UNIVERSIDADE FEDERAL DO PARANÁ

ALINE NASCIMENTO XAVIER

ALGORITMOS DE SISTEMAS DE RECOMENDAÇÃO PARA TRABALHOS CIENTÍFICOS

CURITIBA
2023

ALINE NASCIMENTO XAVIER

ALGORITMOS DE SISTEMAS DE RECOMENDAÇÃO PARA TRABALHOS CIENTÍFICOS

Monografia apresentada como requisito parcial à conclusão do curso de graduação em Matemática Industrial, pela Universidade Federal do Paraná.
Orientador: Prof. Dr. Lucas Garcia Pedroso

CURITIBA
2023

Resumo

Nesta monografia abordamos a aplicação de técnicas de Processamento de Linguagem Natural (NLP) na recomendação de trabalhos científicos publicados ao longo dos anos pelos discentes e docentes do Departamento de Matemática da Universidade Federal do Paraná (UFPR). Os modelos Word2Vec, GloVe e BERT foram utilizados para extrair informações semânticas e contextuais dos documentos de texto, com o objetivo de gerar recomendações personalizadas e precisas. A coleta de dados foi realizada por meio de *web scraping*, e os modelos pré-treinados foram ajustados para atender às necessidades da pesquisa. Avaliamos a capacidade dos algoritmos em compreender o contexto dos artigos matemáticos e apresentar recomendações concisas. Os resultados obtidos demonstram que o modelo BERT se destaca nessa tarefa, mostrando uma maior compreensão do contexto e gerando recomendações relevantes. Ao final do trabalho, apresentamos considerações finais e discussões sobre os resultados alcançados. Além disso, oferecemos sugestões para pesquisas futuras, visando aprimorar ainda mais a recomendação de trabalhos científicos.

Palavras-chaves: *Machine Learning. Processamento de Linguagem Natural. Recomendação de Texto. Trabalhos Científicos*

Abstract

This monograph addresses the application of Natural Language Processing (NLP) techniques in the recommendation of scientific papers published over the years by students and faculty members of the Department of Mathematics at the Federal University of Paraná (UFPR). The Word2Vec, GloVe, and BERT models were employed to extract semantic and contextual information from text documents, aiming to generate personalized and accurate recommendations. Data collection was performed through web scraping, and the pre-trained models were adjusted to meet the research requirements. We evaluated the algorithms' ability to comprehend the context of mathematical articles and provide concise recommendations. The results obtained demonstrate the superiority of the BERT model in this task, showcasing a better understanding of the context and generating relevant recommendations. In the final sections of the work, we present concluding remarks, discussions on the achieved results, and suggestions for future research to further enhance the recommendation of scientific papers.

Keywords: *Machine Learning. Natural Language Processing. Text Recommendation. Scientific Papers.*

Sumário

Capítulo 1

Introdução

Nos últimos anos, a área de Processamento de Linguagem Natural (NLP) tem experimentado um crescimento notável em diversas aplicações. Com a proliferação de dados textuais disponíveis em diferentes domínios e a necessidade de extrair informações relevantes dessas vastas quantidades de texto, a NLP se tornou uma área de pesquisa e desenvolvimento vital para a compreensão e interação entre humanos e máquinas.

Uma das aplicações mais promissoras da NLP é a recomendação de texto, que busca oferecer aos usuários informações relevantes e personalizadas, como artigos, notícias, produtos, músicas e filmes. A recomendação de texto tornou-se particularmente importante em um mundo cada vez mais digitalizado e personalizado. Os sistemas de recomendação personalizados fazem uso de abordagens como Filtragem Colaborativa (CF) e Informações baseadas em conteúdo (CB). As abordagens colaborativas constroem modelos com base no comportamento passado de usuários semelhantes para recomendar um item, enquanto as de conteúdo, utilizam metadados, como propriedades, tags e descrições do item para realizar a recomendação. Um modelo que utiliza tanto CF quanto CB é chamado de sistema de recomendação híbrido.

Neste trabalho, concentramos nossa atenção na recomendação de artigos científicos. O objetivo é desenvolver um sistema de recomendação que concentre os artigos publicados pelos docentes quanto as teses e dissertações feitas pelos discentes do Departamento de Matemática da Universidade Federal do Paraná (UFPR). Nosso intuito é criar um mecanismo de pesquisa que nos auxilie a encontrar os trabalhos mais relevantes a partir de um tema qualquer da matemática. Para isso, iremos analisar e comparar os modelos Word2Vec, GloVe e BERT, reconhecidos na área de NLP, em termos de sua capacidade de extrair informações semânticas e contextuais dos artigos de texto e gerar recomendações precisas e personalizadas.

No Capítulo 1, apresentaremos os algoritmos, detalhando sua estrutura e funcionamento. No Capítulo 2, descreveremos a construção do sistema de recomendação, incluindo a extração dos dados e os ajustes dos modelos. Os resultados dos testes realizados serão apresentados e discutidos neste capítulo também, permitindo uma análise comparativa entre os modelos avaliados. Por fim, traremos as considerações finais e discussões.

Capítulo 2

Métodos

Em problemas de classificação e regressão em aprendizagem de máquina, quando temos uma *feature* categórica aplicamos métodos de *encoding* para transformá-la em numérica. Mas mais especificamente quando falamos de processamento de linguagem natural, o processo de converter os dados textuais numa representação numérica são de extrema importância, pois precisam representar a estrutura do texto, que no geral possui uma quantidade expressiva de dados, sem impactar o contexto e considerando a capacidade de memória/computacional. Nesse capítulo, veremos alguns dos métodos mais conhecidos de codificação.

2.1 Bag of Words

Em NLP, o modelo mais simples para transformar dados de texto em numéricos se chama *Bag of Words (BoW)*. Em [9] temos que o método se configura como sendo basicamente a criação de um vocabulário único de *tokens* (palavras) do *corpus* (conjunto de textos/documentos). Em seguida, se constrói um vetor de recursos de cada documento que contenha as contagens de quantas vezes cada palavra do vocabulário ocorre no documento, construindo assim uma matriz onde as colunas são o vocabulário e as linhas representam cada documento, onde o valor é a frequência da palavra no documento. Os valores dos vetores de recursos são chamados de *raw term frequencies* (frequências de termos brutos): $tf(t, d)$ - o número de vezes que o termo t ocorre no documento d. Devemos notar que, no modelo *Bag of Words*, a ordem das palavras em um documento não importa. Outro ponto importante é que como cada documento é um subconjunto do vocabulário, os vetores de recursos consistirão principalmente em zeros, ou seja, a representação é esparsa.

2.2 TF-IDF

Term frequency-inverse document frequency (TF-IDF) é um modelo heurístico que assim como o BoW mede o quão importante uma palavra é em um texto. Normalmente em documentos temos muitos termos redundantes que não agregam ou discriminam informação. Para o algoritmo, palavras que ocorrem em muitos documentos devem receber

um peso menor do que as que aparecem em poucos documentos. O método apresentado tem na proposta diminuir a dimensionalidade do vocabulário sem sacrificar o contexto. Primeiramente definimos $tf\text{-}idf(t, d)$ como o produto da frequência de termos e a frequência inversa do documento,

$$tf\text{-}idf(t, d) = tf(t, d) \cdot idf(t, d),$$

onde $tf(t, d)$ é a frequência de termos t que introduzimos na seção 2.1 e $idf(t, d)$ é a frequência inversa do documento d, que pode ser calculada da como

$$idf(t, d) = \log\left(\frac{n_d}{1 + df(d, t)}\right),$$

em que n_d é o número total de documentos e $df(d, t)$ é o número de documentos que contém o termo t. Adicionar a constante 1 ao denominador serve para atribuir um valor diferente de zero a termos que ocorrem em nenhum dos exemplos de treinamento, o *log* é usado para garantir que palavras com baixa frequências irão receber peso maior.

Esse método é considerado rápido e de fácil implementação, possuindo uma complexidade computacional $O(nn_d)$, onde n é o número de palavras no vocabulário e n_d o número de documentos. Em [13] são apresentadas algumas críticas acerca da técnica TF-IDF, a primeira é que o algoritmo não possui formulação matemática, a segunda é que a dimensionalidade para dados textuais é o tamanho do vocabulário por todo o conjunto de dados, resultando em um enorme cálculo na ponderação de todos esses valores.

2.3 Embedding

Os modelos *BoW* e *TF-IDF* geram ao final um vocabulário fixo de palavras através de vetores de alta dimensionalidade, pois cada palavra distinta representa uma dimensão. Um jeito simples de transformar o vocabulário em *features* é aplicar a técnica *one-hot-enconding*, que transforma os índices do vocabulário numa matriz de uns e zeros, tornando o problema extremamente esparso e de difícil manipulação para um modelo.

Uma solução mais sofisticada é o *embedding*, que é uma técnica usada para representar palavras, frases e outras unidades textuais em vetores densos, onde as novas dimensões capturam propriedades sintáticas e semânticas entre as palavras, de forma que os termos relacionados estejam próximos neste espaço vetorial contínuo.

Os *embeddings* são gerados usando técnicas de aprendizado profundo, como redes neurais, e são frequentemente pré-treinados em grandes *corpus* de texto antes de serem ajustados (*fine-tuning*) para uma tarefa específica.

Existem vários tipos de *embeddings* em NLP, cada um capturando diferentes aspectos dos dados de texto. Os métodos mais comuns incluem *embeddings* de palavras, frases e documentos.

2.3.1 Word2Vec

O *Word2Vec* foi desenvolvido em 2013 por uma equipe de pesquisadores do Google liderada por Tomas Mikolov [11]. O algoritmo mapeia cada palavra do *corpus* em vetores densos através de uma rede neural de três camadas que aprende representações das palavras otimizando uma função objetivo que envolve tanto a palavra *target* quanto as palavras contexto [5]. Existem duas opções de avaliar as representações geradas pelo modelo. A primeira opção é avaliar a distância entre as palavras e encontrar as que estão mais próximas através da similaridade de cossenos, que nos retorna uma pontuação do grau de semelhança semântica entre duas palavras. A segunda opção é avaliar os padrões linguísticos como relações lineares entre vetores de palavras. Em [11] e [14] os autores observam que o modelo consegue de modo aproximado encontrar relações lineares, como por exemplo, vetor("king") - vetor("man") + vetor("woman") é próximo de vetor("queen"), e que essas relações só são possíveis de serem encontradas quando o algoritmo é treinado em grandes conjuntos de dados.

O algoritmo conta com duas modelagens de redes neurais possíveis: *Continuous Bag of Words (CBOW)* e *Skip-gram*, sendo a modelagem de um o inverso do outro, como mostrado na figura 2.1.

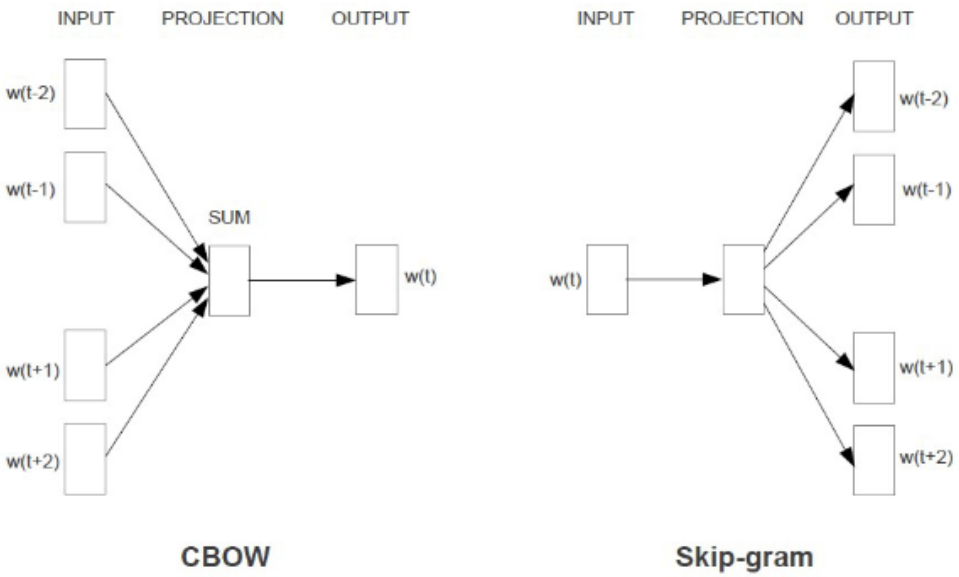

Figura 2.1: CBOW Architectural Models e Skip-Gram

Fonte: [12]

O modelo CBOW aprende *embeddings* prevendo a palavra *target* baseando-se nas palavras de contexto presentes numa determinada janela na vizinhança desta. Por exemplo, considerando a sentença *"deep learning is subset of machine learning"*, com uma janela de contexto de tamanho 2, temos os seguintes pares de (contexto, *target*): ([learning, is], deep),([deep, is, subset], learning), ([deep, learning, subset, of], is) e assim por diante. Sua arquitetura consiste em três camadas, que são conectadas por duas matrizes de pesos W e W', onde W apresenta tamanho $V \times N$, e W' apresenta tamanho $N \times D$, onde V é a dimensão da camada de entrada, N dimensão da camada oculta e D a dimensão da camada de saída. Sendo assim, para uma determinada sequência de palavras $w_1, w_2, w_3, ..., w_v$, a função objetivo do CBOW é maximizar a log-probabilidade média

$$\frac{1}{V}\sum_{i=1}^{V} \log p\left(w_i \middle| w_{i-n}, ..., w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}, ..., w_{i+n}\right),$$

onde n é o tamanho da janela de contexto de treinamento. Um n maior resulta em mais exemplos de treinamento que podem levar a uma maior precisão, às custas de mais tempo de treinamento. A formulação básica do CBOW, define $p(w_{t+j}|w_t)$ usando a função softmax

$$p(w_o|w_i) = \frac{\exp\left(v_{w_i}^T v'_{w_o}\right)}{\sum_{w=1}^{V} \exp\left(v_{w_i}^T v'_w\right)}$$

v_w e v'_w são duas representações vetoriais da palavra w, onde v_w segue das linhas da matriz de pesos de entrada W, e v'_w tem origem na matriz W' de saída. Segundo [12] tal formulação é impraticável pois seu custo computacional é alto. Para contornar tal problema os autores propõem o uso da softmax hierárquica, que não será discutida nesse trabalho.

A outra modelagem do tipo Word2Vec é o Skip-Gram, que funciona de modo exatamente oposto ao CBOW. Nele o objetivo é gerar os *embeddings* predizendo as palavras de contexto dado a palavra *target*. Por exemplo, considerando a sentença "deep learning is subset of machine learning" , com uma janela de contexto de tamanho 2, temos os seguintes pares (contexto, target): ([learning, is], deep), ([deep, is, subset], learning), ([deep, learning, subset, of], is) e assim por diante. A função objetivo do Skip-Gram é mostrada a seguir, e também maximiza a log-probabilidade média

$$\frac{1}{V}\sum_{i=1}^{V} \sum_{-n \leq j \leq n, n \neq 0} \log p\left(w_{i+j} \middle| w_j\right),$$

onde V é o tamanho do vocabulário e n o tamanho da janela.

De acordo com o autor Mikolov [11], ambos modelos apresentam vantagens e desvantagens, por exemplo, o Skip-Gram é mais eficiente com poucos dados de treino e palavras pouco frequentes são bem representadas, enquanto o CBOW é mais rápido e representa bem as palavras frequentes. No entanto, aprender os vetores de saída é o principal problema da formulação apresentada pois é muito custoso, e a fim de contornar essas dificuldades, o autor implementa mais dois algoritmos, chamados de Negative sampling e Hierarchical softmax, que não serão explorados neste trabalho.

2.3.2 GloVe

O algoritmo *Global Vectors* para representações de palavras, conhecido como "GloVe" [3] foi proposto por Pennington et al. em 2014. Ao contrário da proposta do Word2Vec, que avalia apenas a vizinhança da palavra, o GloVe combina informações locais e globais do texto, através do cálculo da matriz co-ocorrência entre as palavras do *corpus*. Segundo [1], GloVe é um modelo baseado em contagem, que aprende a relação do texto, calculando a frequência com que as palavras aparecem umas com as outras. Assim como explicado por [3] e [5], os autores propõem que a taxa de probabilidade de co-ocorrência se dê pelo modelo mais geral 2.1, onde F pode depender de alguns parâmetros não especificados. O número de possibilidades para F segundo os autores é vasto, porém aplicando alguns critérios é possíbel selecionar um escolha única.

$$F\Big(u_i, u_j, v_k\Big) = \frac{P_{ik}}{P_{jk}} \tag{2.1}$$

Seja u_i, v_j palavras focais e v_k vetor de palavras de contexto, e P_{ik} e P_{jk} representam a probabilidade das palavras i e j co-ocorrerem com a palavra k.

Segundo os autores, como os espaços vetoriais são lineares, podemos reescrever a equação 2.1, como

$$F\Big(dot\Big(u_i - u_j, v_k\Big)\Big) = \frac{P_{ik}}{P_{jk}}$$

pois um palavra e seu contexto são intercambiáveis na matriz de co-ocorrência. A fim de manter a simetria, podemos introduzir vieses b_i e b_k e reescrever o modelo como

$$u_i^T u_k + b_i + b_k = log(X_{ik}),$$

onde X_{ik} representa a frequência de co-ocorrência entre as palavras i e k. Os autores em [3] propõem um novo modelo de regressão de quadrados mínimos ponderados

$$J = \sum_{i,k=1}^{V} f(X_{ik})(u_i^T v_k + b_i + b_k - log(X_{ik}))^2,$$

sendo V o tamanho do vocabulário e $f(x)$ uma função ponderadora incluída na função objetivo. Para que co-ocorrências raras ou frequentes não sejam superponderadas, f é definida como

$$f(x) = \begin{cases} (x/x_{max})^{\alpha}, \text{se } x \leq x_{max} \\ 1, \text{caso contrário.} \end{cases}$$

Os autores usaram gradiente descendente estocástico, com $x_{max} = 100$ e $\alpha = 0.75$ para treinar o modelo.

2.3.3 BERT

BERT (*Bidirectional Encoder Representations from Transformers*) é um modelo de NLP pré-treinado em dados não rotulados, considerado o "estado da arte" para muitas tarefas, como perguntas e respostas e inferência de linguagem. O algoritmo foi desenvolvido por um grupo de pesquisadores do Google em 2018 [4], e é utilizado em 11 das tarefas de linguagem mais comuns, como análise de sentimento, geração de texto e sumarização. Os autores basearam-se no campo da visão computacional, onde o conceito de *transfer learning* é amplamente utilizado para pré-treinar um modelo para uma tarefa ampla, e depois fazer o *fine-tuning*, onde se utiliza o modelo treinado como base na geração de um novo modelo específico. Para tarefas em NLP, os autores citam o modelo ELMO que exemplifica como uma rede neural pré-treinada produz *embeddings* de palavras que podem ser usados como *features* para outras tarefas. O grande diferencial desse método é a quantidade de dados utilizados para treinamento que diretamente contribui para o entendimento amplo da linguagem, foram 3.3 bilhões de palavras provindas do Wikipedia e Google's BooksCorpus.

Na primeira parte do algoritmo é realizado o pré-treinamento. Os autores inovaram propondo uma abordagem bidirecional oposta às abordagens conhecidas até aquele momento. Nelas a direção de treinamento era direita-para-esquerda ou esquerda-para-direita, ou seja o modelo olhava apenas para uma direção sequencialmente. No BERT a direção não é considerada, o algoritmo recebe a sequência completa de palavras possibilitando apreender o contexto da palavra baseando-se em toda a sua vizinhança. De forma simplificada, nessa etapa são realizados dois passos, no primeiro chamado de *Masked LM*, o método aleatoriamente escolhe 15% das palavras de cada sentença e substitui por uma máscara, e então tenta prever o valor inicial do termo mascarado a partir das palavras em volta que não foram ocultadas (contexto). No segundo passo, chamado de *Next Sentence Prediction (NSP)*, o algoritmo procura entender a relação entre duas sentenças, e retorna de forma binarizada se a sentença B é ou não sequência da sentença A. Para tornar o pré-treino ágil, os autores utilizaram-se de *Transformers*, algoritmo de aprendizagem profunda, que possui um mecanismo chamado *self-attention*, que possibilita o modelo focar apenas no conteúdo importante e não desperdiçar recursos computacionais processando informações irrelevantes.

A segunda parte do algoritmo refere-se ao fine-tuning, onde utilizando-se dos parâmetros pré-treinados é possível modelar várias tarefas de NLP. Para tal, basta adicionar a tarefa específica no BERT e recalcular todos os parâmetros de ponta a ponta. Comparado com o pré-treinamento o fine-tuning é uma etapa rápida.

Capítulo 3

Experimentos

Neste capítulo, iremos abordar todas as etapas envolvidas na construção do nosso sistema de recomendação de artigos científicos. Exploraremos os métodos utilizados para obtenção e tratamento dos dados, desde a extração dos dados brutos até a preparação desses dados em um formato adequado para ferramentas de aprendizado de máquina. Além disso, faremos comentários sobre a implementação dos algoritmos apresentados no capítulo anterior e os resultados obtidos nos testes.

3.1 Extração dos Dados

Com o objetivo de criar uma base de dados que contivesse os artigos científicos publicados pelos professores, teses e dissertações elaboradas pelos alunos da pós-graduação do Departamento de Matemática (DMAT) da UFPR ao longo dos anos, identificamos que o próprio site do DMAT continha os trabalhos realizados pelos alunos da pós-graduação, totalizando 124 trabalhos datados desde 2006.

Para encontrar os artigos publicados pelos professores do departamento, inicialmente pesquisamos os currículos Lattes no site do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), que listam as publicações. No entanto, para acessar os artigos completos, era necessário ser redirecionado para o site onde ocorreu a publicação original, geralmente em uma revista científica.

Durante a coleta de dados, notamos que havia uma grande quantidade de artigos para cada professor em sites diferentes, tornando a extração desses dados manualmente uma tarefa muito onerosa. Portanto, decidimos buscar uma fonte única que contivesse os artigos, e após algumas análises, optamos por utilizar o Google Scholar, pois continha todas as informações necessárias para a construção do *dataset*. A Figura 3.1 mostra o perfil de um professor do DMAT na plataforma, juntamente com todos os artigos relacionados a ele. No entanto, o *Google Scholar* consulta diferentes fontes para listar os artigos, e alguns deles não são científicos. Outro problema encontrado foi que nem todos os professores do departamento de matemática possuíam perfil no *Google Scholar*, pois é necessário um cadastro prévio. Dos 53 docentes ativos, apenas 17 possuíam perfil.

Apesar das ressalvas mencionadas em relação ao *Google Scholar*, essa foi a fonte que

continha a maior quantidade de informações compiladas e estruturadas sobre os artigos científicos. A partir da fonte escolhida, foi desenvolvido um *script* que coletasse os dados de maneira automatizada por meio de *web scraping* dos artigos contidos nas páginas dos professores.

Durante o processo de *scraping*, enfrentamos alguns problemas relacionados ao *Google*, uma vez que o *script* realizava muitas requisições ao site e, muitas vezes, essas requisições eram bloqueadas devido à percepção de atividade de robô. No entanto, ao final do processo, conseguimos coletar um total de 178 artigos publicados pelos docentes.

Figura 3.1: Perfil de um docente no Google Scholar.

As teses e dissertações publicadas pelos alunos, disponíveis no site do DMAT, também foram extraídos por meio de um *script*. Nesse caso, não enfrentamos problemas com as requisições. No entanto, os dados não estavam completamente estruturados. Algumas páginas de artigos não continham resumo, autor ou as informações estavam desordenadas, principalmente nos trabalhos publicados há muito tempo. Por esse motivo, foi necessário inserir manualmente muitos dados após o *script*.

Após a extração dos dados criamos nosso *dataset* com um total de 302 trabalhos científicos. As informações coletadas incluem autor, título, resumo, data, jornal publicado e orientador.

3.2 Limpeza dos Dados

Com o conjunto de dados criado, agora podemos prosseguir para a etapa de limpeza e tratamento dos dados. Neste trabalho, daremos ênfase à manipulação dos resumos escritos em inglês, uma vez que essa língua oferece excelentes ferramentas de análise semântica amplamente reconhecidas. Além disso, a maioria dos artigos publicados pelos professores

está redigida em inglês.

Para preparar os dados extraídos e torná-los adequados para utilização em modelos, é necessário realizar algumas etapas importantes. O primeiro passo consiste na remoção de caracteres indesejados, como pontuação e dígitos numéricos, além de padronizar o texto, convertendo todas as letras para minúsculas.

Outra técnica comumente utilizada nesse contexto é o *stemming*, que reduz uma palavra à sua forma original. Isso nos permite relacionar palavras semelhantes e reduzir a dimensionalidade do vocabulário. Por exemplo, o *stemming* identifica que *"computing", "computer"* e *"computers"* possuem a mesma raiz, *"compute"*. No entanto, é importante ressaltar que essa técnica reduz as palavras à forma raiz sem considerar o significado da palavra gerada, como no caso da palavra "thu"derivada de "thus". Para contornar esse tipo de problema, podemos utilizar a técnica de lematização, que é semelhante ao *stemming*, mas mais sofisticada e complexa computacionalmente. A lematização nos permite obter a forma canônica da palavra gramaticalmente correta.

Por fim, utilizaremos mais uma técnica para reduzir a dimensionalidade do vocabulário: as *stop words*. Essas são palavras comuns na linguagem que geralmente são filtradas ou removidas durante o processamento de linguagem natural (NLP), pois não possuem um significado importante e podem introduzir ruídos. Exemplos de stop words em inglês são *"a", "an", "the", "and", "but", "is"*.

O próximo passo consiste na divisão do texto de cada documento em elementos individuais, chamados *tokens*. Uma maneira simples de realizar a tokenização é separar o documento nos espaços em branco entre as palavras. Por exemplo a frase *"A runner likes running and runs a LOT!!!"* aplicando a limpeza, lematização e a tokenização temos como resultado o vetor [*"runner", "like", "run", "run", "lot"*].

3.3 Metodologia

Após a limpeza e tokenização dos dados, os métodos discutidos no capítulo anterior foram aplicados. Neste trabalho, foram utilizados os algoritmos Word2Vec, GloVe e BERT para a realização dos testes. O objetivo principal é obter as melhores representações das palavras, visando a construção de um sistema de recomendação de artigos científicos. Para alcançar esse objetivo, é crucial que esses algoritmos sejam treinados em grandes *corpus* de textos, permitindo assim a aprendizagem do contexto e da semântica das palavras. No entanto, devido à limitação de tempo e recursos computacionais, optou-se por utilizar modelos pré-treinados disponibilizados por grupos de pesquisadores renomados. Esses modelos são treinados em dados genéricos não rotulados coletados da internet. Por exemplo, o modelo Word2Vec foi treinado em um *corpus* contendo 100 bilhões de palavras, fornecendo um *embedding* de 3 milhões de palavras e frases representadas em um vetor de dimensão 300. Os detalhes dos modelos pré-treinados utilizados neste estudo estão apresentados na tabela 3.1.

Nome	Modelo	Corpus	Dimensão
word2vec-google-news-300	Word2Vec	Google News Corpus	300
GloVe-wiki-gigaword-300	GloVe	Wikipedia 2014 + Gigaword 5	300
all-mpnet-base-v2	BERT	Vários	768

Tabela 3.1: Modelos pré-treinados.

Queremos obter as melhores recomendações para os artigos científicos, o que requer que esses algoritmos realizem o *embedding* (mapeamento das palavras para vetores densos) de forma ideal, preservando o contexto e a semântica. Para atingir esse objetivo, faremos uso de modelos pré-treinados disponibilizados por grupos de pesquisadores que possuem os recursos computacionais e o tempo necessários para gerar esses *embeddings*. Os modelos pré-treinados Word2Vec e GloVe foram obtidos através da biblioteca Gensim [7], e o BERT da HuggingFace [10]. Tanto os *scrpts* criados para o *web scrapping*, quanto os modelos e testes utilizados nessa monografia foram desenvolvidos na liguagem de programação Python.

O pré-treinamento em grandes volumes de texto proporciona aos modelos um aprendizado amplo, porém limitado quando se trata de tarefas com palavras de domínios específicos. Modelos pré-treinados geralmente têm dificuldades em incorporar adequadamente essas palavras ou até mesmo não as consideram, o que é o caso ao lidarmos com artigos científicos na área da matemática. Nesse domínio, existem muitas palavras reservadas e com significados diferentes do uso comum. Esses fatores restringem o uso direto de modelos pré-treinados, mas podemos contornar essas limitações por meio do fine-tuning, uma técnica que nos permite adicionar informações específicas do domínio aos modelos pré-treinados, reduzindo a lacuna nesses campos. Neste trabalho usaremos o *dataset* contendo os artigos científicos coletados e tratados. Durante um teste realizado com o Word2Vec pré-treinado, encontramos várias palavras comuns na matemática que não estavam presentes no vocabulário, como “lagrangian”, “stationarity”, “semigroup”e “polynomially”.

Conforme discutido no capítulo anterior, o BERT foi desenvolvido com a capacidade de realizar fine-tuning em tarefas específicas. Em nossa implementação, não tivemos dificuldades para adicionar o *dataset* criado a partir dos artigos e recalcular os *embedding*s. Por outro lado, para os modelos Word2Vec e GloVe, a arquitetura permite apenas a alteração dos pesos das palavras presentes na interseção entre o vocabulário do modelo e o *dataset*, ignorando palavras que não estão contidas no vocabulário pré-treinado.

Após a realização do fine-tuning com o conjunto de dados, obtemos um modelo final para cada algoritmo, capaz de compreender o contexto e a semântica de artigos científicos na área da matemática. Com o modelo pronto, podemos realizar a recomendação de artigos com base em uma nova frase ou texto relacionado ao domínio da matemática. Nosso objetivo é encontrar o artigo científico do *dataset* que seja mais semelhante à nova frase. Para isso, utilizamos o modelo para fazer o *embedding* da nova frase e, em seguida, comparamos com os *embedding*s dos resumos dos artigos científicos em nosso *dataset*. Utilizamos a similaridade de cossenos como medida para realizar essa comparação, pois é

amplamente utilizada em problemas de recomendação de texto.

A similaridade de cossenos mede a similaridade entre dois documentos, independentemente de seus tamanhos. Matematicamente, cada documento é projetado como um vetor em um espaço multidimensional e, em seguida, calculamos o ângulo do cosseno entre esses dois vetores. A fórmula nos fornece um valor no intervalo de [-1, 1]. No nosso caso específico, o intervalo é de [0, 1] porque todos os vetores são positivos. Quanto mais próximo de 1 for o resultado, maior será a similaridade entre os vetores. Embora essa medida nos ajude a encontrar os vetores mais similares dentro do espaço, ela não nos fornece informações sobre a qualidade da recomendação. Além disso, como é calculada em cada espaço gerado por um modelo específico, comparar os resultados entre diferentes modelos seria incorreto.

No estudo [2], os autores mencionam a dificuldade de validar modelos de similaridade devido à falta de conjuntos de validação ou referências confiáveis. O objetivo desse estudo é propor um novo algoritmo baseado no BERT capaz de realizar a recomendação de vinhos com base em um conjunto de dados de descrições. Para comparar o desempenho do algoritmo proposto com outros métodos reconhecidos, os autores precisaram criar um conjunto de dados de validação anotados por um profissional de vinhos. Somente assim conseguiram avaliar e medir a qualidade da recomendação oferecida pelo seu algoritmo.

Para esta monografia, não dispomos de um *dataset* de validação criado por um especialista em matemática ou de um método quantitativo para avaliar a qualidade das recomendações entre os modelos. Portanto, as observações apresentadas na próxima seção são de natureza qualitativa e baseadas apenas na compreensão do autor.

Por fim, a figura 3.2 mostra a arquitetura desse projeto.

Figura 3.2: Diagrama do fluxo do projeto.

3.4 Testes

Nesta seção, apresentamos uma análise das principais recomendações de artigos científicos gerados pelos modelos Word2Vec, GloVe e BERT para o campo da matemática, com base em uma nova sentença/frase. Nosso objetivo é explorar a capacidade desses algoritmos em identificar e sugerir artigos relevantes nesse campo de estudo.

Considerando as top 5 recomendações de cada algoritmo, construímos uma tabela com informações relevantes, incluindo ID para identificação dos artigos, título, autor, fonte (para distinguir trabalhos realizados por discentes e docentes), score obtido na similaridade de cossenos e as palavras em comum entre o resumo do artigo e a frase sugerida. É importante destacar que todo o trabalho foi baseado nos resumos dos artigos, mas não incluímos essa informação na tabela devido ao tamanho extenso dos resumos.

Essas recomendações serão avaliadas de forma subjetiva, levando em consideração nossas percepções e análises. O objetivo é iniciar uma discussão a partir dessas recomendações e explorar as possíveis conexões entre a frase sugerida e os artigos recomendados. Essas considerações subjetivas nos ajudarão a compreender a capacidade dos modelos em identificar trabalhos relevantes na área da matemática, além de fornecer insights sobre a eficácia dos algoritmos em tarefas de recomendação.

Neste primeiro teste, buscamos os artigos mais semelhantes com base na frase “theoretical aspects of Augmented Lagrangian methods”. As tabelas a seguir apresentarão

as recomendações geradas pelos modelos Word2Vec, GloVe e BERT, juntamente com as informações relevantes dos artigos recomendados.

Word2Vec		
1º	ID:	162
	Título:	Numerical modeling of mechanical wave propagation
	Autor:	G Seriani, SP Oliveira
	Fonte:	Artigo
	Score:	0.694
	Palavras:	[methods]
2º	ID:	107
	Título:	Scalar wave propagation in 2D: a BEM formulation based on the operational quadrature method
	Autor:	Ana Ibis Abreu, Jose Antonio Marques Carrer, Webe João Mansur
	Fonte:	Artigo
	Score:	0.683
	Palavras:	[ ]
3º	ID:	276
	Título:	Um algoritmo de filtro globalmente convergente sem derivadas da função objetivo para otimização restrita e algoritmos de pivotamento em blocos principais para problemas de complementaridade linear
	Autor:	Priscila Savulski Ferreira
	Fonte:	Tese/Dissertação
	Score:	0.68
	Palavras:	[methods]
4º	ID:	157
	Título:	Weighted quadrature rules for finite element methods
	Autor:	Saulo P Oliveira, Alexandre L Madureira, Frederic Valentin
	Fonte:	Artigo
	Score:	0.68
	Palavras:	[ ]
5º	ID:	268
	Título:	Métodos de Gauss-Newton para problemas de qualidade mínimos não lineares : teoria, validação numérica e aplicação em Geofísica
	Autor:	Monique Bonfim de Souza
	Fonte:	Tese/Dissertação
	Score:	0.675
	Palavras:	[theoretical, methods]

Tabela 3.2: Exemplo 1 - Word2Vec.

GloVe		
1º	ID:	162
	Título:	Numerical modeling of mechanical wave propagation
	Autor:	G Seriani, SP Oliveira
	Fonte:	Artigo
	Score:	0.799
	Palavras:	[methods]
2º	ID:	241
	Título:	Materiais manipuláveis e recursos digitais no ensino de trigonometria
	Autor:	Paulo César Tavares de Souza
	Fonte:	Tese/Dissertação
	Score:	0.797
	Palavras:	[ ]
3º	ID:	29
	Título:	The definability of physical concepts - doi: 10.5269/bspm.v23i1-2.7471
	Autor:	Adonai S Sant'Anna
	Fonte:	Artigo
	Score:	0.796
	Palavras:	[theoretical]
4º	ID:	58
	Título:	Oral History in Mathematics Education: an overview
	Autor:	Antonio Vicente Marafioti Garnica, Carlos Roberto Vianna
	Fonte:	Artigo
	Score:	0.793
	Palavras:	[ ]
5º	ID:	126
	Título:	Scalar wave equation by the boundary element method: a D-BEM approach with non-homogeneous initial conditions
	Autor:	JAM Carrer, WJ Mansur, RJ2487012 Vanzuit
	Fonte:	Artigo
	Score:	0.7934
	Palavras:	[ ]

Tabela 3.3: Exemplo 1 - GloVe.

BERT		
1º	ID:	9
	Título:	A new family of penalties for augmented Lagrangian methods
	Autor:	Luiz Carlos Matioli, Clóvis C Gonzaga
	Fonte:	Artigo
	Score:	0.641
	Palavras:	[Lagrangian, methods]
2º	ID:	280
	Título:	Uma classe de métodos de lagrangiano aumentado
	Autor:	Elvis Manuel Rodriguez Torrealba
	Fonte:	Tese/Dissertação
	Score:	0.63
	Palavras:	[Augmented, Lagrangian, methods]
3º	ID:	256
	Título:	Algoritmos baseados em métodos de lagrangeano aumentado para problemas de equilíbrio
	Autor:	Elvis Manuel - Rodriguez Torrealba
	Fonte:	Tese/Dissertação
	Score:	0.609
	Palavras:	[Augmented, Lagrangian, methods]
4º	ID:	4
	Título:	Programaçao nao linear sem derivadas
	Autor:	Lucas Garcia Pedroso
	Fonte:	Artigo
	Score:	0.594
	Palavras:	[theoretical, Augmented, Lagrangian]
5º	ID:	18
	Título:	A unified approach to multiplier and proximal methods
	Autor:	Rómulo Castillo, Clóvis C Gonzaga, Elizabeth W Karas, Luiz C Matioli
	Fonte:	Artigo
	Score:	0.583
	Palavras:	[Lagrangian, methods]

Tabela 3.4: Exemplo 1 - BERT.

Podemos perceber que, para os algoritmos Word2Vec e GloVe, não houve recomendações de artigos ou resumos que contivessem o termo "Augmented Lagrangian". Ou seja, não houve uma recomendação explícita com base nesse termo. No entanto, podemos observar que, para o Word2Vec 3.2, com exceção da recomendação do artigo 157 que aborda métodos para elementos finitos, as outras recomendações aparentemente podem fazer sentido. Por exemplo, há artigos sobre ondas, embora o resumo não mencione explicitamente o conteúdo do Lagrangiano aumentado, é possível que o artigo aborde o assunto em seu conteúdo. A mesma observação se aplica ao artigo 276, que fala sobre algoritmos de otimização.

Para o GloVe 3.3, não temos nenhuma recomendação explícita com base nos termos

da nossa sentença. O algoritmo também apresenta duas recomendações de artigos sobre ondas, seguindo as mesmas considerações apresentadas para o Word2Vec. No entanto, o GloVe também apresenta 3 recomendações que aparentemente não fazem sentido, como sugerir um artigo sobre a história oral da educação matemática ou sobre materiais para ensino de trigonometria.

Já para o BERT 3.4, temos 5 recomendações explícitas com base no termo principal da nossa sentença, e todas as sugestões fazem sentido. Neste teste, pudemos ver que no *dataset* havia artigos com o termo "Augmented Lagrangian"e apenas o BERT foi capaz de fazer a recomendação mais precisa, pois identificou o termo principal dentro da frase fornecida, enquanto o GloVe e Word2Vec não foram capazes de identificar adequadamente. Um detalhe interessante é que o GloVe apresentou a maior similaridade de cossenos, mas forneceu a pior recomendação.

A biblioteca que utilizamos para o GloVe e o Word2Vec oferece um recurso onde podemos verificar quais outras palavras no *corpus* são mais semelhantes a uma determinada palavra. As tabelas 3.5 e 3.6 mostram as 10 principais palavras mais semelhantes a "Lagrangian". As palavras apresentadas para o Word2Vec não fazem sentido, enquanto as do GloVe mostram algum relacionamento com o campo da matemática. O BERT não possui esse recurso, pois gera *embeddings* de acordo com o contexto da palavra. Por exemplo, a palavra "banco"pode ser interpretada como um objeto ou uma instituição financeira.

	Palavra	Score
1	Hanif_Mohammad	0.303
2	Pony_Express_riders	0.272
3	Ronjan	0.27
4	Flyingbolt	0.268
5	Lala_Amarnath	0.267
6	contracting_spinal_meningitis	0.266
7	Susanthika	0.264
8	Tendulkar_Laxman	0.264
9	Pandit_ji	0.262
10	Ramesh_Tendulkar	0.26

Tabela 3.5: Exemplo 1 - Palavras Similares Word2Vec.

	Palavra	Score
1	hamiltonian	0.586
2	variational	0.508
3	invariant	0.49
4	formula_21	0.484
5	equations	0.481
6	formula_18	0.479
7	formula_19	0.477
8	formula_1	0.476
9	dynamical	0.472
10	lagrange	0.47

Tabela 3.6: Exemplo 1 - Palavras Similares GloVe.

No segundo teste, buscamos as recomendações para a frase "penalization based optimization methods for constrained problems". A seguir, estão as tabelas com os resultados.

Word2Vec		
1º	ID:	202
	Título:	Nonmonotone line-search methods for convexly constrained multi-objective optimization problems
	Autor:	Maria Eduarda Pinheiro
	Fonte:	Tese/Dissertação
	Score:	0.778
	Palavras:	[optimization, methods, constrained, problems]
2º	ID:	279
	Título:	Três contribuições em otimização não-linear e não-convexa
	Autor:	Geovani Nunes Grapiglia
	Fonte:	Tese/Dissertação
	Score:	0.757
	Palavras:	[optimization, constrained, problems]
3º	ID:	7
	Título:	New optimization algorithms for structural reliability analysis
	Autor:	SRd Santos, LC Matioli, AT Beck
	Fonte:	Artigo
	Score:	0.745
	Palavras:	[based, optimization, methods, problems]
4º	ID:	256
	Título:	Algoritmos baseados em métodos de lagrangeano aumentado para problemas de equilíbrio
	Autor:	Elvis Manuel - Rodriguez Torrealba
	Fonte:	Tese/Dissertação
	Score:	0.725
	Palavras:	[based, methods, constrained, problems]
5º	ID:	4
	Título:	Programaçao nao linear sem derivadas
	Autor:	Lucas Garcia Pedroso
	Fonte:	Artigo
	Score:	0.722
	Palavras:	[based, optimization]

Tabela 3.7: Exemplo 2 - Word2Vec.

GloVe		
1º	ID:	278
	Título:	Um estudo de buscas unidirecionais aplicadas ao método BFGS
	Autor:	Diego Manoel Panonceli
	Fonte:	Tese/Dissertação
	Score:	0.893
	Palavras:	[based, optimization, methods, constrained, problems]
2º	ID:	202
	Título:	Nonmonotone line-search methods for convexly constrained multi-objective optimization problems
	Autor:	Maria Eduarda Pinheiro
	Fonte:	Tese/Dissertação
	Score:	0.892
	Palavras:	[optimization, methods, constrained, problems]
3º	ID:	279
	Título:	Três contribuições em otimização não-linear e não-convexa
	Autor:	Geovani Nunes Grapiglia
	Fonte:	Tese/Dissertação
	Score:	0.884
	Palavras:	[optimization, constrained, problems]
4º	ID:	198
	Título:	Algoritmos de lagrangiano aumentado aplicados ao problema não linear contínuo de alocação de recursos
	Autor:	Juliana Gomes da Silva
	Fonte:	Tese/Dissertação
	Score:	0.881
	Palavras:	[problems]
5º	ID:	7
	Título:	New optimization algorithms for structural reliability analysis
	Autor:	SRd Santos, LC Matioli, AT Beck
	Fonte:	Artigo
	Score:	0.880
	Palavras:	[based, optimization, methods, problems]

Tabela 3.8: Exemplo 2 - GloVe.

BERT		
1º	ID:	197
	Título:	A função da penalidade exata e algumas suavizações : aspectos teóricos e computacionais
	Autor:	Mariana da Rosa
	Fonte:	Tese/Dissertação
	Score:	0.715
	Palavras:	[based, optimization, methods, constrained, problems]
2º	ID:	280
	Título:	Uma classe de métodos de lagrangiano aumentado
	Autor:	Elvis Manuel Rodriguez Torrealba
	Fonte:	Tese/Dissertação
	Score:	0.675
	Palavras:	[methods, constrained, problems]
3º	ID:	256
	Título:	Algoritmos baseados em métodos de lagrangeano aumentado para problemas de equilíbrio
	Autor:	Elvis Manuel - Rodriguez Torrealba
	Fonte:	Tese/Dissertação
	Score:	0.649
	Palavras:	[based, methods, constrained, problems]
4º	ID:	181
	Título:	A bundle-filter method for nonsmooth convex constrained optimization
	Autor:	Elizabeth Karas, Ademir Ribeiro, Claudia Sagastizábal, Mikhail Solodov
	Fonte:	Artigo
	Score:	0.629
	Palavras:	[optimization, methods, constrained, problems]
5º	ID:	18
	Título:	A unified approach to multiplier and proximal methods
	Autor:	Rómulo Castillo, Clóvis C Gonzaga, Elizabeth W Karas, Luiz C Matioli
	Fonte:	Artigo
	Score:	0.607
	Palavras:	[methods, problems]

Tabela 3.9: Exemplo 2 - BERT.

Neste teste, os três algoritmos apresentaram recomendações de artigos relacionados ao tema pesquisado. Observou-se que os algoritmos Word2Vec e GloVe não tiveram dificuldades em realizar as recomendações, sugerindo que esses algoritmos tendem a ter um melhor desempenho em palavras menos específicas, diferentes de "Lagrangian"por exemplo.

Por fim, no último teste, apresentamos os artigos recomendados para o tema "machine learning".

Word2Vec		
1º	ID:	241
	Título:	Materiais manipuláveis e recursos digitais no ensino de trigonometria
	Autor:	Paulo César Tavares de Souza
	Fonte:	Tese/Dissertação
	Score:	0.50
	Palavras:	[learning]
2º	ID:	250
	Título:	Métodos de busca direta para seleção de parâmetros em máquinas de vetores suporte
	Autor:	Natalha Cristina da Cruz Machado Benatti
	Fonte:	Tese/Dissertação
	Score:	0.46
	Palavras:	[machine, learning]
3º	ID:	263
	Título:	Aprendizagem de máquina aplicada à previsão dos movimentos do Ibovespa
	Autor:	Aline Cristiane Finkler
	Fonte:	Tese/Dissertação
	Score:	0.443
	Palavras:	[machine, learning]
4º	ID:	17
	Título:	Modular knight distance in graphs and applications on the n-queens problem
	Autor:	Oliver Kolossoski, Luiz Carlos Matioli, Elvis MR Torrealba, Juliana G Silva
	Fonte:	Artigo
	Score:	0.43
	Palavras:	[ ]
5º	ID:	252
	Título:	Fenômenos cíclicos : modelagem com funções trigonométricas : parte II
	Autor:	Luiz Fabiano Effgen
	Fonte:	Tese/Dissertação
	Score:	0.42
	Palavras:	[learning]

Tabela 3.10: Exemplo 3 - Word2Vec.

GloVe		
1º	ID:	241
	Título:	Materiais manipuláveis e recursos digitais no ensino de trigonometria
	Autor:	Paulo César Tavares de Souza
	Fonte:	Tese/Dissertação
	Score:	0.673
	Palavras:	[learning]
2º	ID:	263
	Título:	Aprendizagem de máquina aplicada à previsão dos movimentos do Ibovespa
	Autor:	Aline Cristiane Finkler
	Fonte:	Tese/Dissertação
	Score:	0.625
	Palavras:	[machine, learning]
3º	ID:	250
	Título:	Métodos de busca direta para seleção de parâmetros em máquinas de vetores suporte
	Autor:	Natalha Cristina da Cruz Machado Benatti
	Fonte:	Tese/Dissertação
	Score:	0.624
	Palavras:	[machine, learning]
4º	ID:	222
	Título:	A study about Lq-norm least squares support vector machine with feature selection
	Autor:	Felipe de Jesus Kutz
	Fonte:	Tese/Dissertação
	Score:	0.624
	Palavras:	[machine]
5º	ID:	128
	Título:	Brazildam: A benchmark dataset for tailings dam detection
	Autor:	Edemir Ferreira, Matheus Brito, Remis Balaniuk, Mário S Alvim, Jefersson A dos Santos
	Fonte:	Artigo
	Score:	0.611
	Palavras:	[machine, learning]

Tabela 3.11: Exemplo 3 - GloVe.

As tabelas apresentadas mostram que os algoritmos Word2Vec e GloVe não conseguiram compreender completamente que o termo buscado estava relacionado à área de aprendizado de máquina. Em ambos os casos, o artigo mais similar foi um trabalho sobre ensino de trigonometria, indicando que os algoritmos deram mais peso à palavra "learning".

A tabela 3.12 contém as recomendações para a frase "machine learning", e a tabela 3.13 para a frase "Artificial Intelligence". Em ambos os casos, o BERT apresentou artigos

coerentes. Pode-se observar que os artigos 222, 250, 286 e 263 coincidem nas duas recomendações, o que mostra que o algoritmo entende que as frases de busca pertencem ao mesmo contexto. Também é válido ressaltar que em ambas as recomendações houveram sugestões de artigos que não continham as palavras de busca, demonstrando mais uma vez que o BERT é capaz de compreender o contexto do texto.

BERT - Machine Learning		
1º	ID:	222
	Título:	A study about Lq-norm least squares support vector machine with feature selection
	Autor:	Felipe de Jesus Kutz
	Fonte:	Tese/Dissertação
	Score:	0.503
	Palavras:	[machine]
2º	ID:	250
	Título:	Métodos de busca direta para seleção de parâmetros em máquinas de vetores suporte
	Autor:	Natalha Cristina da Cruz Machado Benatti
	Fonte:	Tese/Dissertação
	Score:	0.455
	Palavras:	[machine, learning]
3º	ID:	286
	Título:	Sobre o uso de regressão por vetores suporte para a construção de modelos em um método de região de confiança sem derivadas
	Autor:	Adriano Verdério
	Fonte:	Tese/Dissertação
	Score:	0.417
	Palavras:	[machine, learning]
4º	ID:	263
	Título:	Aprendizagem de máquina aplicada à previsão dos movimentos do Ibovespa
	Autor:	Aline Cristiane Finkler
	Fonte:	Tese/Dissertação
	Score:	0.377
	Palavras:	[machine, learning]
5º	ID:	262
	Título:	Análise teórica de máquinas de vetores suporte e aplicação a classificação de caracteres
	Autor:	Evelin Heringer Manoel Krulikovski
	Fonte:	Tese/Dissertação
	Score:	0.362
	Palavras:	[ ]

Tabela 3.12: Exemplo 3 - BERT, Machine Learning.

BERT - Artificial Intelligence		
1º	ID:	222
	Título:	A study about Lq-norm least squares support vector machine with feature selection
	Autor:	Felipe de Jesus Kutz
	Fonte:	Tese/Dissertação
	Score:	0.277
	Palavras:	[ ]
2º	ID:	263
	Título:	Aprendizagem de máquina aplicada à previsão dos movimentos do Ibovespa
	Autor:	Aline Cristiane Finkler
	Fonte:	Tese/Dissertação
	Score:	0.255
	Palavras:	[artificial]
3º	ID:	286
	Título:	Sobre o uso de regressão por vetores suporte para a construção de modelos em um método de região de confiança sem derivadas
	Autor:	Adriano Verdério
	Fonte:	Tese/Dissertação
	Score:	0.238
	Palavras:	[ ]
4º	ID:	241
	Título:	Materiais manipuláveis e recursos digitais no ensino de trigonometria
	Autor:	Paulo César Tavares de Souza
	Fonte:	Tese/Dissertação
	Score:	0.228
	Palavras:	[ ]
5º	ID:	250
	Título:	Métodos de busca direta para seleção de parâmetros em máquinas de vetores suporte
	Autor:	Natalha Cristina da Cruz Machado Benatti
	Fonte:	Tese/Dissertação
	Score:	0.223
	Palavras:	[ ]

Tabela 3.13: Exemplo 3 - BERT, Artificial Intelligence.

Capítulo 4

Conclusões

Esse trabalho explorou a aplicação de técnicas de Processamento de Linguagem Natural (NLP) na recomendação de texto, com foco nas abordagens baseadas nos modelos Word2Vec, GloVe e BERT. Ao longo do estudo, foi possível compreender como esses modelos podem ser utilizados para extrair informações semânticas e contextuais dos documentos de texto, a fim de gerar recomendações mais precisas e personalizadas.

Com o objetivo de criar um sistema de recomendação de artigos científicos, atuamos desde a extração dos dados via web scraping ao ajustes de modelos pré-treinados altamente reconhecidos na NLP. Entre os métodos apresentados, o grande destaque deste estudo foi o modelo BERT, que apresentou uma capacidade notável de compreender o contexto dos textos, e realizar recomendações concisas de artigos científicos.

Como futuros trabalhos, existem diversos aspectos importantes que podem ser explorados. Um deles é o aumento do dataset, seja coletando mais artigos publicados pelos docentes (incentivando o cadastro no google scholar), e/ou adicionando as monografias dos discentes da graduação. O BERT foi revolucionário quando lançado em 2018, e novos algoritmos continuaram sendo desenvolvidos e aprimorados. Algoritmos como BART (Bidirectional and AutoRegressive Transformers) e GPT (Generative Pre-trained Transformer) são exemplos de abordagens mais recentes que podem ser exploradas para melhorar a capacidade da recomendação de texto. Por fim, é interessante desenvolver métricas de avaliação da qualidade das recomendações, criando por exemplo um conjunto de dados de validação, que permitirá uma análise mais precisa do desempenho dos modelos. Essas possibilidades de trabalhos futuros permitirão o desenvolvimento de um sistema de recomendação mais eficiente e eficaz.

Referências Bibliográficas

[1] E. M. Dharma e et al. **The accuracy comparison among Word2Vec, GloVe, and FastText towards Convolution Neural Network (CNN) text classification.**. Journal of Theoretical and Applied Information Technology, [S. l.], v. 100, n. 2, p. -, 31 jan. 2022.

[2] I. Malkiel e et al. **RecoBERT: A Catalog Language Model for Text-Based Recommendations**.

[3] J. Pennington e et al. **GloVe: Global vectors for word representation**. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014. p. 1532-1543.

[4] J. Devlin e et al. **BERT: Pre-training of deep bidirectional transformers for language understanding.**. arXiv preprint arXiv:1810.04805 (2018).

[5] K. K. Subramanyam e S. Sangeetha. **A survey of embeddings in clinical natural language processing**. Journal of Biomedical Informatics, [S. l.], p. 2758-2765, 15 jan. 2020.

[6] M. Naili, A. H. Chaibi e et al. **Comparative study of word embedding methods in topic segmentation**. Procedia Computer Science, [S. l.], p. 340-349, 7 set. 2017.

[7] **RaRe-Technologies/gensim-data**. Disponível em: https://github.com/RaRe-Technologies/gensim-data. Acesso em: 15 jan. 2023.

[8] S. Rashka e V. Mirjalili. **Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2**. 3. ed. [S. l.]: Packt Publishing, 2019. 741 p. ISBN 978-1-78995-575-0.

[9] S. Kanwal, S. Nawaz, M. K. Malik e Z. Nawaz. **A Review of Text-Based Recommendation Systems**. IEEE Access, vol. 9, pp. 31638-31661, 2021, doi: 10.1109/ACCESS.2021.3059312.

[10] **sentence-transformers/all-mpnet-base-v2· Hugging Face**. Disponível em: https://huggingface.co/sentence-transformers/all-mpnet-base-v2. Acesso em: 15 jan. 2023.

[11] T. Mikolov, K. Chen, G. Corrado, e J. Dean. **Efficient estimation of word representations in vector space**. Proceedings of the International Conference on Learning Representations (ICLR 2013), 2013, pp. 1–12.

[12] T. Mikolov e et al. **Distributed representations of words and phrases and their compositionality**. Advances in neural information processing systems, 2013, 26.

[13] W. Zhang, T. Yoshida e X. Tang. **A comparative study of TFIDF, LSI and multi-words for text classification**. Elsevier, [S. l.], v. 38, p. 2758-2765, 27 jan. 2023.

[14] **Word2Vec**. [S. l.], 29 jul. 2013. Disponível em: https://code.google.com/archive/p/Word2Vec/. Acesso em: 15 jan. 2023.