UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS VII - PATOS
CENTRO DE CIÊNCIAS EXATAS E SOCIAIS APLICADAS
CURSO DE GRADUAÇÃO EM BACHARELADO EM COMPUTAÇÃO

JAIRO SOARES DE LIMA

SISTEMA DE RECOMENDAÇÃO PARA TRABALHOS CIENTÍFICOS PRODUZIDOS PELA UEPB

PATOS - PB
2024

JAIRO SOARES DE LIMA

SISTEMA DE RECOMENDAÇÃO PARA TRABALHOS CIENTÍFICOS PRODUZIDOS PELA UEPB

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Computação do Centro de Ciências Exatas e Sociais Aplicadas da Universidade Estadual da Paraíba, como requisito à obtenção do título de bacharel em Computação.

Orientador: Dr. Demetrio Gomes Mestre

PATOS - PB
2024

L732s Lima, Jairo Soares de.
Sistema de recomendação para Trabalhos Científicos produzidos pela UEPB [manuscrito] / Jairo Soares de Lima. - 2024.
45 p. : il. colorido.

Digitado.
Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências Exatas e Sociais Aplicadas, 2024.
"Orientação : Prof. Dr. Demetrio Gomes Mestre, Coordenação do Curso de Computação - CCEA. "

1. Processamento de Linguagem Natural. 2. Aprendizado de Máquina. 3. Filtragem Baseada em Conteúdo. I. Título

21. ed. CDD 005.3

Elaborada por Camile de A. Gomes - CRB - 15/559 COBIB/UEPB

JAIRO SOARES DE LIMA

SISTEMA DE RECOMENDAÇÃO PARA TRABALHOS CIENTÍFICOS PRODUZIDOS PELA UEPB

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Computação do Centro de Ciências Exatas e Sociais Aplicadas da Universidade Estadual da Paraíba, como requisito à obtenção do título de bacharel em Computação.

Trabalho aprovado em 05/06/2024.

BANCA EXAMINADORA

Demetrio Gomes Mestre

Prof. Dr. Demetrio Gomes Mestre
(Orientador)

Janderson Jason Barbosa Aguiar

Prof. Dr. Janderson Jason Barbosa Aguiar
(Examinador)

Jannayna Domingues Barros Filgueira

Prof. Dra. Jannayna Domingues Barros Filgueira
(Examinadora)

AGRADECIMENTOS

Em primeiro lugar, gostaria de expressar minha gratidão a Deus por me abençoar com saúde e sabedoria ao longo desta jornada.

Gostaria também de estender minha sincera apreciação à minha esposa, Géssica Martins Rufino, pelo seu amor, apoio e compreensão incondicionais. Seu incentivo e paciência têm sido uma fonte constante de força, e sou verdadeiramente abençoado por tê-la ao meu lado.

À minha família, especialmente à minha mãe, Leiliane, meu pai, Janicleudo, e meu irmão, Miguel, sou eternamente grato pelo seu amor, incentivo e sacrifícios sem fim. Suas crenças inabaláveis em mim tem sido a força motriz por trás do meu sucesso.

Por fim, gostaria de expressar minha sincera gratidão ao meu orientador, Dr. Demetrio Gomes Mestre, por sua orientação, expertise e incentivo inestimáveis ao longo desta jornada. Sua dedicação à excelência e apoio inabalável foram fundamentais para moldar meu crescimento acadêmico e profissional.

A todos aqueles que me apoiaram e encorajaram ao longo do caminho, obrigado do fundo do meu coração. Suas contribuições foram inestimáveis, e sou profundamente grato por seu apoio e incentivo.

“Se eu vi mais longe,
foi por estar sobre ombros de gigantes."
Isaac Newton

RESUMO

Com o crescimento exponencial de dados na internet, encontrar informações relevantes tornou-se um desafio cada vez maior. Essa problemática se estende ao ambiente acadêmico, na qual a quantidade de artigos gerados em conferências e simpósios é vasta. Diante desse cenário, este estudo propõe o desenvolvimento e avaliação de um sistema de recomendação de trabalhos acadêmicos. Utilizando o repositório online da UEPB como base, o trabalho combina técnicas de Processamento de Linguagem Natural, Aprendizado de Máquina e Filtragem Baseada em Conteúdo para aprimorar a descoberta de artigos relevantes para estudantes e pesquisadores. O procedimento envolveu a coleta de dados por meio de web scraping, seguida de pré-processamento, incluindo a remoção de stopwords e lematização. Além disso, o estudo analisou diversas técnicas, como similaridade de Cossenos, similaridade de Jaccard, o modelo BERT, BM25 e ChatGPT. Essas técnicas foram aplicadas e avaliadas com o propósito de identificar a combinação mais eficaz de métodos e algoritmos.

Palavras-chave: Sistema de recomendação; Processamento de Linguagem Natural; Aprendizado de Máquina; Filtragem Baseada em Conteúdo.

ABSTRACT

With the exponential growth of data on the internet, finding relevant information has become an increasingly significant challenge. This issue extends to the academic environment, where the volume of articles generated in conferences and symposiums is vast. In light of this scenario, this study proposes the development and evaluation of a recommendation system for academic papers. Using the UEPB online repository as a basis, the work combines techniques from Natural Language Processing, Machine Learning, and Content-Based Filtering to enhance the discovery of relevant articles for students and researchers. The procedure involved data collection through web scraping, followed by preprocessing steps such as stopword removal and lemmatization. Additionally, the study analyzed various techniques, including Cosine Similarity, Jaccard Similarity, the BERT model, BM25, and ChatGPT. These techniques were applied and evaluated to identify the most effective combination of methods and algorithms.

Keywords: Recommendation System; Natural Language Processing; Machine Learning; Content-Based Filtering.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

BERT	*Bidirectional Encoder Representations from Transformers*
BM25	*Best Matching 25*
FBC	*Filtragem Baseada em Conteúdo*
GPT	*Transformador pré-treinado generativo*
ML	*Machine Learning*
NLP	*Natural Language Processing*
SR	*Sistema de Recomendação*
TF-IDF	*Term Frequency-Inverse Document Frequency*
UEPB	*Universidade Estadual da Paraíba*

SUMÁRIO

1 INTRODUÇÃO

No cenário contemporâneo, o mundo está imerso em uma era de transformações aceleradas, impulsionadas pelo avanço exponencial das tecnologias de informação e comunicação. O advento da internet, em particular, revolucionou não apenas a maneira como interagimos com o mundo, mas também redefiniu a própria essência da sociedade. Compartilhamento instantâneo de informações, comunicações globais em tempo real e o livre acesso ao conhecimento tornaram-se elementos intrínsecos ao nosso cotidiano.

Segundo a Cisco, de acordo com seu relatório anual sobre a internet, o número de usuários conectados à rede em 2018 era de aproximadamente 3,9 bilhões. No entanto, em 2023, esse número já ultrapassou 5,3 bilhões de usuários (Cisco, 2023). Com esse aumento significativo no volume de usuários, também acompanha um aumento no fluxo de dados na internet.

O gigantesco volume de dados trazido pela internet criou um ecossistema rico em informações, com bilhões de pessoas, dispositivos e sistemas gerando, compartilhando e armazenando dados de maneira contínua. Esse imenso volume de informações trouxe consigo desafios substanciais, sendo um deles a seguinte questão: como podemos extrair e transformar esses dados em informações úteis? (Machado, 2018).

Em resposta a esse desafio, os sistemas de recomendação (SR) surgem como uma solução engenhosa e altamente relevante. Em um mundo saturado de informações, a busca por produtos, filmes, músicas ou conteúdo de interesse pessoal pode ser comparada a encontrar uma agulha em um vasto palheiro virtual. Grandes empresas, como Amazon, Netflix e Google, adotam esses sistemas de recomendação para aprimorar a experiência do usuário. Eles se tornaram aliados tanto para as empresas quanto para os usuários, simplificando a busca e ajudando os usuários a encontrar exatamente o que desejam de maneira intuitiva (Aggarwal, 2016).

Na Netflix, em particular, os usuários têm a oportunidade de avaliar filmes e, com base nessas avaliações, são sugeridos outros filmes, levando em consideração as preferências de outros usuários que avaliaram o mesmo filme. Já na Amazon, ao adquirirmos um produto, frequentemente nos deparamos com a seguinte recomendação: "Pessoas que compraram este item também compraram os seguintes itens". Este é mais um exemplo de como os sistemas de recomendação são aplicados para identificar e auxiliar os usuários a encontrar exatamente o que estão buscando (Aggarwal, 2016).

Como destacado anteriormente, os sistemas de recomendação desempenham um papel essencial, sendo cuidadosamente desenvolvidos para discernir as preferências individuais de cada usuário, compreendendo seus gostos e necessidades exclusivas. No contexto acadêmico, na qual a busca por artigos, dados e recursos relevantes desempenha um papel crucial na qualidade do trabalho acadêmico, os sistemas de recomendação emergem como aliados inestimáveis para pesquisadores.

Os pesquisadores enfrentam o desafio de lidar com um vasto volume de conhecimento científico publicado. Diante disso, encontrar os melhores trabalhos para respaldar suas ideias

se torna uma tarefa árdua, pois, mesmo com essa abundância de informações, a maioria não atende às suas necessidades específicas. Isso leva à chamada 'sobrecarga de informações', na qual apenas uma pequena parte é realmente relevante (Sugiyama; Kan, 2013).

A demanda é crescente por estudos que tenha como foco a criação de ferramentas eficazes e acessíveis para recomendar trabalhos acadêmicos, levando em consideração as necessidades individuais dos pesquisadores. Nesse cenário, os sistemas de recomendação se destacam como soluções promissoras para alcançar esse objetivo.

1.1 Objetivo Geral

Desenvolver e avaliar um sistema de recomendação de trabalhos acadêmicos que combina técnicas de Processamento de Linguagem Natural (PLN), Aprendizado de Máquina (ML) e Filtragem Baseada em Conteúdo (FBC).

1.2 Objetivos Específicos

Para alcançar o objetivo geral deste trabalho, foram estabelecidos os seguintes objetivos específicos:

- **Fundamentação Teórica:** Realizar uma revisão bibliográfica abrangente dos conceitos e modelos de sistemas de recomendação, destacando os principais métodos e abordagens utilizados em trabalhos acadêmicos semelhantes.
- **Coleta de Dados:** Utilizar técnicas de *web scraping*[1] para coletar metadados detalhados, como títulos, resumos, autores e datas de publicação, entre outros dados dos trabalhos acadêmicos disponíveis no repositório da Universidade Estadual da Paraíba (UEPB).
- **Análise Exploratória:** Realizar uma análise exploratória dos dados coletados para identificar tendências, formato dos dados e possíveis padrões, incluindo a análise de frequência de palavras-chave e a criação de visualizações informativas.
- **Algoritmos de Recomendação:** Utilizar os principais algoritmos de recomendação que leve em consideração os tópicos identificados na análise exploratória, utilizando ML e FBC.
- **Avaliação dos Resultados:** Avaliar a relevância e a coesão das recomendações fornecidas pelo sistema em cada algoritmo.

1.3 Justificativa

Com a expansão da internet, as universidades adotaram a prática de compartilhar seus trabalhos acadêmicos em formato digital, tornando-os acessíveis ao público em geral. Entretanto,

[1] Técnica usada para coletar dados não estruturados visando convertê-los em dados estruturados.

essa disseminação resultou em uma produsão de trabalhos que abordam uma ampla diversidade de tópicos, tornando a identificação dos trabalhos mais relevantes uma tarefa desafiadora.

Os pesquisadores frequentemente enfrentam esse desafio na busca por artigos que sejam pertinentes às suas pesquisas. Devido à vasta quantidade de publicações em diferentes periódicos, eles são obrigados a investir um tempo considerável na tarefa de identificar os trabalhos mais adequados para seus projetos (Lee; Lee; Kim, 2013). Essa sobrecarga de informações representa um obstáculo significativo para os pesquisadores, afetando diretamente a eficiência e a qualidade de suas pesquisas acadêmicas. Portanto, é essencial explorar abordagens inovadoras, como sistemas de recomendação, para otimizar a identificação de trabalhos acadêmicos relevantes e promover avanços significativos no campo da pesquisa científica.

A Universidade Estadual da Paraíba (UEPB), como muitas outras instituições de ensino superior, possui um repositório digital extenso, composto por uma diversidade de trabalhos acadêmicos em diversas áreas do conhecimento. Com mais de 28 mil trabalhos disponíveis para consulta atualmente. Nesse contexto, a presente pesquisa visa desenvolver e avaliar um sistema de recomendação de artigos científicos, utilizando técnicas de processamento de linguagem natural, aprendizado de máquina e filtragem baseada em conteúdo.

Portanto, diante da crescente demanda por soluções que auxiliem os pesquisadores na seleção de artigos acadêmicos de forma eficiente e personalizada, esta pesquisa busca preencher uma lacuna importante no cenário acadêmico, contribuindo para a otimização da busca por conhecimento e aprimorando a qualidade das pesquisas realizadas na UEPB.

1.4 Metodologia

Dado que os principais propósitos deste trabalho são a coleta de dados do repositório digital da UEPB, a aplicação dos algoritmos de recomendação e a análise dos resultados obtidos, as etapas do trabalho foram organizadas da seguinte forma:

- **Coleta de dados**: Nesta seção, além da coleta de dados, realizaremos o pré-processamento dos dados e uma análise exploratória.
- **Algoritmos de Recomendação**: Definir os algoritmos e técnicas de recomendação utilizados na análise.
- **Implementação e Resultados**: Após a implementação de cada algoritmo, procedemos com as recomendações para um tópico específico, a fim de avaliar o desempenho.
- **Avaliação dos Resultados**: Avaliamos e comparamos os resultados obtidos por cada algoritmo, identificando possíveis áreas de aprimoramento.

1.5 Estrutura do trabalho

Este trabalho está organizado em cinco capítulos, distribuídos da seguinte forma: no Capítulo 1, apresenta-se uma visão geral do trabalho, abordando a contextualização do problema, os objetivos, a justificativa e a estrutura do estudo; no Capítulo 2, são expostos os conceitos e os trabalhos relacionados a este trabalho de conclusão de curso; no Capítulo 3, detalha-se a estratégia de coleta de dados e os algoritmos de recomendação selecionados; no Capítulo 4, são demonstrados os resultados obtidos; e, no Capítulo 5, são apresentadas as considerações finais. Por fim, as referências utilizadas no trabalho estão listadas ao final do documento.

2 REFERENCIAL TEÓRICO

Neste capítulo estão presentes os tópicos necessários para o entendimento deste trabalho.

2.1 Processamento de Linguagem Natural

O processamento de linguagem natural, ou também denominado Natural Language Processing (NLP), é uma subdivisão da computação e linguística com o propósito de desenvolver modelos computacionais e recursos linguísticos para automatizar o processamento de linguagens humanas (Mcshane; Nirenburg, 2021). Considerado um campo interdisciplinar focado no processamento de linguagem oral e escrita produzida por seres humanos (Caseli; Freitas; Viola, 2022).

O NLP enfrenta desafios significativos devido à notável variabilidade inerente às línguas humanas. A natureza complexa, aberta e dinâmica das línguas resulta em imprecisões e ambiguidades que podem dificultar a interpretação precisa. A presença de limites difusos e a riqueza de significados contextuais geram obstáculos para a análise e compreensão automáticas, exigindo abordagens flexíveis e adaptáveis (Caseli; Freitas; Viola, 2022).

O NPL desempenha um papel vital em várias aplicações conhecidas, como o Google Tradutor, assistentes virtuais, como a Siri da Apple e a Alexa da Amazon, que conseguem interpretar comandos de voz dos usuários e executar uma diversidade de tarefas de maneira intuitiva. Os chatbots também assumem uma função essencial no suporte ao cliente e em diferentes interações automatizadas (Alcarde, 2023).

Antes de trabalhar com NPL, é crucial realizar um pré-processamento dos dados a fim de otimizar e extrair as melhores características. Dentre as técnicas essenciais relatadas por Caseli, Freitas e Viola (2022), o pré-processamento envolve etapas como:

- **Tokenização:** Os tokens são unidades essenciais obtidas através da tokenização, processo fundamental que transforma sequências de caracteres em unidades significativas para o processamento. Essa etapa é realizada por ferramentas como as expressões regulares, que identificam padrões específicos e extraem trechos relevantes. A tokenização é vital no processamento de linguagem natural, permitindo a compreensão e tratamento adequado de palavras e termos nos textos analisados.

- **Lematização:** É um processo de normalização linguística que reduz as palavras flexionadas para sua forma base ou canônica, conhecida como lema. Isso é crucial em muitas aplicações de PLN, pois simplifica o vocabulário e permite uma análise mais precisa, evitando a redundância de termos derivados que possuem o mesmo significado fundamental.

- **Remoção de *StopWords*:** *Stopwords* são termos comuns, como preposições e conjunções, removidos durante o processamento de linguagem natural devido à sua contribuição limitada para o significado do texto. Sua exclusão otimiza a eficiência computacional e

a precisão das análises, permitindo que os algoritmos se concentrem em palavras-chave mais relevantes.

2.1.1 Word Embeddings ou Vetores de Palavras

As *embeddings* são representações vetoriais estáticas nas quais cada palavra é mapeada como um ponto em um espaço de *n* dimensões. Por exemplo, embeddings de 300 dimensões empregam 300 valores reais para representar uma palavra específica. Essas representações podem ser adquiridas a partir de um corpus, seja por contagem de frequência, como no *GloVe*, ou por meio de modelos neurais, como o *Word2Vec*. Através dessas representações vetoriais, é possível identificar similaridades sintáticas e semânticas, as quais são inferidas com base no contexto de ocorrência das palavras no corpus utilizado durante o treinamento (Caseli; Freitas; Viola, 2022).

2.2 Sistema de Recomendação

O primeiro trabalho registrado sobre um sistema de recomendação (SR) foi publicado em 1994 por Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom e John Riedl, com o título *"GroupLens: an open architecture for collaborative filtering of netnews"*. Esse estudo pioneiro introduziu a abordagem de filtragem colaborativa, que é amplamente empregada em muitos sistemas de recomendação modernos (Resnick et al., 1994).

Os SRs são recursos de *software* que oferecem ao usuário sugestões de itens para utilização, abrangendo uma variedade de contextos de decisão, como sugestões de compras, recomendações musicais e orientações sobre notícias online (Ricci; Rokach; Shapira, 2011). Esses sistemas podem ser adaptados para diversas aplicações, sem haver uma abordagem universalmente superior. A seleção do método a ser empregado depende de diversos fatores relacionados à natureza das informações disponíveis para o desenvolvedor, bem como às características dos itens e dos usuários do sistema em questão (Matos, 2021).

Conforme os autores Ricci, Rokach e Shapira (2011), Bai et al. (2019), Ricci, Rokach e Shapira (2011) e Aggarwal (2016) podemos classificar os SR em até 6 tipos, de acordo com sua abordagem:

- **Baseado em conteúdo**: O sistema se aprimora na sugestão de itens que compartilham semelhanças com aqueles que o usuário apreciou anteriormente. Essa similaridade é determinada com base nas características associadas aos itens que estão sendo comparados. Por exemplo, se um usuário deu uma avaliação positiva a um filme de comédia, o sistema pode aprender a sugerir outros filmes do mesmo gênero (Ricci; Rokach; Shapira, 2011).
- **Filtragem Colaborativa**: Baseia-se na similaridade entre usuários, comparando os itens de usuários que são considerados similares. Por exemplo, se o usuário A possui itens semelhantes aos do usuário B, eles são considerados "vizinhos,"e os itens de A que não estão presentes em B podem ser recomendados a B (Bai et al., 2019).

- **Demográfico**: Este tipo de sistema recomenda itens com base no perfil demográfico do usuário. A suposição é que diferentes recomendações devem ser geradas para diferentes nichos demográficos. Muitos sites da *web* adotam soluções simples e eficazes de personalização com base em dados demográficos. Por exemplo, os usuários são direcionados para sites específicos com base em seu idioma ou país (Ricci; Rokach; Shapira, 2011).

- **Baseado em Conhecimento**: Sistemas de recomendação baseados em conhecimento são úteis para itens menos comuns, como imóveis, automóveis, turismo, serviços financeiros e itens de luxo. Eles utilizam similaridades entre as preferências dos clientes e as descrições dos itens, empregando bases de conhecimento para direcionar o processo de recomendação e permitindo que os usuários especifiquem explicitamente suas preferências (Aggarwal, 2016).

- **Baseado na Comunidade**: Sistemas baseados na comunidade recomendam itens com base nas preferências dos amigos dos usuários, aproveitando a confiança que as pessoas têm nas recomendações de amigos em comparação com recomendações anônimas. Com o aumento das redes sociais, há um interesse crescente nesses sistemas, também conhecidos como sistemas de recomendação social. Eles modelam as relações sociais dos usuários e as preferências de seus amigos, utilizando as avaliações fornecidas por eles para gerar recomendações (Ricci; Rokach; Shapira, 2011).

- **Híbridos**: Esses sistemas são baseados na combinação das técnicas mencionadas acima. Um sistema híbrido que combina as técnicas A e B procura utilizar as vantagens de A para resolver as desvantagens de B. Por exemplo, os métodos de filtragem colaborativa sofrem com problemas de novos itens, ou seja, não podem recomendar itens que não possuem avaliações. Isso não limita as abordagens baseadas em conteúdo, já que a previsão para novos itens é baseada em suas descrições (características), que geralmente são facilmente disponíveis (Ricci; Rokach; Shapira, 2011).

2.3 *Web Scraping*

Web scraping, ou raspagem de dados da *web*, é uma técnica usada para coletar dados não estruturados de sites, visando convertê-los em dados estruturados. Isso implica transformar as informações obtidas em estruturas compreensíveis, como planilhas, bancos de dados ou arquivos com valores separados por vírgula (Sirisuriya et al., 2015).

Corroborando com esse princípio, Zhao (2017) relata que o *web scraping* é o processo automatizado de extrair dados da *web*, requerendo apenas programação específica para a automação do *software*. Atualmente, é possível converter informações de grandes sites em conjuntos de dados organizados por meio de técnicas de *web scraping*.

2.4 Algoritmos e Técnicas para Sistema de recomendação

Todas as técnicas serão implementadas utilizando a linguagem de programação *Python*. Segundo Santos (2022a), *Python* dispõe de uma ampla gama de bibliotecas e ferramentas complementares para solucionar problemas no processamento de dados e na inteligência artificial, além de ser uma linguagem de fácil leitura.

Foram utilizadas diversas técnicas e algoritmos para aprimorar os resultados do sistema de recomendação neste projeto. Entre esses métodos, inclui-se o algoritmo de similaridade de cossenos, similaridade de Jaccard, BM25, TF-IDF, BERT *Embedding* e o modelo da OpenAI Chatgpt.

2.4.1 Similaridade de Cossenos

Em geral, uma função de similaridade é uma função que aceita a entrada de dois objetos e calcula a similaridade entre eles, retornando em forma de números reais. O valor retornado pela função de similaridade geralmente varia no intervalo de 0 a 1. Neste método, as similaridades entre dois vetores n-dimensionais são calculadas procurando o valor do cosseno do ângulo entre os dois. A fórmula de similaridade de cossenos é dada da seguinte forma conforme a figura 1 (Fiarni; Maharani, 2019).

Figura 1 – Similaridade de Cossenos

$$similarity\ (x, y) = \cos(\theta) = \frac{\sum_{i=1}^{n} x.y}{\|x\|\|y\|}$$

Fonte: Fiarni e Maharani (2019)

Onde:

- x e y são os vetores que queremos comparar.
- θ é o ângulo entre os vetores x e y.
- $\sum_{i=1}^{n} x \cdot y$ são as i-ésimas componentes dos vetores x e y.
- $||x|| \cdot ||y||$ são as normas dos vetores x e y.

Quanto mais próximo de 1 for o resultado da função de similaridade, mais semelhantes são considerados os dois objetos avaliados. Em uma função que produz um valor no intervalo de 0 a 1, o valor 1 indica que os dois objetos são exatamente iguais, enquanto o valor 0 indica que são completamente diferentes (Soares, 2017).

2.4.2 Similaridade de Jaccard

O coeficiente de Jaccard é um algoritmo amplamente utilizados nos sistemas de recomendação, ele tem como proposta é um método popular para calcular similaridades entre usuários ou itens. No cálculo, demonstrado na figura 2, considera apenas o número de avaliações comuns entre os dois usuários. Os benefícios de utilizar esse método são maximizados quando o número de avaliações comuns é maior (Jain; Mahara; Tripathi, 2020).

Figura 2 – Similaridade de Jaccard

$$\mathrm{Sim}(u_a, u_b) = \frac{|I_{u_a}| \cap |I_{u_b}|}{|I_{u_a}| \cup |I_{u_b}|}$$

Fonte: Pant et al. (2020)

Onde:

- I_{ua} é o conjunto de itens avaliados pelo usuário ua.
- I_{ub} é o conjunto de itens avaliados pelo usuário ub.
- $|I_{ua} \cap I_{ub}|$ é o conjunto de itens avaliados pelo usuário ub.
- $|I_{ua} \cup I_{ub}|$ é o número total de itens distintos nos conjuntos I_{ua} e I_{ub}.

Neste estudo, para o sistema de recomendação, uma dos algoritmos empregado será a Similaridade de Jaccard para calcular a similaridade com base na *query* de busca e nos textos já pré-processados. Essa técnica auxiliará na avaliação da sobreposição entre os conjuntos de palavras-chave presentes na *query* de busca e nos trabalhos considerados para a recomendação.

2.4.3 BM25

O BM25 (*Best Matching 25*) é um algoritmo de pontuação usado para classificar e recuperar documentos relevantes em uma consulta de pesquisa. Ele é uma extensão do modelo de *Okapi* BM25, que considera a frequência dos termos e o comprimento do documento para calcular a relevância. Ele se destaca como um método eficaz para classificar e recuperar documentos de acordo com a relevância em várias aplicações de processamento de linguagem natural e recuperação de informações (Robertson; Zaragoza et al., 2009).

2.4.4 TF-IDF

O algoritmo TF-IDF (*Term Frequency-Inverse Document Frequency*) é uma técnica de ponderação de palavras que avalia a importância de um termo em um documento. A Frequência do Termo (TF) mede quantas vezes um termo aparece em um documento, enquanto a Frequência

Inversa do Documento (IDF) avalia a raridade de um termo em todo o corpus. O TF-IDF é calculado como o produto desses dois fatores e é usado para criar perfis de usuários e representações de itens em um modelo vetorial. A similaridade de cosseno é então aplicada entre o perfil do usuário e as representações de itens para recomendações personalizadas. Este método garante que termos raros e importantes tenham maior peso do que termos comuns, proporcionando uma avaliação mais precisa da relevância dos itens (Afoudi; Lazaar; Al Achhab, 2021).

2.4.5 BERT Embedding

BERT (*Bidirectional Encoder Representations from Transformers*) é um modelo de processamento de linguagem natural pré-treinado que foi desenvolvido para entender o contexto das palavras em uma frase de forma mais profunda. Ele utiliza uma arquitetura baseada em *transformers*, permitindo que o modelo compreenda o significado das palavras com base no contexto das palavras que as rodeiam. O BERT é capaz de capturar relações entre palavras em duas direções, o que o torna eficaz em uma ampla gama de tarefas de processamento de linguagem natural, incluindo compreensão de linguagem, tradução automática, perguntas e respostas, entre outras (Devlin et al., 2018).

2.4.6 Chatgpt

O *ChatGPT* é uma criação da empresa OpenIA, uma ferramenta alimentada por inteligência artificial projetada para entender e gerar texto de maneira semelhante à de um humano. Ele é construído com base na tecnologia de modelos de linguagem, especificamente o GPT (*Generative Pre-trained Transformer*), que permite ao sistema compreender e produzir texto em uma ampla variedade de estilos e contextos (OpenIA, 2022).

2.5 Avaliação de um Sistema de Recomendação

Segundo Aggarwal (2016) a avaliação de sistemas de recomendação envolve diversos pontos-chave, incluindo precisão, cobertura, confiança, novidade, serendipidade, diversidade, robustez, estabilidade e escalabilidade. Por exemplo, a precisão refere-se à exatidão das recomendações feitas, enquanto a novidade avalia a capacidade do sistema de apresentar itens desconhecidos ao usuário, aumentando sua descoberta de novos interesses. A cobertura, por outro lado, mede a extensão em que o sistema é capaz de recomendar itens para diferentes tipos de usuários, abrangendo uma variedade de preferências.

Embora a precisão das previsões seja importante, a eficácia de um sistema de recomendação não se resume apenas a antecipar as preferências do usuário. Além disso, os usuários frequentemente buscam descobrir novos itens, preservar sua privacidade e receber respostas rápidas do sistema. Essas expectativas mais amplas tornam a precisão das previsões apenas uma parte da equação para avaliar um sistema de recomendação (Ricci; Rokach; Shapira, 2011).

2.6 Trabalhos Relacionados

O estudo conduzido por Kreutz e Schenkel (2022) oferece uma revisão abrangente da literatura relacionada aos SR de artigos científicos. A pesquisa abrange diversos aspectos, incluindo fontes de dados, algoritmos empregados e métricas utilizadas para avaliar a eficácia dos sistemas. Ao final, são sintetizados os principais resultados extraídos das publicações analisadas.

No trabalho de Ferreira (2023) foi proposto um SR para teses e dissertações do Programa de Pós-graduação de Engenharia Elétrica e Computação (PPgEEC) da UFRN, usando *web scraping* e *embeddings* da OpenAI. O objetivo é gerar recomendações de trabalhos acadêmicos com base nos dados coletados, incluindo resultados de agrupamentos por similaridade.

O estudo de Souza, Lichtnow e Gasparini (2022) propõe uma estratégia de pós processamento para otimizar a eficiência de um SR de artigos científicos. A estratégia considera o ano de publicação e evita a recomendação de uma extensa lista de artigos do mesmo ano, além de substituir artigos que compartilham os mesmos termos do perfil do usuário. O objetivo é diversificar as recomendações e evitar a saturação de temas específicos.

Na pesquisa de Bai et al. (2019) é dissertado sobre os SR de artigos científicos na qual ela classifica os sistemas em quatro grupos com base em suas técnicas de recomendação: filtragem baseada em conteúdo, filtragem colaborativa, método baseado em grafo e método híbrido. Verificou-se que os métodos híbridos e baseados em conteúdo se destacam como as técnicas mais utilizadas em sistemas de recomendação de artigos, essa constatação evidencia que a filtragem colaborativa, por não considerar o conteúdo textual dos artigos e sim as avaliações de outros usuários, pode apresentar limitações na entrega de recomendações relevantes à pesquisa atual do usuário.

No estudo conduzido por Lee, Lee e Kim (2013), um sistema de recomendação foi elaborado empregando métodos de filtragem colaborativa, *web crawler* para a coleta de artigos, e técnicas de pré-processamento, como remoção de *stopwords* e lematização. Contudo, o trabalho destaca uma limitação significativa, pois o sistema não dispunha de informações adicionais sobre o interesse contínuo do usuário nos artigos, tornando desafiador distinguir recomendações verdadeiramente relevantes.

De acordo com os estudos de Bai et al. (2019) e Lee, Lee e Kim (2013), a filtragem colaborativa foi excluída deste estudo por diversos motivos. Primeiramente, o repositório online da UEPB não fornece informações adicionais sobre o interesse contínuo do usuário em um artigo, impossibilitando a avaliação de sua relevância. Ademais, a filtragem colaborativa apresenta a limitação de não utilizar o conteúdo textual como base para as recomendações, o que pode levar à entrega de sugestões irrelevantes.

Sukestiyarno, Sapolo e Sofyan (2023) em seu estudo sobre sistemas de recomendação de cursos, foi realizado um comparativo entre os algoritmos de Similaridade de Jaccard e Similaridade de Cosseno. Utilizando o conjunto de dados *Coursera Free Dataset*, composto por 975 instâncias, o trabalho evidenciou situações em que cada algoritmo superou o outro. Esses resultados destacam a eficácia do Sistema de Recomendação com os algoritmos de Similaridade

de Jaccard e Cosseno.

O presente trabalho tem como diferencial às abordagens integrativas implementadas no desenvolvimento de um Sistema de Recomendação (SR) de artigos científicos. O foco reside na busca pelos métodos e técnicas mais eficazes, visando determinar a combinação mais adequada para criar um SR específico no contexto da UEPB.

3 METODOLOGIA

Neste capítulo estão presentes os métodos utilizados para a coleta de dados e os algoritmos aplicados.

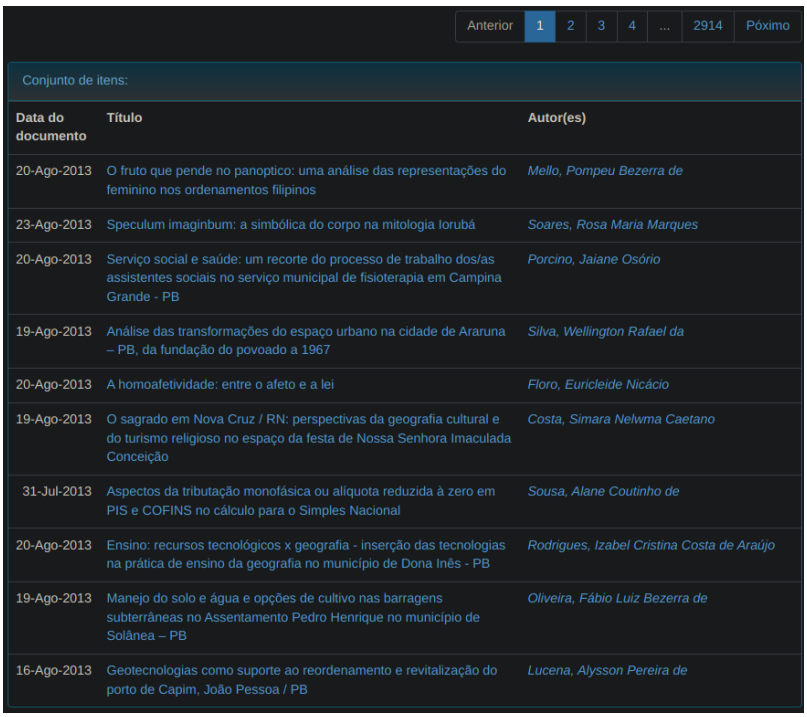
3.1 Coleta de Dados

Até a data desta pesquisa, o repositório online da UEPB abriga aproximadamente 28 mil artigos, cobrindo uma variedade de temas ao longo do período de 2000 a 2024. Para coletar os dados desses artigos, empregamos a técnica de *web scraping*, possibilitando a extração automatizada das informações de páginas da *web* em grande escala. Esse processo foi facilitado pela utilização da linguagem *Python* em conjunto com a biblioteca *Beautiful Soup*, desenvolvida especialmente para analisar e extrair dados de arquivos HTML e XML.

3.2 Estrutura do Site e Extração dos Dados

Ao acessar o repositório, nos deparamos com uma tabela que exibe os títulos dos 10 primeiros artigos do repositório, acompanhados pelos respectivos links de acesso, como ilustrado na figura 3. No entanto, necessitamos acessar os demais dados de cada artigo individualmente. Para isso, é preciso compilar uma lista com os links específicos de acesso a cada artigo. Tal tarefa requer a navegação página por página para extrair esses links de forma isolada, facilitando assim a extração dos dados de cada artigo.

Figura 3 – Estrutura da página.



Fonte: autor.

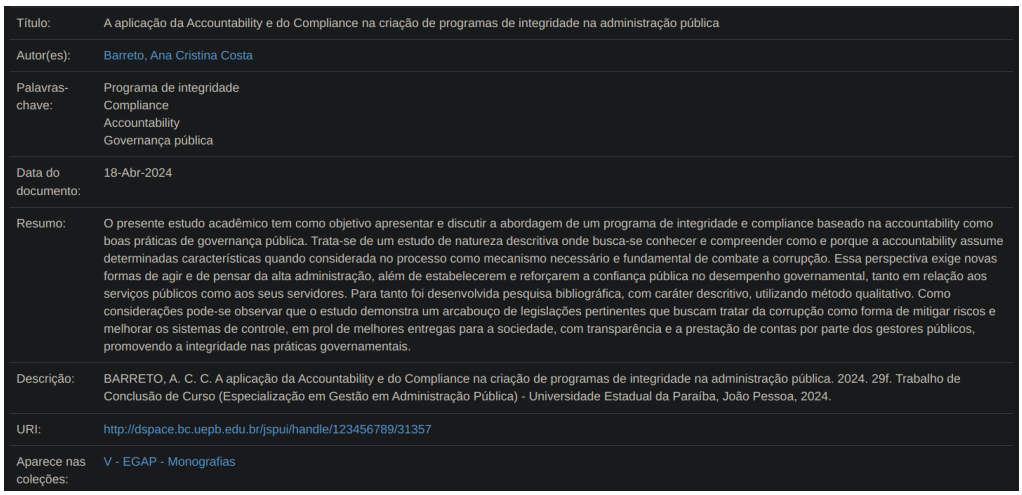
Para evitar a necessidade de percorrer todas as páginas repetidamente, optamos por utilizar um arquivo CSV contendo uma coluna de URLs para armazenar os links dos artigos. Para alcançar esse objetivo, é necessário inicialmente desenvolver um algoritmo que seja capaz de extrair os links de uma única página. Posteriormente, expandiremos esse algoritmo para abranger todas as outras páginas do repositório.

Para realizar essa tarefa, tivemos que lidar com um ponto específico. A página contém mais de um elemento *table* que contém elementos *td*. Portanto, foi necessário selecionar todos os elementos *td* que possuem as classes *oddRowOddCol* e *evenRowOddCol*. Para isso foi desenvolvida a função *getLinkArticles*. Essa função busca os elementos *td* dentro da tabela, na qual cada linha representa os dados de um artigo. Iterando sobre cada linha selecionada, buscamos pelo elemento *a* para extrair o valor da propriedade *href*, que contém o *link* do artigo. Ao final do processo, retornamos todos os *links* encontrados naquela página.

Para completar a coleta de todos os links das páginas, precisamos iterar por cada uma delas. O objetivo é reunir todos esses links e salvá-los em um arquivo CSV para uso futuro na extração dos dados de cada artigo. Para solucionar esse desafio, foi criado uma função denominada *getAllLinkFromAllPages*. Essa função inicia na primeira página, coleta todos os links disponíveis e os armazena. Em seguida, busca por um elemento *a* com o texto "Próximo", desde que não esteja desabilitado. Se esse elemento existir, obtemos o link da próxima página a partir do atributo *href*. Repetimos esse processo até alcançarmos a última página.

Com o arquivo CSV contendo todos os links dos artigos, o próximo passo é desenvolver um algoritmo para extrair os dados de cada artigo individualmente. Este processo será replicado para todos os links armazenando os dados em um *dataframe*. Posteriormente, foi utilizado esse *dataframe* para realizar uma análise dos dados coletados. Primeiramente, estabeleceu-se quais atributos serão extraídos: título, autor, palavras-chave, data, descrição e URI. Todos esses dados estão presentes na estrutura do artigo quando acessamos os links correspondentes como apresenta a figura 4.

Figura 4 – Estrutura de um artigo no repositório.



Título:	A aplicação da Accountability e do Compliance na criação de programas de integridade na administração pública
Autor(es):	Barreto, Ana Cristina Costa
Palavras-chave:	Programa de integridade Compliance Accountability Governança pública
Data do documento:	18-Abr-2024
Resumo:	O presente estudo acadêmico tem como objetivo apresentar e discutir a abordagem de um programa de integridade e compliance baseado na accountability como boas práticas de governança pública. Trata-se de um estudo de natureza descritiva onde busca-se conhecer e compreender como e porque a accountability assume determinadas características quando considerada no processo como mecanismo necessário e fundamental de combate a corrupção. Essa perspectiva exige novas formas de agir e de pensar da alta administração, além de estabelecerem e reforçarem a confiança pública no desempenho governamental, tanto em relação aos serviços públicos como aos seus servidores. Para tanto foi desenvolvida pesquisa bibliográfica, com caráter descritivo, utilizando método qualitativo. Como considerações pode-se observar que o estudo demonstra um arcabouço de legislações pertinentes que buscam tratar da corrupção como forma de mitigar riscos e melhorar os sistemas de controle, em prol de melhores entregas para a sociedade, com transparência e a prestação de contas por parte dos gestores públicos, promovendo a integridade nas práticas governamentais.
Descrição:	BARRETO, A. C. C. A aplicação da Accountability e do Compliance na criação de programas de integridade na administração pública. 2024. 29f. Trabalho de Conclusão de Curso (Especialização em Gestão em Administração Pública) - Universidade Estadual da Paraíba, João Pessoa, 2024.
URI:	http://dspace.bc.uepb.edu.br/jspui/handle/123456789/31357
Aparece nas coleções:	V - EGAP - Monografias

Fonte: autor.

Com todos os atributos mapeados em uma classe criada, basta iterarmos por cada link, coletando os dados e salvando-os no *dataframe*. Para isso, desenvolvemos uma função chamada *getDataWithSpecifArticle*. Esta função recebe um link, busca os índices de título, autor, palavras-chave, data, descrição e URI, e coleta os dados correspondentes. Se algum desses atributos não estiver presente, o valor salvo será vazio.

Com a função *getDataWithSpecifArticle* definida, basta criarmos o *dataframe* com as colunas desejadas, iterarmos por todos os links, e coletarmos os dados de cada artigo. Após a coleta, o *dataframe* será utilizado para realizar uma análise detalhada dos dados. Em seguida, geraremos um arquivo CSV que servirá como base de dados para os nossos algoritmos consumirem.

Após a coleta dos dados, foi gerado um *dataframe* com 28.906 registros. Nesta etapa, é necessário analisar os dados para identificar e quantificar registros com dados faltantes. Se o número de registros incompletos for significativo, uma análise mais aprofundada será necessária para entender a causa da falta de padronização. Após a verificação, foi observado que apenas 5 registros apresentam dados faltantes. Como este número é pequeno em relação ao total de registros, optamos por desconsiderar esses registros e removê-los do *dataframe*. Finalmente, os dados foram salvos em um arquivo CSV para uso posterior. Todos os *scripts* usados para coleta de dados estão no Apêndice A.

3.3 Tratamento dos Dados

Com os dados padronizados e sem colunas com dados vazios, é necessário realizar um pré-processamento para otimizar os algoritmos de recomendação. Foi criado um novo atributo para cada artigo, que será a junção do título, resumo e palavras-chave. Em seguida, aplicamos a lematização e a remoção de *stopwords* a esse novo atributo. Utilizamos a biblioteca *Spacy*, uma ferramenta avançada de processamento de linguagem natural em *Python*, para realizar essas operações. O *script* utilizado para o tratamento dos dadoss está no Apêndice B .

3.4 Algoritmos

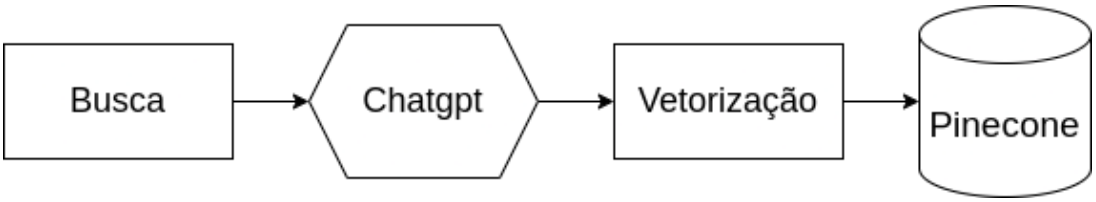
Neste trabalho, foi demonstrado um sistema de recomendação que aplica os algoritmos de similaridade cosseno, similaridade Jaccard, BM25, BERT e ChatGPT. Por fim, avaliamos os resultados obtidos por cada algoritmo e identificamos a melhor combinação para resolver o problema, considerando critério como tempo de resposta, coesão e escalabilidade.

Para a implementação dos algoritmos, foi utilizado diversas bibliotecas, incluindo Pandas, *Sklearn*, *NLTK*, *BM25Okapi*, *Pinecone*, entre outras. Essas bibliotecas fornecem o conjunto de ferramentas necessárias para desenvolver os algoritmos de recomendação selecionados neste trabalho. Todos os algoritmos seguem uma estrutura semelhante: o arquivo CSV, com os dados dos artigos da UEPB, é lido, a vetorização é realizada na coluna que passou pelo pré-processamento e cada algoritmo é aplicado de acordo com suas características específicas, conforme demonstrado no Apêndice C

No modelo *BERT*, foi necessário definir um algoritmo para o cálculo da similaridade, uma vez que *BERT* é um modelo pré-treinado que gera a vetorização dos textos. Utilizou-se o *BERT* em conjunto com a similaridade de cossenos. Todo o processamento de vetorização foi realizado pelo *BERT*, enquanto o cálculo da similaridade foi feito pelo algoritmo de similaridade de cossenos.

Já o *ChatGPT*, foi utilizado como uma ferramenta para auxiliar na melhoria da *string* de busca do usuário. O novo fluxo funcionará da seguinte forma: o usuário insere sua consulta inicial e, em seguida, essa consulta é processada pelo *ChatGPT* para gerar uma versão aprimorada da *string* de busca. Em seguinta vetorizamos o campo no qual ocorreu o pré-processamento e armazenos no *Pinecone*, um banco de dados de vetores projetado para otimizar a eficiência das inteligências artificiais, finalmente utilizaremos a função de consulta do Pinecone, passando a *string* de busca aprimorada. O *Pinecone* aplicará a similaridade de cossenos para encontrar os resultados mais relevantes, figura 5.

Figura 5 – Fluxo da aplicação com ChatGPT.



Fonte: autor.

4 RESULTADOS

Este capítulo detalhará as consultas realizadas, os resultados obtidos por cada algoritmo, o tempo de execução de cada um e uma discussão dos resultados. Para validar o desempenho de cada sistema, foram sorteados dois temas para pesquisa. Todos os algoritmos utilizaram a base de dados da UEPB, que contém mais de 28 mil registros. Para cada consulta, as 8 primeiras recomendações com maior similaridade foram selecionadas e, em seguida, filtradas para manter apenas aquelas com 70% ou mais de similaridade com o termo pesquisado.

4.1 Consulta 1

Para a primeira consulta, o tema escolhido foi "metodologias ativas na educação". Serão apresentadas as 8 primeiras recomendações de cada algoritmo, quando disponíveis.

4.1.1 Similaridade de Cossenos

- "O ensino de geografia na perspectiva das metodologias ativas: estratégias para a dinamização do processo de ensino-aprendizagem"(Souto, 2022).
- "Aplicação de metodologias ativas para o ensino de Ciências/Biologia no âmbito do PIBID"(Silva, 2022).
- "Um Estudo Sobre Metodologias Ativas Como Abordagem Metodológica no Ensino de Programação"(Nóbrega, 2021).
- "Limites e perspectivas de discentes e docentes na utilização de metodologias ativas no ensino da Enfermagem: Revisão integrativa"(Batista, 2019a).
- "Metodologias ativas na educação matemática: uma análise sobre o uso da Sala de Aula Invertida como recurso didático"(Cezar, 2021).
- "O uso das metodologias ativas na educação infantil: processos de ensino e de aprendizagem"(Albuquerque, 2023).
- "Metodologias ativas no ensino de Química a nível médio: uma revisão sistemática no período de 2016 a 2021 na revista Química nova na escola"(Araujo, 2022b).
- "A relevância do planejamento nas aulas da educação infantil"(Lima, 2022).

4.1.2 Similaridade de Jaccard

- "A utilização de metodologias ativas na educação infantil: pontos e contra-pontos"(Neto, 2023).

- "O uso das metodologias ativas na educação infantil: processos de ensino e de aprendizagem"(Albuquerque, 2023).
- "As metodologias ativas e tecnologias educacionais como potencialização da aprendizagem na Educação Física"(Júnior, 2019).

4.1.3 BM25

- "Metodologias de Ensino de Matemática para alunos surdos: uma Revisão Sistemática da Literatura"(Araujo, 2022c).
- "Metodologias ativas na educação matemática: uma análise sobre o uso da Sala de Aula Invertida como recurso didático"(Cezar, 2021).
- "Metodologias ativas no ensino da matemática: o uso de jogos como ferramenta de ensino-aprendizagem"(Sousa, 2020).
- "Um Estudo Sobre Metodologias Ativas Como Abordagem Metodológica no Ensino de Programação"(Nóbrega, 2021).
- "Modelagem matemática como metodologia de ensino na educação básica"(Sousa, 2019).
- "Gamificação na educação: perspectivas e possibilidades no ensino de Física"(Araujo, 2022a).
- "O uso de metodologias ativas na disciplina de matemática no ensino remoto emergencial em escola pública paraibana"(Lacerda, 2023).
- "Ensino de logaritmos e função logarítmica: reflexões sobre experiências do ensino médio à formação inicial de professores de matemática"(Morais, 2019).

4.1.4 Bert

- "Metodologias ativas no ensino da matemática: o uso de jogos como ferramenta de ensino-aprendizagem"(Sousa, 2020).
- "Metodologias ativas na educação matemática: uma análise sobre o uso da Sala de Aula Invertida como recurso didático"(Cezar, 2021).
- "Metodologia facilitadora da aprendizagem no ensino de física: experimentação"(Lacerda, 2020).
- "Ensino da matemática na EJA: desafios e opções metodológicas"(Dantas, 2021).
- "A utilização das tecnologias como mecanismo de promoção de aprendizagens significativas em Física"(Gadelha, 2015).

- "Experimentos e simulações computadorizadas como instrumentos de ensino aprendizagem"(Caldas, 2019).
- "Contribuições dos sistemas de informações gerenciais para a gestão escolar"(Caldas, 2018).

4.1.5 Chatgpt

- "Motivação para aprender e metodologias ativas: potencialidades no processo educativo"(Batista, 2019b).
- "As metodologias ativas e tecnologias educacionais como potencialização da aprendizagem na Educação Física"(Júnior, 2019).
- "O ensino de geografia na perspectiva das metodologias ativas: estratégias para a dinamização do processo de ensino-aprendizagem"(Souto, 2022).
- "Metodologias ativas na educação matemática: uma análise sobre o uso da Sala de Aula Invertida como recurso didático"(Cezar, 2021).
- "Metodologia ativa no processo de Ensino da Matemática voltada para a dislexia: uma abordagem de ensino e aprendizagem utilizando o método de gamificação"(Bezerra, 2022).
- "Limites e perspectivas de discentes e docentes na utilização de metodologias ativas no ensino da Enfermagem: Revisão integrativa"(Batista, 2019a).
- "A utilização de metodologias facilitadoras no processo de ensino-aprendizagem de Inglês como língua estrangeira"(Alves, 2022).
- "O uso das metodologias ativas na educação infantil: processos de ensino e de aprendizagem"(Albuquerque, 2023).

4.2 Consulta 2

Para a segunda consulta, vamos abordar um tema mais específico: "a evolução das políticas de privacidade na era digital". Com essa abordagem, poderemos analisar como os diferentes algoritmos se comportam nesse cenário. Serão apresentadas as 8 primeiras recomendações de cada algoritmo, quando disponíveis

4.2.1 Similaridade de Cossenos

- "O bom uso das ferramentas digitais frente ao direito à privacidade"(Holanda, 2018).
- "Aplicação da lei nº 13.709/2018 como instrumento para a efetivação do direito à privacidade nas relações de consumo durante a pandemia"(Bezerra, 2021).

- "Proteção da privacidade de dados pessoais: relações de consumo e coleta de dados por High Techs, redes sociais e internet"(Costa, 2021).
- "Proteção dos dados pessoais e direito digital: a acepção da privacidade no contexto da LGPD"(Silva, 2019).
- "O direito à privacidade e a produção de provas no processo penal"(Ramos, 2014).
- "Autodeterminação informativa: da recusa ao esquecimento"(Calvet, 2020).
- "Contabilidade digital: um estudo sobre a nova era digital na visão dos alunos da UEPB"(Carneiro, 2018).
- "Privacidade e o consumidor na internet: uma análise do marco civil e a evolução legislativa"(Cunha, 2015).

4.2.2 Similaridade de Jaccard

- "Privacidade e o consumidor na internet: uma análise do marco civil e a evolução legislativa"(Cunha, 2015).
- "Implementação das políticas públicas de inclusão digital na educação em tempos de pandemia"(Silva, 2021).
- "Proteção dos dados pessoais e direito digital: a acepção da privacidade no contexto da LGPD"(Silva, 2019).
- "A implementação de políticas de inclusão digital: uma análise dos telecentros no município de Patos - PB"(Sousa, 2015).

4.2.3 BM25

- "Laboratórios do PROINFO como recurso para o uso pedagógico da informática nas escolas estaduais da 7ª Gerência Regional de Ensino da Paraíba"(Jó, 2017).
- "A educação feminina e a representação dos perfis normalistas nos periódicos paraibanos: dos relatos às imagens"(Silva, 2020).
- "Análise das boas práticas de governança corporativa aplicada a uma cooperativa de crédito na Paraíba"(Silva, 2023).
- "O direito à privacidade e a produção de provas no processo penal"(Ramos, 2014).
- "Os impactos da literacia digital para a Arquivologia contemporânea: a realidade do corpo funcional dos arquivos da CAGEPA, PBPREV e SEAD"(Santos, 2023).

4.2.4 Bert

- "Vulnerabilidades em Redes Windows"(Lourenço, 2013).
- "*Privacy by design* como obrigação de segurança no tratamento de dados pessoais"(Pinheiro, 2021).
- "A Constitucionalização dos Sistemas de Informática"(Andrade, 2014).
- "Análise da Aplicabilidade da Legislação Arquivística no Fórum Juiz Inácio Machado de Souza"(Souza, 2014).
- "Repositórios digitais: histórico e características"(Barroso, 2017).
- "Utilização da Tecnologia *Blockchain* no Setor Público Brasileiro"(Santos, 2022c).
- "Segurança e privacidade na internet das coisas: estudo de caso com a *Thinger.Io* Platform"(França, 2022).

4.2.5 Chatgpt

- "Implementação das políticas públicas de inclusão digital na educação em tempos de pandemia"(Silva, 2021).
- "Democracias desenvolvidas em riscos: controle e manipulação de informações na era digital e os danos às garantias institucionais democráticas"(Filho, 2021).
- "Soberania e direitos humanos na era digital: uma análise da espionagem americana e do conflito privacidade-segurança"(Melo, 2014).
- "Proteção dos dados pessoais e direito digital: a acepção da privacidade no contexto da LGPD"(Silva, 2019).
- "Pirataria digital de livros: entre a democratização da leitura e a proteção dos direitos humanos"(Santos, 2022b).

4.3 Tempo de execução

Para a execução dos algoritmos, foi utilizado um notebook Avell equipado com um processador Intel Core i7 de 10ª geração, com 12 núcleos e uma velocidade de 2,60 GHz. A máquina possui 16 GB de RAM e uma placa de vídeo GeForce GTX 1650, rodando o sistema operacional Ubuntu 22.04. Com essa configuração, obtivemos os seguintes resultados apresentados na tabela 1.

	Cossenos	Jaccard	BM25	Bert	Chatgpt
Consulta 1	4.9s	2 min 3s	13.5s	2 min 3s	8.5s
Consulta 2	2.8s	1 min 57s	14.1s	1 min 59s	3s

Tabela 1 – Tempo de execução de cada algoritmo.

4.4 Discussão dos Resultados

Após a análise dos resultados obtidos, constatou-se que, em termos de tempo de execução, os algoritmos de similaridade de Cosseno e *ChatGPT* se destacaram. No quesito escalabilidade, todos os algoritmos demonstraram bom desempenho, lidando com mais de 28 mil artigos presentes na base de dados. No quesito coesão, observamos que os algoritmos que utilizam o *BERT* e *BM25* de em algumas recomendações, acabaram tangenciando o tema ou sugerindo artigos que não eram coesos com a busca inserida. Os algoritmos que obtiveram melhor desempenho nesse critério foram a técnica de similaridade de Cossenos, similaridade de Jaccard e o *ChatGPT*.

Ao ampliarmos os critérios para incluir diversidade e robustez, observamos que a aplicação que utiliza o *ChatGPT* como parte integrante apresenta maior robustez. Isso se deve à possibilidade de ajustar variáveis para otimizar ainda mais o desempenho. Além disso, demonstrou maior diversidade, isso ficou evidente na consulta 2, com o título "A evolução das políticas de privacidade na era digital", na qual uma das recomendações foi o artigo "Pirataria digital de livros: entre a democratização da leitura e a proteção dos direitos humanos", fazendo um paralelo válido entre pirataria e políticas de privacidade.

Em relação à quantidade de recomendações, os algoritmos de similaridade de Cosseno, *BERT* e ChatGPT apresentaram o maior número de artigos recomendados com similaridade acima de 70% com o termo pesquisado. Por fim, o algoritmo que combinou ChatGPT e similaridade de cosseno obteve os melhores resultados, considerando apenas os critérios listados anteriormente.

5 CONSIDERAÇÕES FINAIS

Com o aumento exponencial de dados disponíveis na internet, a busca por informações relevantes tornou-se cada vez mais desafiadora. Frequentemente, ao realizar uma consulta, somos bombardeados por uma avalanche de informações, o que dificulta a exploração eficaz das opções disponíveis. No contexto acadêmico, esse desafio é ainda mais acentuado devido ao alto índice de novos artigos publicados diariamente. Este trabalho propõe a criação de um sistema de recomendação para artigos científicos, utilizando diversos algoritmos com o objetivo de identificar qual apresenta o melhor desempenho, avaliando critérios como tempo de resposta, coesão, escalabilidade, diversidade e robustez.

Para o presente trabalho, foram elaborados os objetivos, os quais foram alcançados em sua totalidade: a fundamentação teórica na qual foi desenvolvida com base no estado da arte, servindo como ponto de partida para pesquisadores da área de sistemas de recomendação, a coleta de dados, na qual foi criado um *dataframe* com mais de 28 mil registros, a análise exploratória que incluiu a padronização dos dados e o pré-processamento e a aplicados dos algoritmos de recomendação, utilizando o *dataframe* criado na etapa de coleta de dados.

Para a avaliação do sistema de recomendação, foi possível analisar apenas os aspectos técnicos. No entanto, para obter uma precisão maior sobre o impacto do sistema na produção de trabalhos acadêmicos, seria interessante contar com o *feedback* dos usuários e verificar o quanto o sistema ajudou os pesquisadores em seus estudos.

As consultas realizadas aos sistemas de recomendação alcançaram o objetivo inicial propostos de comparar e identificar quais algoritmos de recomendação apresentam melhor desempenho. As recomendações demonstraram uma forte correlação com os temas dos artigos pesquisados, comprovando a eficácia do sistema em identificar conteúdos relevantes e coesos.

Para trabalhos futuros, é possível explorar ainda mais o potencial do *ChatGPT*, além de disponibilizar o sistema de recomendação para auxiliar os estudantes em suas pesquisas. Também seria interessante permitir que os usuários escolham e alternem entre diferentes algoritmos no sistema, possibilitando o recebimento de *feedback* sobre quais algoritmos foram os mais utilizados e eficazes.

REFERÊNCIAS

AFOUDI, Y.; LAZAAR, M.; Al Achhab, M. Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network. *Simulation Modelling Practice and Theory*, v. 113, p. 102375, 2021. ISSN 1569-190X. Disponível em: https://www.sciencedirect.com/science/article/pii/S1569190X21000836. Citado na página 20.

AGGARWAL, C. C. *Recommender Systems - The Textbook*. [S.l.]: Springer, 2016. 1-498 p. ISBN 978-3-319-29659-3. Citado 4 vezes nas páginas 11, 16, 17 e 20.

ALBUQUERQUE, T. de L. O uso das metodologias ativas na educação infantil: processos de ensino e de aprendizagem. 2023. Citado 3 vezes nas páginas 27, 28 e 29.

ALCARDE, C. C. *Processamento de Linguagem Natural: O Poder da Linguagem na Era da Inteligência Artificial*. [S.l.: s.n.], 2023. 60 p. Citado na página 15.

ALVES, S. das N. A utilização de metodologias facilitadoras no processo de ensino-aprendizagem de inglês como língua estrangeira. 2022. Citado na página 29.

ANDRADE, R. S. A constitucionalização dos sistemas de informática. 2014. Citado na página 31.

ARAUJO, A. L. da S. Gamificação na educação: perspectivas e possibilidades no ensino de física. 2022. Citado na página 28.

ARAUJO, M. S. G. de. Metodologias ativas no ensino de química a nível médio: uma revisão sistemática no período de 2016 a 2021 na revista química nova na escola. 2022. Citado na página 27.

ARAUJO, S. G. de. Metodologias de ensino de matemática para alunos surdos: uma revisão sistemática da literatura. 2022. Citado na página 28.

BAI, X. et al. Scientific paper recommendation: A survey. *IEEE Access*, v. 7, p. 9324–9339, 2019. Citado 2 vezes nas páginas 16 e 21.

BARROSO, P. A. de L. Repositórios digitais: histórico e características. 2017. Citado na página 31.

BATISTA, R. E. B. Limites e perspectivas de discentes e docentes na utilização de metodologias ativas no ensino da enfermagem: Revisão integrativa. 2019. Citado 2 vezes nas páginas 27 e 29.

BATISTA, T. S. Motivação para aprender e metodologias ativas: potencialidades no processo educativo. 2019. Citado na página 29.

BEZERRA, F. R. B. Metodologia ativa no processo de ensino da matemática voltada para a dislexia: uma abordagem de ensino e aprendizagem utilizando o método de gamificação. 2022. Citado na página 29.

BEZERRA, R. de L. Aplicação da lei nº 13.709/2018 como instrumento para a efetivação do direito à privacidade nas relações de consumo durante a pandemia. 2021. Citado na página 29.

CALDAS, L. E. D. de. Experimentos e simulações computadorizadas como instrumentos de ensino-aprendizagem. 2019. Citado na página 29.

CALDAS, R. F. Contribuições dos sistemas de informações gerenciais para a gestão escolar. 2018. Citado na página 29.

CALVET, C. B. Autodeterminação informativa: da recusa ao esquecimento. 2020. Citado na página 30.

CARNEIRO, G. B. Contabilidade digital: um estudo sobre a nova era digital na visão dos alunos da uepb. 2018. Citado na página 30.

CASELI, H.; FREITAS, C.; VIOLA, R. Processamento de linguagem natural. *Sociedade Brasileira de Computação*, 2022. Citado 2 vezes nas páginas 15 e 16.

CEZAR, A. V. de L. A. Metodologias ativas na educação matemática: uma análise sobre o uso da sala de aula invertida como recurso didático. 2021. Citado 3 vezes nas páginas 27, 28 e 29.

CISCO. Cisco annual internet report, 2018–2023. 2023. Disponível em: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html. Citado na página 11.

COSTA, J. G. da. Proteção da privacidade de dados pessoais: relações de consumo e coleta de dados por high techs, redes sociais e internet. 2021. Citado na página 30.

CUNHA, A. de F. C. Privacidade e o consumidor na internet: uma análise do marco civil e a evolução legislativa. 2015. Citado na página 30.

DANTAS, V. H. M. da N. Ensino da matemática na eja: desafios e opções metodológicas. 2021. Citado na página 28.

DEVLIN, J. et al. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. Disponível em: http://arxiv.org/abs/1810.04805. Citado na página 20.

FERREIRA, H. R. *Sistema de recomendação para trabalhos acadêmicos*. Dissertação (B.S. thesis) — Universidade Federal do Rio Grande do Norte, 2023. Citado na página 21.

FIARNI, C.; MAHARANI, H. Product recommendation system design using cosine similarity and content-based filtering methods. *IJITEE (International Journal of Information Technology and Electrical Engineering)*, v. 3, n. 2, p. 42–48, 2019. Citado na página 18.

FILHO, L. H. G. A. Democracias desenvolvidas em riscos: controle e manipulação de informações na era digital e os danos às garantias institucionais democráticas. 2021. Citado na página 31.

FRANçA, C. de M. Segurança e privacidade na internet das coisas: estudo de caso com a thinger.io platform. 2022. Citado na página 31.

GADELHA, M. G. de A. A utilização das tecnologias como mecanismo de promoção de aprendizagens significativas em física. 2015. Citado na página 28.

HOLANDA, L. D. C. de. O bom uso das ferramentas digitais frente ao direito à privacidade. 2018. Citado na página 29.

JAIN, G.; MAHARA, T.; TRIPATHI, K. N. A survey of similarity measures for collaborative filtering-based recommender system. In: SPRINGER. *Soft Computing: Theories and Applications: Proceedings of SoCTA 2018*. [S.l.], 2020. p. 343–352. Citado na página 19.

Jó, L. R. Laboratórios do proinfo como recurso para o uso pedagógico da informática nas escolas estaduais da 7ª gerência regional de ensino da paraíba. 2017. Citado na página 30.

JúNIOR, A. F. de S. As metodologias ativas e tecnologias educacionais como potencialização da aprendizagem na educação física. 2019. Citado 2 vezes nas páginas 28 e 29.

KREUTZ, C. K.; SCHENKEL, R. Scientific paper recommendation systems: a literature review of recent publications. *International Journal on Digital Libraries*, Springer, v. 23, n. 4, p. 335–369, 2022. Citado na página 21.

LACERDA, A. H. O uso de metodologias ativas na disciplina de matemática no ensino remoto emergencial em escola pública paraibana. 2023. Citado na página 28.

LACERDA, C. R. A. Metodologia facilitadora da aprendizagem no ensino de física: experimentação. 2020. Citado na página 28.

LEE, J.; LEE, K.; KIM, J. G. *Personalized Academic Research Paper Recommendation System*. 2013. Citado 2 vezes nas páginas 13 e 21.

LIMA, S. A. de. A relevância do planejamento nas aulas da educação infantil. 2022. Citado na página 27.

LOURENçO, T. J. B. Vulnerabilidades em redes windows. 2013. Citado na página 31.

MACHADO, F. N. R. *Big Data O Futuro dos Dados e Aplicações*. [S.l.]: Érica, 2018. Citado na página 11.

MATOS, P. C. Estudo comparativo de algoritmos de sistemas de recomendação de filmes. 2021. Citado na página 16.

MCSHANE, M.; NIRENBURG, S. *Linguistics for the Age of AI*. [S.l.: s.n.], 2021. ISBN 9780262363136. Citado na página 15.

MELO, C. G. de. Soberania e direitos humanos na era digital: uma análise da espionagem americana e do conflito privacidade-segurança. 2014. Citado na página 31.

MORAIS Ângela da S. Ensino de logaritmos e função logarítmica: reflexões sobre experiências do ensino médio à formação inicial de professores de matemática. 2019. Citado na página 28.

NETO, V. G. da S. A utilização de metodologias ativas na educação infantil: Pontos e contra-pontos. 2023. Citado na página 27.

NóBREGA, D. D. da. Um estudo sobre metodologias ativas como abordagem metodológica no ensino de programação. 2021. Citado 2 vezes nas páginas 27 e 28.

OPENIA. *Aligning language models to follow instructions*. 2022. Disponível em: https://openai.com/index/instruction-following/. Acesso em: 25 de Nov de 2023. Citado na página 20.

PANT, M. et al. *Soft computing: theories and applications: proceedings of SoCTA 2016*. [S.l.]: Springer, 2020. Citado na página 19.

PINHEIRO, T. L. B. Privacy by design como obrigação de segurança no tratamento de dados pessoais. 2021. Citado na página 31.

RAMOS, M. G. O direito à privacidade e a produção de provas no processo penal. 2014. Citado na página 30.

RESNICK, P. et al. Grouplens: An open architecture for collaborative filtering of netnews. In: *Proceedings of the 1994 ACM conference on Computer supported cooperative work*. [S.l.: s.n.], 1994. p. 175–186. Citado na página 16.

RICCI, F.; ROKACH, L.; SHAPIRA, B. Introduction to recommender systems handbook. In: *Recommender systems handbook*. [S.l.]: Springer, 2011. p. 1–35. Citado 3 vezes nas páginas 16, 17 e 20.

ROBERTSON, S.; ZARAGOZA, H. et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, Now Publishers, Inc., v. 3, n. 4, p. 333–389, 2009. Citado na página 19.

SANTOS, A. C. S. Os impactos da literacia digital para a arquivologia contemporânea: a realidade do corpo funcional dos arquivos da cagepa, pbprev e sead. 2023. Citado na página 30.

SANTOS, B. B. S. d. *Uma análise exploratória de dados e o uso de aprendizado de máquina para classificação de doenças cardiovasculares*. Dissertação (B.S. thesis) — Universidade Federal do Rio Grande do Norte, 2022. Citado na página 18.

SANTOS, R. V. dos. Pirataria digital de livros: entre a democratização da leitura e a proteção dos direitos humanos. 2022. Citado na página 31.

SANTOS, W. L. dos. Utilização da tecnologia blockchain no setor público brasileiro. 2022. Citado na página 31.

SILVA, D. A. da. Implementação das políticas públicas de inclusão digital na educação em tempos de pandemia. 2021. Citado 2 vezes nas páginas 30 e 31.

SILVA, J. S. da. Análise das boas práticas de governança corporativa aplicada a uma cooperativa de crédito na paraíba. 2023. Citado na página 30.

SILVA, M. A. da C. A educação feminina e a representação dos perfis normalistas nos periódicos paraibanos: dos relatos às imagens. 2020. Citado na página 30.

SILVA, M. G. da. Aplicação de metodologias ativas para o ensino de ciências/biologia no âmbito do pibid. 2022. Citado na página 27.

SILVA, R. A. N. e. Proteção dos dados pessoais e direito digital: a acepção da privacidade no contexto da lgpd. 2019. Citado 2 vezes nas páginas 30 e 31.

SIRISURIYA, D. S. et al. A comparative study on web scraping. 2015. Citado na página 17.

SOARES, V. H. A. S. Combinações de similaridade semântica e frequência de termos para agrupamento de textos. Universidade Federal de Viçosa, 2017. Citado na página 18.

SOUSA, F. F. de. Modelagem matemática como metodologia de ensino na educação básica. 2019. Citado na página 28.

SOUSA, L. G. de. A implementação de políticas de inclusão digital: uma análise dos telecentros no município de patos - pb. 2015. Citado na página 30.

SOUSA, L. L. A. de. Metodologias ativas no ensino da matemática: o uso de jogos como ferramenta de ensino-aprendizagem. 2020. Citado na página 28.

SOUTO, R. J. da S. O ensino de geografia na perspectiva das metodologias ativas: estratégias para a dinamização do processo de ensino-aprendizagem. 2022. Citado 2 vezes nas páginas 27 e 29.

SOUZA, E. d. S. de; LICHTNOW, D.; GASPARINI, I. Estratégia de pós-processamento aplicada a um sistema de recomendaçao de artigos para a melhora da diversidade. In: SBC. *Anais do XI Brazilian Workshop on Social Network Analysis and Mining*. [S.l.], 2022. p. 216–221. Citado na página 21.

SOUZA, M. de Lourdes Silva de. Análise da aplicabilidade da legislação arquivística no fórum juiz inácio machado de souza. 2014. Citado na página 31.

SUGIYAMA, K.; KAN, M.-Y. Exploiting potential citation papers in scholarly paper recommendation. In: . [S.l.: s.n.], 2013. ISBN 978-1-4503-2077-1. Citado na página 12.

SUKESTIYARNO, Y. L.; SAPOLO, H. A.; SOFYAN, H. Application of recommendation system on e-learning platform using content-based filtering with jaccard similarity and cosine similarity algorithms. Preprints, 2023. Citado na página 21.

ZHAO, B. Web scraping. *Encyclopedia of big data*, Springer Living ed. Cham, v. 1, 2017. Citado na página 17.

A *SCRIPTS* PARA EXTRAÇÃO DOS DADOS

```
def getAllLinkFromAllPages(url):
  currentUrl = url
  page_count = 1
  while currentUrl is not None:
    page = BeautifulSoup(urlopen(currentUrl))
    articles = getLinkArticles(page)

    with open("links_all_uepb.csv", 'a') as csvLinks:
      writer = csv.writer(csvLinks)
      for link in articles:
          writer.writerow([f'{base_url}{link}'])

    linkNextPage = page.find(lambda tag: tag.name == 'a' and 'Proximo'
   in tag.text and not tag.get('disabled'), href=True)

    if linkNextPage is not None:
      linkNextPage['href']
      currentUrl = f'{base_url}'+ linkNextPage['href']
      print(page_count)
      page_count += 1
    else:
      print(f"Finalizou!!! com {page_count} paginas")
      currentUrl = None
```

Listing A.1 – Função de extração dos *links* de todas as páginas.

```
def getAllLinkFromAllPages(url):
  currentUrl = url
  page_count = 1
  while currentUrl is not None:
    page = BeautifulSoup(urlopen(currentUrl))
    articles = getLinkArticles(page)

    with open("links_all_uepb.csv", 'a') as csvLinks:
      writer = csv.writer(csvLinks)
      for link in articles:
          writer.writerow([f'{base_url}{link}'])

    linkNextPage = page.find(lambda tag: tag.name == 'a' and 'Proximo'
   in tag.text and not tag.get('disabled'), href=True)

    if linkNextPage is not None:
      linkNextPage['href']
      currentUrl = f'{base_url}'+ linkNextPage['href']
      print(page_count)
      page_count += 1
```

```
    else:
      print(f"Finalizou!!! com {page_count} paginas")
      currentUrl = None
```

Listing A.2 – Função de extração dos *links* de todas as páginas.

```
class Article:
  def __init__(self, title, author, keywords, date, resume, description,
    uri):
    self.title = title
    self.author = author
    self.keywords = keywords
    self.date= date
    self.resume = resume
    self.description = description
    self.uri = uri

def getDataWithSpecifArticle(url):
    page = BeautifulSoup(urlopen(url))
    data = page.find_all('td', { "class": "metadataFieldValue"})
    title, author, keywords, date, resume, description, uri = [''] * 7
    for i, datum in enumerate(data):
        try:
            value = datum.get_text(separator="\n")
            if i == 0:
                title = value
            elif i == 1:
                author = value
            elif i == 2:
                keywords = value
            elif i == 3:
                date = value
            elif i == 4:
                resume = value
            elif i == 5:
                description = value
            elif i == 6:
                uri = value
        except IndexError:
            pass
    article = Article(title, author, keywords, date, resume, description
   , uri)

    return article
```

Listing A.3 – Função para coleta de dados de um artigo.

```
df = pd.read_csv("links_all_uepb.csv", header=None, names=['link'])

```

```
names_columns = ['title', 'author', 'keywords', 'date', 'resume', '
    description', 'uri']

articlesDB = pd.DataFrame(columns=names_columns)

for index, row in df.iterrows():
    article = getDataWithSpecifArticle(row['link'])

    print(f'{calc_percent(index+1, df.shape[0]):.2f}')

    article_data = {
        'title': article.title,
        'author': article.author,
        'keywords': article.keywords,
        'date': article.date,
        'resume': article.resume,
        'description': article.description,
        'uri': article.uri
    }

    article_df = pd.DataFrame([article_data], columns=names_columns)

    articlesDB = pd.concat([articlesDB, article_df], ignore_index=True)


articlesDB.head()
```

Listing A.4 – Função responsável pela extração de dados de todos os artigos.

```
num_rows_with_empty_values = sum(articlesDB.apply(lambda x: '' in x.
    values, axis=1))

print("Numero de linhas com pelo menos uma coluna vazia:",
    num_rows_with_empty_values)

rows_with_empty_values = articlesDB[articlesDB.apply(lambda x: '' in x.
    values, axis=1)]

# Obter os indices (IDs) das linhas com valores vazios
indices_to_remove = rows_with_empty_values.index

# Remover as linhas com valores vazios do DataFrame
articlesDB = articlesDB.drop(indices_to_remove)

# Verificar o DataFrame atualizado
print(articlesDB)

# Salvar DataFrame em um arquivo CSV
```

```
17 articlesDB.to_csv("articles_all_uepb.csv", index=False)
```

Listing A.5 – Padronização dos dados

B *SCRIPT* PARA TRATAMENTO DOS DADOS COLETADOS

```
import pandas as pd
import spacy

# Carregar modelo em portugues para o spaCy
nlp = spacy.load('pt_core_news_sm')

# Funcao para lematizar e remover stopwords
def preprocess_text(title, keywords, resume):
    # Concatenar titulo, palavras-chave e resumo
    text = f"{title}. {keywords}. {resume}"

    # Processamento com spaCy
    doc = nlp(text)

    # Lematizacao e remocao de stopwords
    tokens = [token.lemma_ for token in doc if not token.is_stop]

    # Juncao dos tokens
    processed_text = ' '.join(tokens)

    # Remover caracteres de escape
    processed_text = processed_text.replace('\r\n', ' ').replace('\n', '
    ')

    return processed_text

# Carregar o arquivo CSV
df = pd.read_csv('articles_all_uepb.csv')

# Aplicar a lematizacao e remocao de stopwords nas colunas de titulo,
   palavras-chave e resumo
df['processed_text'] = df.apply(lambda row: preprocess_text(row['title'
   ], row['keywords'], row['resume']), axis=1)

# Salvar o novo DataFrame em um novo arquivo CSV
df.to_csv('articles_uepb_processed.csv', index=False)
```

Listing B.1 – Lematização e Remoção de *Stopwords*.

C ESTRUTURA DOS ALGORITMOS

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer

# Baixar as stopwords e inicializar o lematizador
import nltk
nltk.download('stopwords')
nltk.download('wordnet')

# Carregar o arquivo CSV processado
df = pd.read_csv('articles_uepb_processed.csv')

# Inicializar o lematizador
lemmatizer = WordNetLemmatizer()

# Lista de stopwords em portugues
stop_words = set(stopwords.words('portuguese'))

# Criar um vetorizador TF-IDF
vectorizer = TfidfVectorizer()

# Aplicar o vetorizador na coluna processada
tfidf_matrix = vectorizer.fit_transform(df['processed_text'])

# Funcao para obter recomendacoes com base em uma string de busca
def get_recommendations(search_string, df, vectorizer, lemmatizer,
    stop_words):
    # Aplicar a lematizacao e remocao de stopwords na string de busca
    search_words = [lemmatizer.lemmatize(word) for word in search_string
    .split() if word.lower() not in stop_words]
    search_string = " ".join(search_words)

    # Aplicar o vetorizador na string de busca
    search_vector = vectorizer.transform([search_string])

    # Calcular a similaridade de cossenos entre a string de busca e os
    artigos
    search_cosine_similarities = cosine_similarity(search_vector,
    tfidf_matrix).flatten()

    # Obter indices dos artigos mais similares
    indices = sorted([i for i in enumerate(search_cosine_similarities)],
    key=lambda i: search_cosine_similarities[i], reverse=True)
```

```
41
42      # Exibir os titulos dos artigos mais similares
43      if indices:
44          for i in indices:
45              print(f"{df['title'][i]}")
46      else:
47          print("Nenhum artigo encontrado com similaridade minima.")
48
49  # Solicitar ao usuario uma string de busca
50  user_input = "Texto para Busca"
51  print("sua busca: ",user_input)
52
53  # Obter e exibir recomendacoes
54  get_recommendations(user_input, df, vectorizer, lemmatizer, stop_words)
```

Listing C.1 – Exemplo da estrutura dos algoritmos.