grupos de notícias, definidos pelo usuário como relevantes e não relevantes, e gerados com a técnica TF-IDF foram usados na a implementação do processo de classificação. Cada notícia é transformada em um vetor e os vetores representativos são gerados a partir de um conjunto de vetores. Ou seja, o conjunto de notícias escolhido pelo usuário, é transformado em um vetor representativo (vetor_relevantes) e o conjunto não escolhido é transformado em outro vetor (vetor_nao_relevantes). Cada vetor representativo foi comparado com os vetores correspondentes às demais notícias. Tal comparação foi realizada por meio do cálculo da distância de cosseno, visando avaliar a similaridade entre os textos. Dessa maneira, cada vetor de notícias resultou em duas pontuações: uma associada ao vetor representativo das notícias relevantes (p_rel) e outra referente ao vetor representativo das notícias não relevantes (p_nao_rel). Essas pontuações variam numa escala de zero a um, onde valores próximos a zero indicam vetores menos semelhantes, enquanto valores próximos a um denotam vetores mais semelhantes. A partir dessas pontuações, cada notícia a ser classificada teve suas duas pontuações comparadas. Se a pontuação obtida do cálculo com o vetor representativo das notícias relevantes fosse superior à pontuação oriunda do cálculo com o vetor representativo das notícias não relevantes (p_rel > p_nao_rel), tal notícia é incluída no grupo de notícias consideradas relevantes ao usuário. Por outro lado, caso contrário, a notícia é classificada como não relevante ao usuário. O Algoritmo 1 representa esse processo.

Algoritmo 1 Uso dos Vetores para Classificação

```
1: Inicialização de variáveis
2: variável p_rel
3: variável p_nao_rel
4: for cada noticia a ser classificada do
5:     p_rel = CalcularSimilaridade(vetor_relevantes, vetor_noticia)
6:     p_nao_rel = CalcularSimilaridade(vetor_nao_relevantes, vetor_noticia)
7:         // Notícias de referência são ignoradas
       if p_rel > p_nao_rel then
8:         ClassificarComoRelevante()
9:     else
10:        ClassificarComoNaoRelevante()
11:    end if
12: end for
13: Fim
```

Quando existe uma base de notícias previamente rotulada, o ENoW também usa algoritmos classificadores em AM. Para a geração do modelo de classificação, o conjunto de dados de notícias foi dividido em conjuntos de treinamento, teste e classificação, usando uma proporção de 60%, 10% e 30%, respectivamente. Na sequência, a implementação envolve o equilíbrio das classes, para serem distribuídas nos dados de treinamento. Em seguida, uma lista de modelos de classificação é definida, incluindo os classificadores NB, RF e MLP. No âmbito da condução dos experimentos, vale salientar que os classificadores RF, MLP e NB foram empregados sem a aplicação de ajustes personalizados nos seus parâmetros. Essa abordagem, conhecida como configuração padrão, tem por base o pressuposto de que os valores pré-definidos dos parâmetros fornecidos pelos pacotes de bibliotecas padrão são suficientes para conduzir as tarefas de classificação de forma adequada. Dessa forma, os classificadores foram instanciados e utilizados diretamente sobre a base de dados rotulados, sem intervenções manuais nos parâmetros que regulam seu funcionamento. Ademais, a aplicação do ENoW emprega os vetores representativos das notícias categorizadas como relevantes e não relevantes no treinamento dos algoritmos de AM.

5.6 CONSIDERAÇÕES

Este capítulo descreveu o ambiente de implementação, bibliotecas e ferramentas utilizadas, o desenvolvimento do próprio ENoW, o pré-processamento dos dados coletados e o processo de classificação de notícias. O ENoW foi implementado para manter a proveniência dos dados coletados e possuir extensibilidade em termos de fontes de notícias e projeto, permitindo coletas de diferentes assuntos, conforme a escolha do usuário. A proveniência dos dados abrange a fonte da notícia, quando ela ocorreu e a data da realização da coleta.

6 EXPERIMENTOS

Neste Capítulo são apresentados os resultados de um experimento que engloba o processo de extração das notícias, o subsequente pré-processamento dessas informações, bem como a sua aplicação no processo de classificação. A Seção 6.1 detalha o processo de coleta de dados. A Seção 6.2 descreve a base de dados gerada no experimento. A Seção 6.3 explora o processo de rotulação manual da base de dados. A Seção 6.4 detalha a experimentação com a classificação das notícias usando modelo previamente treinado. A Seção 6.5 apresenta a classificação das notícias com a intervenção humana. Finalmente, a Seção 6.6 analisa e faz considerações dos resultados.

6.1 REALIZAÇÃO DA COLETA DE DADOS COM O ENOW

Os experimentos com o ENoW iniciaram com a inserção de 89 páginas de jornais *online* à base de metadados. Também foi inserido um projeto e uma *string* no banco de dados. O projeto foi denominado como "Caravelas-Portuguesas"e a *string* usada foi "caravelas-portuguesas". A escolha desse projeto para a coleta baseou-se em projetos em andamento na instituição. Posteriormente, uma seleção de 42 fontes de notícias *online* foi associada ao projeto e à referida *string*. Na sequência, realizou-se a coleta das notícias daquelas páginas de jornal que continham a *string* inserida. A Figura 6.1 oferece uma visualização da execução deste procedimento.

Figura 6.1: Execução da coleta.

O processo de coleta de dados foi concluído em um intervalo de tempo de 18 minutos e 13 segundos, durante os quais foram extraídas informações de cada notícia, incluindo título, resumo, data e localização, conteúdo completo e imagem associada. Os dados, referentes às imagens, resultantes dessa coleta, ocupam um volume de armazenamento de 548 MB no disco rígido. Os dados referentes à parte estruturada da base de dados ocupam um volume de armazenamento de 3.59 MB no disco rígido do sistema de gerenciamento de banco de dados. Todo o experimento foi realizado em um computador equipado com um processador Intel(R) Core(TM) i5-8265U, que opera a uma velocidade de *clock* de 1.60GHz. A RAM instalada neste computador é de 8,00 GB e o sistema operacional é um sistema de 64 bits. O processador é baseado em x64, o que é compatível com sistemas operacionais de 64 bits.

6.2 BASE DE DADOS

A fase de extração de dados resultou na construção de uma base composta por 623 notícias. A Figura 6.2 apresenta os títulos de algumas notícias coletadas, enquanto a distribuição da quantidade de notícias por fonte de notícias pode ser visualizada no *dashboard* apresentado na Figura 6.3.

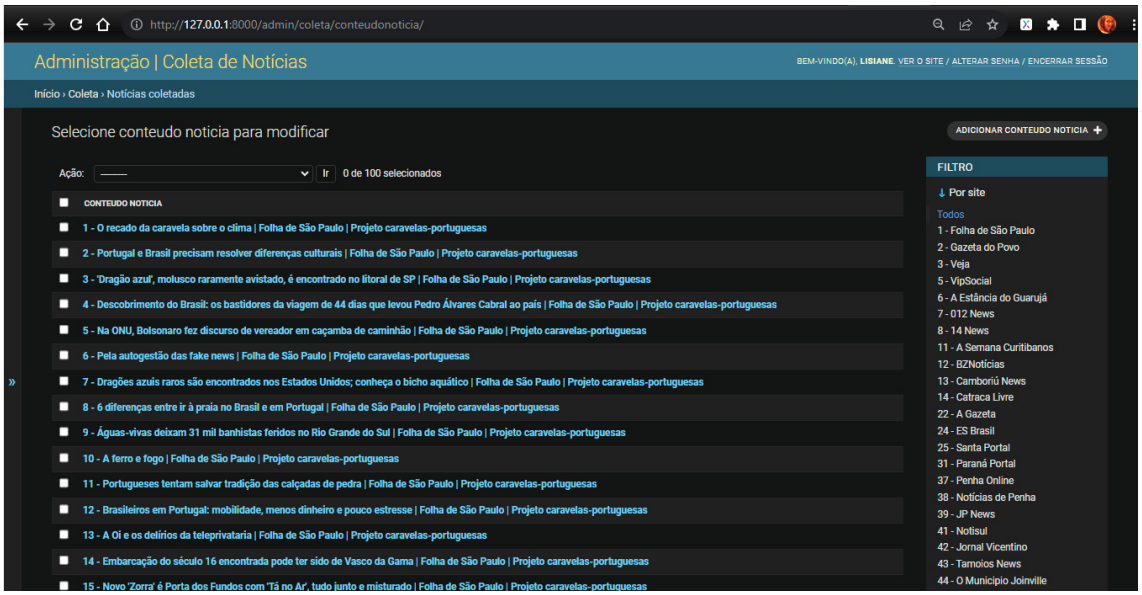


Figura 6.2: Notícias coletadas.

Figura 6.3: Quantidade de notícias coletadas por páginas de notícias.

6.3 ROTULAÇÃO DA BASE DE DADOS

A atribuição de rótulos às notícias foi conduzida de maneira manual, estabelecendo assim a base para a posterior análise e avaliação das diferentes abordagens de classificação dos dados. Os dados receberam rótulos distintos: "Caravelas-portuguesas", que se relacionam com notícias sobre o organismo vivo de nome científico *Physalia physalis*, e "Não caravelas-portuguesas", representando notícias que abordam outros assuntos. Os resultados deste processo de rotulação são refletidos de maneira quantitativa na Tabela 6.1, sendo 85 notícias referentes às caravelas-portuguesas e 538 referentes a outros assuntos. Essa base é reservada para ser aplicada no processo de classificação dos dados.

Rotulação manual	Quantidade de notícias
Caravelas-portuguesas	85
Não caravelas-portuguesas	538
Total	**623**

Tabela 6.1: Rotulação manual dos dados.

6.4 CLASSIFICAÇÃO COM MODELO PREVIAMENTE TREINADO

Nesta Seção são reportados os resultados dos experimentos considerando a existência de uma base previamente rotulada manualmente para gerar um modelo de classificação.

Para este experimento, os dados foram divididos em três conjuntos distintos: 60% para treinamento (374 notícias), 10% para teste (62 notícias) e 30% para classificação (187 notícias). Como a base era desbalanceada, a técnica de *oversampling* foi aplicada. Os algoritmos de AM utilizados foram o NB, RF e MLP. As métricas de desempenho utilizadas foram acurácia, precisão, *recall* e *F1-Score*. A Tabela 6.2 contém as quantidades de VPs, VNs, FPs e FNs. As métricas de desempenho de cada classificador podem ser visualizadas na Tabela 6.3.

Classificador	VP	VN	FP	FN
NB	157	23	2	5
RF	162	20	5	0
MLP	160	23	2	2

Tabela 6.2: VPs, VNs, FPs e FNs dos classificadores.

Classificador	acurácia	precisão	*recall*	*F1-Score*	Tempo de treinamento	Tempo de classificação
NB	0,9626	0,8214	0,9200	0,8679	2400ms	601ms
RF	0,9733	1,0000	0,8000	0,8889	1612ms	387ms
MLP	0,9786	0,9200	0,9200	0,9200	702ms	299ms

Tabela 6.3: Métricas de desempenho de cada classificador.

Observa-se que o classificador MLP demonstrou o desempenho mais eficaz dentre os classificadores avaliados. A acurácia destacou-se em 0,9786. A precisão, de 0,9200, ficou pouco abaixo do classificador RF, o *recall*, de 0,9200, igualou-se ao NB e o *F1-Score*, de 0,9200, foi superior aos classificadores NB e RF. O MLP também apresentou um menor tempo de treinamento e classificação, resultando em 702ms e 299ms, respectivamente.

6.5 CLASSIFICAÇÃO COM INTERVENÇÃO HUMANA APÓS COLETA DE NOTÍCIAS

Após a conclusão da coleta de informações, o ENoW disponibilizou ao usuário dez notícias selecionadas aleatoriamente, permitindo-lhe optar por aquelas de seu interesse, especificamente aquelas relacionadas ao cnidário caravela-portuguesa. No experimento realizado, dentre as dez opções, todas versavam sobre diferentes tópicos. Posteriormente, novas notícias foram carregadas. Quando esse evento ocorre, as dez notícias não escolhidas são automaticamente inseridas no grupo de outros assuntos. Esse carregamento de novas notícias foi realizado quatro

vezes, até que pelo menos uma notícia sobre a caravela-portuguesa fosse incluída nas opções oferecidas.

Assim, para a primeira iteração do processo, os conjuntos de notícias eram compostos por uma notícia sobre a caravela-portuguesa e trinta e nove sobre temas diversos. Ao acionar o botão "Executar processamento", as etapas de PLN são iniciadas. O pré-processamento inclui operações de limpeza, *tokenização*, remoção de *stop words* e lematização. Em seguida, é realizado o processo de vetorização usando TF-IDF. Detalhes das etapas de pré-processamento podem ser visualizados nas Figuras 6.4 e 6.5, onde são aplicadas a um título de notícia e a um texto completo, respectivamente. A Figura 6.6 ilustra o processo de vetorização com TF-IDF.

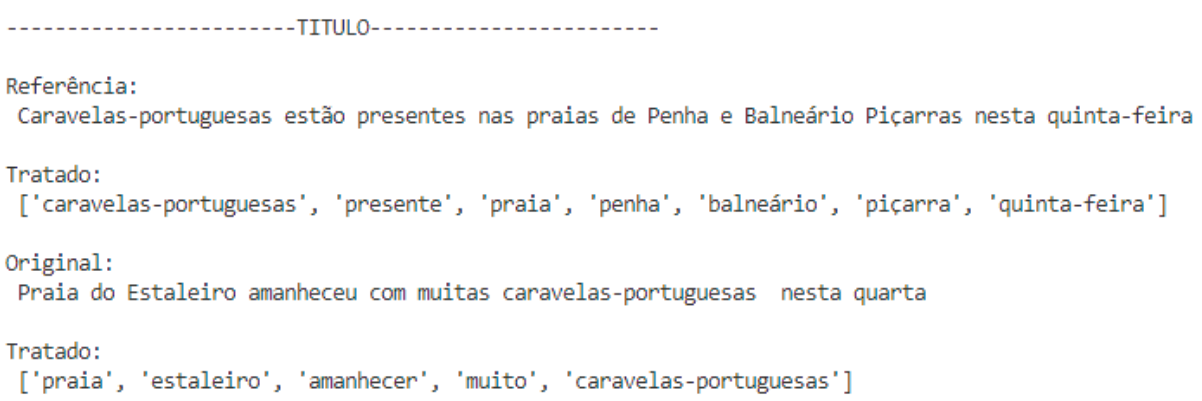


Figura 6.4: Etapas de pré-processamento em um título de notícia.

Figura 6.5: Etapas de pré-processamento em uma notícia completa.

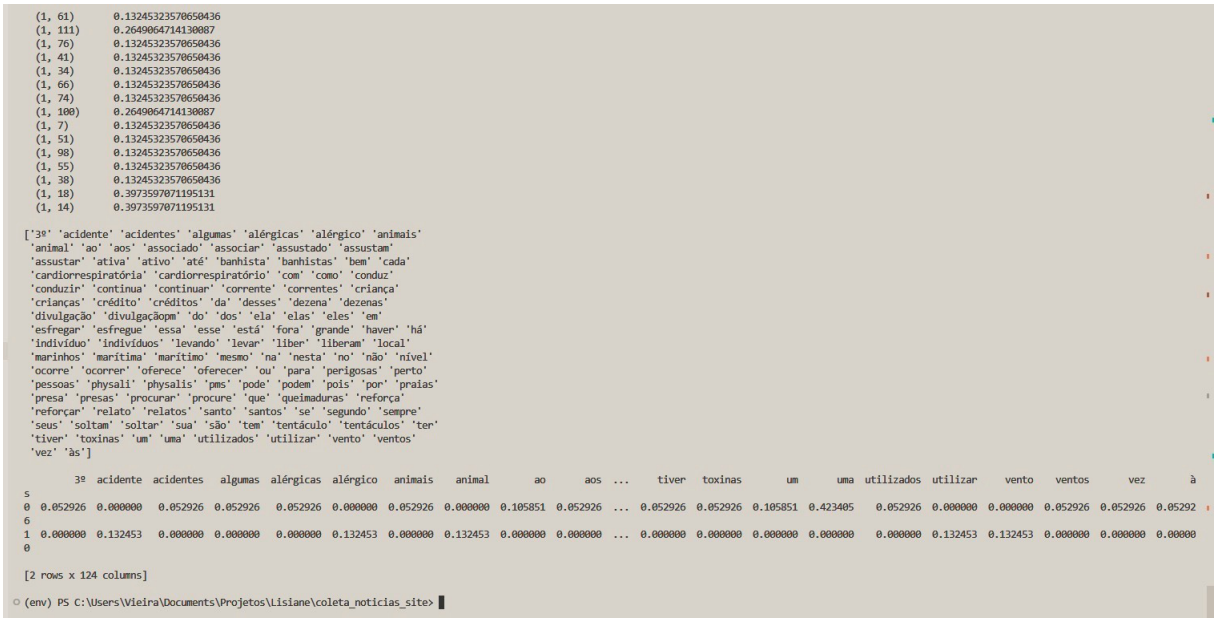


Figura 6.6: Vetorização com TF-IDF em uma das notícias coletadas.

O procedimento de limpeza demandou um minuto e sete segundos. A vetorização foi concluída em cinquenta e cinco segundos. Após o processo de preparação dos dados, foram realizados dois experimentos de classificação. O primeiro envolveu a classificação usando a distância de cosseno. O segundo utilizou algoritmos de AM para fazer a classificação. As próximas Seções apresentam os resultados destes dois experimentos.

6.5.1 Classificação por Distância de Cosseno

A distância de cosseno foi calculada, gerando duas pontuações para cada notícia a ser classificada: uma referente à caravela-portuguesa e outra referente a outros assuntos. O cálculo de distância foi executado em um minuto e quatorze segundos durante a primeira iteração. Em seguida, as métricas de desempenho foram computadas. As notícias são apresentadas ao usuário, assim como os resultados do processo de classificação. Dessa forma, o usuário pode avaliar o quão satisfeito está com a classificação, decidindo se deseja encerrar o processo ou realizar mais iterações. Ao longo do experimento, um total de oito iterações foi realizado. A cada nova iteração, são apresentadas mais dez notícias, seguindo um critério não aleatório. Em outras palavras, para garantir uma seleção equitativa, o ENoW exibe as dez notícias com as pontuações mais elevadas, tanto do conjunto referente às caravelas-portuguesas quanto do conjunto relativo a outros temas. As notícias selecionadas e não selecionadas vão integrando os conjuntos de notícias relacionadas à caravela-portuguesa e de outros assuntos. A Tabela 6.4 exibe os identificadores das notícias que integram os conjuntos referentes às caravelas-portuguesas e a outros assuntos em cada iteração. A Tabela 6.5 apresenta a quantidade de notícias usadas em cada iteração e o tempo de execução de cada iteração. A quantidade de notícias resultantes como VPs, VNs, FPs e FNs em cada iteração podem ser visualizadas na Tabela 6.6. As métricas de desempenho de cada iteração são observadas na Tabela 6.7.

Iteração	IDs escolhidos	IDs não escolhidos
1	117	22, 50, 57, 64, 87, 112, 139, 162, 182, 196, 210, 214, 225, 230, 239, 245, 254, 261, 281, 316, 339, 351, 360, 369, 383, 422, 438, 454, 477, 483, 492, 497, 516, 517, 542, 567, 591, 606, 617
2	1, 3, 9, 18, 21	53, 165, 257, 412, 433
3	23, 24, 25, 26, 66, 77	259, 375, 385, 521
4	80, 88, 90, 91, 92	20, 85, 155, 165, 374
5	93, 94, 95, 96, 97	282, 315, 416, 490, 495
6	118, 206, 333, 367, 623	170, 200, 267, 401, 602
7	120, 205, 311, 328	29, 558, 574, 114, 484, 277
8	119, 124, 123, 194, 514	147, 211, 262, 366, 570

Tabela 6.4: Iterações *human-in-the-loop* no processo de similaridade de textos.

Iteração	Quantidade de IDs escolhidos	Quantidade de IDs não escolhidos	Tempo de processamento
1	1	39	1m14s
2	5	5	1m16s
3	6	4	1m18s
4	5	5	1m21s
5	5	5	1m14s
6	5	5	1m15s
7	4	6	1m19s
8	5	5	1m14s

Tabela 6.5: Quantidade de notícias usadas como referência e tempo de processamento em cada iteração.

Percebe-se que, na primeira iteração, houve uma apresentação significativamente maior de notícias sobre outros temas em comparação às notícias sobre caravelas-portuguesas. Isso decorreu da existência de apenas 85 notícias no banco de dados referentes ao cnidário caravela-portuguesa, em contraste com 538 notícias sobre outros temas. Dado o objetivo do ENoW de considerar todas as notícias oferecidas, sejam sobre caravelas-portuguesas ou não, o conjunto de dados utilizado para os vetores representativos tornou-se desequilibrado. A partir da segunda iteração, há um maior equilíbrio nas notícias oferecidas ao usuário.

Iteração	VP	VN	FP	FN
1	25	491	8	59
2	43	482	12	36
3	43	476	14	30
4	52	465	20	16
5	50	458	22	13
6	51	454	21	7
7	47	449	20	7
8	42	444	20	7

Tabela 6.6: VPs, VNs, FPs e FNs de cada iteração.

Iteração	acurácia	precisão	*recall*	*F1-Score*
1	0,8850	0,7575	0,2976	0,4273
2	0,9162	0,7818	0,5443	0,6417
3	0,9218	0,7543	0,5890	0,6614
4	0,9349	0,7222	0,7647	0,7428
5	0,9355	0,6944	0,7936	0,7406
6	0,9474	0,7083	0,8793	0,7845
7	0,9483	0,7014	0,8703	0,7767
8	0,9473	0,6774	0,8571	0,7567

Tabela 6.7: Métricas de desempenho de cada iteração.

A primeira iteração demonstrou um desempenho bom em termos de acurácia e precisão, mas um desempenho inferior em *recall* e *F1-Score*. A acurácia de 0,8850 indica que 88,50% das notícias foram corretamente classificadas. A precisão de 0,7575 revela que cerca de 75,75% das notícias classificadas como positivas eram verdadeiramente positivas. O *recall* de 0,2976 indica que aproximadamente 29,76% das notícias verdadeiramente positivas foram corretamente classificadas. O baixo *recall* reflete no valor do *F1-Score*, que foi de 0,4273. Na segunda iteração, observa-se um equilíbrio maior nas notícias apresentadas ao usuário, com cinco notícias sobre caravelas-portuguesas e cinco sobre outros temas. Houve uma melhoria em todas as métricas, especialmente no *recall*, indicando uma maior habilidade do classificador em identificar corretamente as notícias verdadeiramente positivas. Nas iterações subsequentes, da terceira à oitava, a acurácia e a precisão mantiveram-se em níveis elevados, indicando que a classificação manteve uma taxa satisfatória de predições corretas e uma proporção significativa de VPs em relação aos FPs. O *recall* variou, mas geralmente permaneceu em um nível moderado, sinalizando a capacidade da classificação em identificar VPs. O *F1-Score* também variou, refletindo a harmonização entre precisão e *recall* em cada iteração. Em resumo, ao longo das iterações, houve uma melhoria global nas métricas de desempenho, indicando que o processo de classificação está sendo refinado e tornando-se mais eficaz em classificar corretamente as notícias de acordo com o interesse do usuário. Essa melhoria é evidente no equilíbrio entre precisão e *recall*, indicando uma melhor capacidade de identificar as notícias relevantes. Considerou-se encerrar as iterações no oitavo *loop* devido ao equilíbrio e à estabilização das métricas de desempenho. É importante mencionar que, a cada iteração, o número de notícias nos grupos representativos aumentou em dez unidades, enquanto o grupo a ser classificado diminuiu em dez unidades, resultando em variação na quantidade de notícias classificadas de uma iteração para outra. O menor tempo de execução registrado foi de 1 minuto e 14 segundos, alcançado nas iterações 1, 5 e 8.

6.5.2 Classificação com AM

Realizou-se uma análise adicional utilizando a abordagem *human-in-the-loop* em conjunto com algoritmos classificadores em AM. Considerando a ausência de uma base rotulada manualmente e a necessidade de interação do usuário, as notícias oferecidas ao usuário foram utilizadas como dados de treinamento: aquelas selecionadas foram incorporadas ao conjunto relacionado às caravelas-portuguesas, enquanto as não selecionadas foram designadas ao conjunto de outros assuntos. Após a aplicação do PLN e da vetorização com TD-IDF, essas notícias foram empregadas para o treinamento dos algoritmos NB, RF e MLP. Neste cenário, foram conduzidas cinco iterações. A Tabela 6.8 apresenta os identificadores das notícias que foram usadas para treinamento em cada iteração, para os três algoritmos classificadores em AM. A Tabela 6.9 destaca a quantidade de notícias por grupos (caravelas-portuguesas e outros assuntos)

usadas para treinamento nos algoritmos classificadores, bem como o tempo de processamento dessas notícias. O menor tempo ocorreu na iteração 3, contando com 1m12s. A Tabela 6.10 apresenta os VPs, VNs, FPs e FNs de cada classificador em cada iteração. Na Tabela 6.11 podem ser visualizadas as métricas de desempenho de cada classificador por iteração. Os resultados detalhados da análise experimental serão expostos na próxima seção.

Iteração	**IDs escolhidos**	**IDs não escolhidos**
1	105, 319	8, 291, 307, 414, 433, 446, 486, 550
2	104, 106, 107, 222, 367	211, 229, 387, 499, 600
3	90, 120, 205, 311, 329	69, 292, 296, 488, 595
4	94, 97, 119, 206, 328	81, 300, 409, 455, 485
5	66, 95, 96, 118, 123	244, 424, 572, 581, 616

Tabela 6.8: Iterações *human-in-the-loop* no processo de classificação com AM.

Iteração	**Quantidade de IDs escolhidos**	**Quantidade de IDs não escolhidos**	**Tempo de processamento**
1	2	8	1m19s
2	5	5	1m17s
3	5	5	1m12s
4	5	5	1m17s
5	5	5	1m15s

Tabela 6.9: Quantidade de notícias usadas para treinamento e tempo de processamento em cada iteração.

Na primeira iteração, as notícias escolhidas estão desequilibradas em relação às notícias não escolhidas. A partir da segunda iteração, percebe-se um maior equilíbrio.

Iteração	**Classificador**	**VP**	**VN**	**FP**	**FN**
1	NB	0	530	0	83
1	RF	2	530	0	81
1	MLP	33	530	0	50
2	NB	51	525	0	27
2	RF	10	525	0	68
2	MLP	51	523	2	27
3	NB	56	518	2	17
3	RF	31	320	0	42
3	MLP	46	517	3	27
4	NB	60	510	5	8
4	RF	39	515	0	29
4	MLP	51	511	4	17
5	NB	56	503	7	7
5	RF	42	509	1	21
5	MLP	48	505	5	15

Tabela 6.10: VPs, VNs, FPs e FNs de cada iteração.

Iteração	Classificador	acurácia	precisão	*recall*	*F1-Score*	Tempo de treinamento	Tempo de classificação
1	NB	0,8646	0,0000	0,0000	0,0000	1307ms	1196ms
1	RF	0,8678	1,0000	0,0240	0,0468	1259ms	1123ms
1	MLP	0,9184	1,0000	0,3975	0,5688	1008ms	976ms
2	NB	0,9552	1,0000	0,6538	0,7906	1299ms	1191ms
2	RF	0,8872	1,0000	0,1282	0,2272	1261ms	1117ms
2	MLP	0,9519	0,9622	0,6538	0,7785	1002ms	974ms
3	NB	0,9679	0,9655	0,7671	0,8549	1311ms	1189ms
3	RF	0,9291	1,0000	0,4246	0,5960	1256ms	1115ms
3	MLP	0,9494	0,9387	0,6301	0,7540	999ms	978ms
4	NB	0,9777	0,9230	0,8823	0,9021	1298ms	1192ms
4	RF	0,9502	1,0000	0,5735	0,7289	1262ms	1119ms
4	MLP	0,9639	0,9272	0,7500	0,8292	1005ms	969ms
5	NB	0,9755	0,8888	0,8888	0,8888	1297ms	1190ms
5	RF	0,9616	0,9767	0,6666	0,7923	1264ms	1117ms
5	MLP	0,9650	0,9056	0,7619	0,8275	999ms	965ms

Tabela 6.11: Métricas de desempenho de cada classificador por iteração.

Percebe-se que o algoritmo NB obteve uma evolução significativa da primeira iteração para a segunda. A acurácia aumentou de 86,46% para 95,52%, indicando uma melhoria na precisão geral do modelo. A precisão aumentou de 0% para 100%, indicando uma transição de não ter acerto para acertar todas as predições positivas. O *recall* aumentou de 0% para 65,38%, indicando que o modelo começou a identificar mais notícias VPs. O *F1-Score* teve um aumento de 0% para 79,06%, mostrando um equilíbrio mais forte entre precisão e *recall*. Na terceira iteração, a acurácia continua aumentando, evidenciando uma melhoria contínua na precisão geral do modelo. Há um aumento na taxa de FPs, evidenciada pela queda da precisão. Porém, nota-se que o modelo está identificando mais VPs, devido ao aumento do *recall*. Também há um equilíbrio mais refinado entre precisão e *recall*, com o aumento do *F1-Score*. Da quarta para a quinta iteração houve uma pequena variação entre todas as métricas: a acurácia teve uma leve diminuição de 97,77% para 97,55%; a precisão teve uma queda um pouco maior, de 92,30% para 88,88%; o *recall* aumentou levemente, de 88,23% para 88,88% e o *F1-Score* foi de 90,21% para 88,88%. Percebeu-se uma certa estabilidade nas métricas acurácia e *recall*, o que demonstrou um início de equilíbrio e constância no modelo.

O algoritmo RF possui um destaque no *recall*, que teve um aumento da primeira para a segunda iteração, de 2,40% para 12,82%, demonstrando uma melhoria significativa na identificação de VPs. O *F1-Score* também aumentou, de 4,68% para 22,72%, mostrando um equilíbrio mais refinado entre precisão e *recall*. Na terceira iteração, o *recall* se mantém evoluindo significativamente, com um aumento de 12,82% para 42,46%. O *F1-Score* passa de 22,72% para 59,60%, o que demonstra uma melhora na harmonização entre precisão e *recall*. Na transição da terceira para a quarta iteração, percebe-se uma evolução de todas as métricas, com exceção da precisão, que se manteve constante, no valor de 100%, da primeira até a quarta iteração. Apenas na quinta iteração que a precisão reduz de 100% para 97,67%.

O algoritmo MLP demonstrou pequenas variações na acurácia, da primeira à quinta iteração. A precisão iniciou em 100%, mas passou a cair para 96,22% na segunda iteração. O *recall* teve um aumento mais considerável, da primeira para a segunda iteração, de 39,75% para

65,38%, mostrando uma melhoria significativa na identificação de VPs. O *F1-Score* também aumentou consideravelmente de 56,88% para 77,85%, refinando o equilíbrio entre precisão e *recall*. Na transição da segunda para a terceira iteração, todas as métricas tiveram quedas. Percebe-se que na última transição, da quarta para a quinta iteração, a acurácia do MLP se manteve praticamente equilibrada, passando de 96,39% para 96,50%. A precisão teve uma leve queda, de 92,72% para 90,56%, indicando uma pequena diminuição na taxa de VPs. O *recall* teve um pequeno aumento de 75% para 76,19% e o *F1-Score* teve uma leve queda de 82,92% para 82,75%, mantendo praticamente o mesmo equilíbrio da iteração anterior.

Em todas as iterações e para todos os classificadores, observa-se uma tendência geral de melhoria na acurácia e nas métricas relacionadas à precisão, *recall* e *F1-Score*. A melhoria geral nas métricas indica que os modelos estão se ajustando e melhorando em sua capacidade de classificar corretamente as notícias.

Entre a quarta e a quinta iteração, observa-se uma estabilidade geral nas métricas de desempenho para os classificadores NB e MLP. Para o classificador RF, há uma pequena diminuição na acurácia, precisão, *recall* e *F1-Score*, indicando uma ligeira redução no desempenho do modelo nessa transição. A estabilidade nas métricas para o NB e o MLP sugere que esses modelos mantiveram um desempenho consistente, enquanto a pequena redução nas métricas do RF indica uma flutuação leve, mas não significativa, no desempenho do modelo entre essas duas iterações. Também constata-se que o classificador NB demonstrou o desempenho mais destacado ao longo das iterações, exceto na primeira iteração, na qual o MLP obteve o desempenho superior. Em relação ao tempo mínimo de treinamento e classificação, percebe-se que o MLP se sobressaiu em todas as iterações.

6.6 CONSIDERAÇÕES

A **Classificação Automática**, utilizando o classificador NB, resultou em uma acurácia de 96,26%, com uma precisão de 82,14%, um *recall* de 92,00% e um *F1-Score* de 86,79%. Utilizando o classificador RF, a acurácia foi de 97,33%, com uma precisão de 100%, um *recall* de 80% e um *F1-Score* de 88,89%. Com o classificador MLP, acurácia alcançou 97,86%, com uma precisão de 92%, um *recall* de 92% e um *F1-Score* de 92%. Percebe-se que na **Classificação Automática**, o melhor modelo para a classificação foi o MLP, considerando a existência de uma base, contendo 623 notícias, previamente treinada. Nota-se que o tamanho da base de treinamento possui impacto no resultado da classificação.

No caso da **Filtragem com Intervenção Humana** empregou-se a classificação baseada na distância de cosseno e a classificação utilizando o AM. A classificação baseada na distância de cosseno, na quinta iteração, resultou em uma acurácia de 93,55%, uma precisão de 69,44%, um *recall* de 79,36% e um *F1-Score* de 74,06%. Por outro lado, a classificação com AM, também na quinta iteração, obteve uma acurácia de 97,55%. A precisão foi de 88,88%, o *recall* foi de 88,88% e o *F1-Score* foi de 88,88% para o classificador NB. Quanto ao RF, a acurácia alcançou 96,16%, com uma precisão de 97,67%, um *recall* de 66,66% e um *F1-Score* de 79,23%. Para o classificador MLP, a acurácia foi de 96,50%, a precisão atingiu 90,56%, o *recall* foi de 76,19% e *F1-Score* foi de 82,75%. A classificação utilizando AM, na abordagem de **Filtragem com Intervenção Humana**, demonstrou um desempenho superior quando comparada à classificação baseada na distância de cosseno. Ou seja, na inexistência de uma base rotulada por humanos, todas as técnicas de AM se mostraram superiores à técnica baseada em similaridade por distância de cosseno. Dentre essas técnicas de AM, o NB superou o MLP em todas as iterações, exceto na primeira. Isso significa que para uma base pequena (de até 50 notícias), esta técnica pode ser uma boa alternativa de classificação.

7 CONSIDERAÇÕES FINAIS

Esta dissertação se baseou na premissa de desenvolver um sistema que não apenas armazenasse metadados de páginas de jornais brasileiros e extraísse informações de notícias, mas também realizasse um pré-processamento desses dados para atender a duas abordagens distintas, embora convergindo para o mesmo objetivo. A primeira abordagem concentrou-se na categorização dos dados das notícias e partiu do princípio de que não havia uma base de dados pré-existente com rótulos manuais, havendo apenas poucas notícias apontadas pelo usuário como relevantes e irrelevantes a ele, para facilitar essa categorização. Nesse sentido, utilizou-se o cálculo do cosseno para procurar apresentar notícias similares à(s) escolhida(s) pelo usuário, por meio do processo *human-in-the-loop*. A segunda abordagem empregou a aplicação de técnicas de AM para realizar a classificação, usando os algoritmos NB, RF e MLP e uma base de dados devidamente rotulada para os processos de treinamento, teste e validação. Além dessas duas abordagens e aproveitando os algoritmos em AM, procedeu-se a uma adicional classificação. Nesta etapa, as notícias apresentadas ao usuário foram utilizadas para o treinamento dos algoritmos NB, RF e MLP. Dessa forma, as notícias escolhidas receberam rótulos de caravelas-portuguesas, enquanto as não selecionadas foram rotuladas como outros assuntos.

O experimento teve início com a coleta de notícias de jornais *online* brasileiros. Esse conjunto de dados obtido foi submetido a um PLN, no qual determinadas notícias foram selecionadas para servir como referência no cálculo da distância de cosseno para avaliar a similaridade de textos com as demais notícias do conjunto. Esse processo foi iterativo, no qual o usuário escolhe as notícias relevantes para ele. Observou-se que a classificação por meio do cálculo da distância de cosseno apresentou desempenho satisfatório em comparação à classificação utilizando uma base de dados previamente rotulada, possibilitando a classificação das notícias mesmo na ausência de uma base de dados com rótulos prévios. Houve uma distinção no tempo de processamento, onde a classificação com algoritmos de AM utilizando a base rotulada foi mais rápida em comparação aos outros processamentos realizados nos demais experimentos.

Além disso, o ENoW teve duas publicações:

- Simpósio Brasileiro de Banco de Dados, no ano de 2022, em Búzios, na sessão de Workshop de Teses e Dissertações em Bancos de Dados (Reips e Hara, 2022).

- Simpósio Brasileiro de Banco de Dados, no ano de 2023, em Belo Horizonte, na sessão de Demos (Reips et al., 2023).

Trabalhos Futuros:

A seguir, são descritos alguns possíveis trabalhos futuros em relação ao ENoW:

- Aprimoramento da extração de dados, explorando a biblioteca *Newspaper* de forma mais abrangente, permitindo a obtenção de informações por meio de consultas mais específicas em vez de uma simples *string* de busca. Isso poderia ser alcançado por meio da integração com mecanismos de indexação, como o *Google*, que facilitaria a coleta de informações de uma variedade de *sites*, exigindo apenas a estrutura do *Google* e uma análise dos *sites* retornados, separando apenas os *sites* de notícias. A partir deste ponto, a biblioteca *Newspaper* gerenciaria todo o processo de coleta;

- Considerar estratégias de gerenciamento de dados para lidar com o aumento de volume de dados à medida que o projeto progride, garantindo que o sistema seja escalável e eficiente;

- Explorar técnicas de deduplicação de dados, especialmente quando há similaridade espacial e temporal entre os registros, a fim de otimizar o armazenamento e melhorar a qualidade dos dados;

- Refinar o processo de categorização de notícias, explorando diferentes técnicas e ajustando hiperparâmetros para alcançar um melhor desempenho. Realizar também um número maior de iterações;

- Investigar o uso de outros algoritmos, além dos já mencionados, de AM para a classificação de notícias, ajustando parâmetros, não usando apenas os parâmetros padrões dos algoritmos, a fim de realizar análises adicionais e comparativas;

- Analisar outras alternativas de vetorização, como o Global Vectors for Word Representation (GloVe), que representa palavras como vetores em um espaço vetorial contínuo, capturando as relações semânticas e sintáticas entre palavras com base na frequência de co-ocorrência das palavras no contexto do conjunto de treinamento;

- Explorar estratégias de ordenamento das dez primeiras notícias apresentadas ao usuário, visando um equilíbrio na seleção, que atualmente está sendo realizada de maneira aleatória. Isso evita que o usuário necessite carregar outras dez notícias devido à falta de interesse na primeira apresentação. Essa exploração implica em usar algoritmos de agrupamento ao oferecer as dez notícias ao usuário. Assim, consegue-se oferecer uma notícia central de cada grupo ao usuário na primeira iteração. A partir da segunda iteração, tornar-se-ia viável usar o algoritmo KNN para o oferecimento das dez próximas notícias;

- Considerar estratégias de aprimoramento da similaridade textual, visando a otimização do tempo de processamento. A ampliação do volume de dados para o cálculo da distância de cosseno resulta em um aumento do tempo de processamento, podendo acarretar desaceleração do sistema;

- Analisar técnicas de similaridade semântica, como o SBERT, por exemplo, que utiliza uma arquitetura neural para gerar representações semânticas das sentenças, permitindo uma medida mais avançada e contextualizada de similaridade;

- Explorar técnicas de *active learning*;

- Realizar a classificação das imagens coletadas, não somente dos textos das notícias;

- Possibilitar ao usuário a alteração do critério de relevância na seleção das dez notícias apresentadas a ele. Este procedimento implica em uma análise prévia das notícias selecionadas: caso estas se diferenciem consideravelmente daquelas escolhidas em iterações anteriores, uma nova interface pode ser exibida ao usuário, permitindo que ele prossiga no processo de similaridade com um novo foco de interesse, sem comprometer os processamentos anteriores. Dessa maneira, o usuário pode ajustar o foco conforme necessário, tendo a flexibilidade de retornar a qualquer momento às iterações do interesse anterior. O sistema não deverá perder a referência na transição de relevância dentro do mesmo processo.

Essas possíveis direções de trabalho futuro visam aprimorar e expandir o ENoW, tornando-o mais eficaz e adaptável às demandas crescentes e às complexidades associadas à categorização e análise de dados jornalísticos.

REFERÊNCIAS

Aizawa, A. (2003). An information-theoretic perspective of tf-idf measures. *Information Processing & Management*, 39(1):45–65.

Alpaydin, E. (2020). *Introduction to Machine Learning*. MIT Press.

Barbosa, A. B. G. e Cavalcanti, A. B. (2020). Web scraping e análise de dados. Em *Anais do V Congresso Nacional de Pesquisa e Ensino em Ciências*, Campina Grande. Realize Editora.

Bentley, F., Quehl, K., Wirfs-Brock, J. e Bica, M. (2019). Understanding online news behaviors. Em *Proceedings of the CHI Conference on Human Factors in Computing Systems*, páginas 1–11, New York. ACM Digital Library.

Berger, B., Waterman, M. S. e Yu, Y. W. (2020). Levenshtein distance, sequence comparison and biological database search. *IEEE Transactions on Information Theory*, 67(6):3287–3294.

Berrar, D. (2018). Bayes' theorem and naive bayes classifier. *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, 403:412.

Chaulagain, B., Shakya, A., Bhatt, B., Newar, D. K. P., Panday, S. P. e Pandey, R. K. (2019). Casualty information extraction and analysis from news. Em *Proceedings of the 16th ISCRAM Conference*, páginas 1002–1011, València, Spain.

da S. Lima, G. (2023). *Classificação de notícias digitais utilizando processamento de linguagem natural*. Trabalho de conclusão de curso, Universidade Federal de Uberlândia, Minas Gerais.

do Nascimento, L. S., Hara, C. S., Junior, M. N. e Noernberg, M. (2022). Redes sociais como uma fonte de dados alternativa para monitorar águas-vivas no brasil. Em *Livro de Memórias do IV SUSTENTARE e VII WIPIS: Workshop internancional de Sustentabilidade, Indicadores e Gestão de Recursos Hídricos (Online) - Even3*, Piracicaba.

dos Santos, R. J. M. (2021). Análise da usabilidade, da experiência do utilizador, e dos impactos percebidos na saúde relacionados com a qualidade de vida com base na mineração de opinião dos utilizadores. Dissertação de Mestrado, Universidade de Coimbra.

Ferrara, E., Meo, P. D., F., G. e Baumgartner, R. (2014). Web data extraction, applications and techniques: A survey. *Knowledge-Based Systems*, 70:301–323.

Fontana, E. (2020). Introdução aos algoritmos de aprendizagem supervisionada. `https://fontana.paginas.ufsc.br/files/2018/03/apostila_ML.pdf`.

Franceschini, R., Rosi, A., Catani, F. e Casagli, N. (2022). Exploring a landslide inventory created by automated web data mining: the case of italy. *Landslides*, 19(4):841–853.

Hardeniya, N., Perkins, J., Chopra, D., Joshi, N. e Mathur, I. (2016). *Natural language processing: python and NLTK*. Packt Publishing.

Johnson, J. A. (2014). From open data to information justice. *Ethics and Information Technology*, 16:263–274.

Jordan, M. I. e Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349:255–260.

Jr, E. C. T., Lim, Z. W. e Ling, R. (2018). Defining “fake news” a typology of scholarly definitions. *Digital journalism*, 6(2):137–153.

Jurafsky, D. (2000). *Speech & language processing*. Pearson Education India.

Kang, Y., Cai, Z., Tan, C., Huang, Q. e Liu, H. (2020). Natural language processing (nlp) in management research: A literature review. *Journal of Management Analytics*, 7(2):139–172.

Lage, L. F. e Cunha, E. L. T. P. (2022). Mudança semântica e word embeddings: estudos de caso na diacronia do português. *Revista de Estudos da Linguagem*, 30(4).

Lawson, R. (2015). *Web scraping with Python*. Packt Publishing Ltd.

Malley, B., Ramazzotti, D. e Wu, J. T. Y. (2019). Data pre-processing. Em *Secondary Analysis of Electronic Health Records.* Springer.

Meesad, P. (2021). Thai fake news detection based on information retrieval, natural language processing and machine learning. *SN Computer Science*, 2(6):425.

Mitchell, R. (2018). *Web scraping with Python: Collecting more data from the modern web*. O’Reilly Media.

Monarch, R. M. (2021). *Human-in-the-Loop Machine Learning: Active learning and annotation for human-centered AI*. Simon and Schuster.

Moro, L. (2019). Treinamento linguístico de "software" na pós-edição de transcrição e tradução automática em cursos de educação a distância. Dissertação de Mestrado, Universidade Estadual de Campinas.

Moser, G. V. B. et al. (2022). *Análise de similaridade entre TF-IDF e modelos contextualizados de linguagem baseados em tokens*. Trabalho de conclusão de curso, Universidade Federal de Santa Catarina, Florianópolis.

Mosqueira-Rey, E., Hernández-Pereira, E., Alonso-Ríos, D., Bobes-Bascarán, J. e Fernández-Leal, A. (2023). Human-in-the-loop machine learning: A state of the art. *Artificial Intelligence Review*, 56(4):3005–3054.

Moura, T. A. D., da S. Costa, L. M., da Silva, K. V. e Rêgo, A. R. (2023). O jornalismo em transformação. *Brazilian Journal of Development*, 9(1):28–44.

Park, E., Park, J. e Hu, M. (2021). Tourism demand forecasting with online news data mining. *Annals of Tourism Research*, 90.

Park, K., Hong, J. S. e Kim, W. (2020). A methodology combining cosine similarity with classifier for text classification. *Applied Artificial Intelligence*, 34(5):396–411.

Raschka, S., Patterson, J. e Nolet, C. (2020). Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *Information*, 11(4):193.

Rauf, I., Troubitsyna, E. e Porres, I. (2019). A systematic mapping study of api usability evaluation methods. *Computer Science Review*, 33:49–68.

Reips, L. e Hara, C. S. (2022). Integração e rotulação automatizada de dados sobre o cnidário physalia physalis, usando a geolocalização como referência. Em *Anais Estendidos do XXXVII Simpósio Brasileiro de Bancos de Dados*, páginas 105–111. SBC.

Reips, L., Musicante, M., Vargas-Solar, G., Pozo, A. T. e Hara, C. S. (2023). Enow-extrator de dados de notícias da web. Em *Anais Estendidos do XXXVIII Simpósio Brasileiro de Bancos de Dados*, páginas 78–83. SBC.

Riaz, M., Shah, G., Asif, M., Shah, A., Adhikari, K. e Abu-Shaheen, A. (2021). Factors associated with hypertension in pakistan: A systematic review and meta-analysis. *PloS One*, 16(1).

Ribeiro, A. L., da C. Araújo, O. R., Oliveira, L. B. e Inácio, M. M. (2020). Processamento de linguagem natural aplicado à classificação de decretos administrativos brasileiros. Em *Anais Estendidos do XXVI Simpósio Brasileiro de Sistemas Multimídia e Web*, páginas 57–60. Sociedade Brasileira de Computação.

Rodrigues, J. M. N. (2020). *Sumarização automática de textos como ferramenta de representação e organização da informação*. Trabalho de conclusão de curso, Universidade Federal do Maranhão.

Santos, F. A., Kobellarz, J. K., de Souza, F. R., Villas, L. A. e Silva, T. H. (2022). Processamento de linguagem natural em textos de mídias sociais: Fundamentos, ferramentas e aplicações. Em *Manuscrito de Capítulo - Sociedade Brasileira de Computação*.

Sarr, E. N., Ousmane, S. e Diallo, A. (2018). Factextract: automatic collection and aggregation of articles and journalistic factual claims from online newspaper. Em *Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, páginas 336–341. IEEE.

Shorten, C. e Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48.

Silva, B. F. (2021). *Utilização de aprendizagem de máquina para classificação de e-mails em categorias relevantes*. Trabalho de conclusão de curso, Universidade Federal de São Carlos, São Carlos.

Sitikhu, P., Pahi, K., Thapa, P. e Shakya, S. (2019). A comparison of semantic similarity methods for maximum human interpretability. Em *2019 artificial intelligence for transforming business and society (AITB)*, volume 1, páginas 1–4. IEEE.

Tang, B. e He, H. (2015). Kerneladasyn: Kernel based adaptive synthetic data generation for imbalanced learning. Em *2015 IEEE congress on evolutionary computation (CEC)*, páginas 664–671. IEEE.

Tereso, M. (2019). Análise de sentimento a companhias aéreas norte americanas. *ISLA Multidisciplinary e-Journal*, 2(1):52–65.

Upadhyay, S., Pant, V., Bhasin, S. e Pattanshetti, M. K. (2017). Articulating the construction of a web scraper for massive data extraction. Em *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, páginas 1–4, Coimbatore, India. IEEE.

Vargas-Solar, G., Zechinelli-Martini, J., Espinosa-Oviedo, J. A. e Vilches-Blázquez, L. M. (2021). Laclichev: Exploring the history of climate change in latin america within newspapers digital collections. Em *European Conference on Advances in Databases and Information Systems*, páginas 121–132. Springer.

Wang, B. e Kuo, C. J. (2020). Sbert-wk: A sentence embedding method by dissecting bert-based word models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2146–2157.

Wu, X., Xiao, L., Sun, Y., Zhang, J., Ma, T. e He, L. (2022). A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems*, 135:364–381.

Xavier, J. (2019). *O uso da análise de sentimentos no Twitter para avaliar a opinião do público consumidor a respeito do sistema operacional mobile Android 10*. Trabalho de conclusão de curso, Antonio Meneghetti Faculdade - AMF, Restinga Seca.

Zucchi, L. E. A. e dos Reis, J. C. (2021). *Sistema automatizado de questão e respostas em e-commerce baseado em similaridade de sentenças multilíngues*. Projeto final de graduação, Universidade Estadual de Campinas.