CIÊNCIAS EMPRESARIAIS
ESCOLA SUPERIOR
POLITÉCNICO SETÚBAL

PEDRO HENRIQUE
MELLO PEREIRA

Utilização de técnicas de Processamento de Linguagem Natural para construção de um sistema de recomendação baseado na similaridade de vetores de representação textual

Relatório de projeto de investigação do Mestrado em Ciência de Dados Para Empresas

ORIENTADOR

Professor Doutor David Alexandre Mendes Silva Simões

Professor Doutor Vítor Manuel Meneses Barbosa

NOVEMBRO 2025

PEDRO HENRIQUE
MELLO PEREIRA

Utilização de técnicas de Processamento de Linguagem Natural para construção de um sistema de recomendação baseado na similaridade de vetores de representação textual

JÚRI

Presidente: Professora Doutora, Ana de Jesus Pereira Barreira Mendes, ESCE/IPS
Orientador: Professor Doutor, David Alexandre Mendes Silva Simões, ESCE/IPS
Vogal: Professora Ph.D., Paula Cristina Rodrigues Miranda, EST/IPS

NOVEMBRO 2025

Índice

Lista de Figuras

Lista de Tabelas

Lista de abreviaturas e siglas

- **API** - *Application Programming Interface*
- **BERT** - *Bidirectional Encoder Representations from Transformers*
- **CBOW** - *Continuous Bag-of-Words*
- **CGDADOS** - *Coordenação-Geral de Ciência de Dados*
- **EVG** - *Escola Virtual de Governo*
- **ENAP** - *Fundação Escola Nacional de Administração Pública*
- **IA** - *Inteligência Artificial*
- **ISBN** - *International Standard Book Number*
- **LDA** - *Latent Dirichlet Allocation*
- **LSTMs** - *Long Short-Term Memory networks*
- **MLM** - *Masked Language Modeling*
- **NSP** - *Next Sentence Prediction*
- **PDP** - *Plano de Desenvolvimento de Pessoas*
- **PLN** - *Processamento de Linguagem Natural*
- **PMV** - *Produto Mínimo Viável*
- **PNDP** - *Política Nacional de Desenvolvimento de Pessoas*
- **RNNs** - *Recurrent Neural Networks*
- **SBERT** - *Sentence-BERT*
- **SIPEC** - Sistema de Pessoal Civil da Administração Federal
- **SQL** - *Structured Query Language*
- **SR** - *Sistema de Recomendação*
- **TF-IDF** - *Term Frequency - Inverse Document Frequency*

Resumo

Num contexto global de produção massiva de informação, os dados textuais não estruturados — frequentemente desprezados ou subutilizados — apresentam-se como uma oportunidade para organizações que pretendem personalizar a experiência do utilizador sem, contudo, investir montantes elevados em infraestruturas de dados dispendiosas. Uma vez que os sistemas de recomendação tradicionais, especialmente aqueles baseados em filtragem colaborativa, exigem grandes volumes de dados comportamentais, tornando-se excessivamente onerosos para organizações de menor dimensão, este estudo propõe como alternativa acessível e igualmente eficaz o desenvolvimento de sistemas de recomendação inteiramente baseados em conteúdo textual. Para tal, aplicaram-se técnicas de processamento de linguagem natural (PLN) para analisar a similaridade semântica entre vetores de representação textual, com baixo recurso a capacidade computacional.

Foram desenvolvidos dois protótipos — um baseado em *Term Frequency - Inverse Document Frequency* (abordagem de ponderação lexical) e outro em *Sentence BERT* (arquitetura de *transformers* e *embeddings* densos) — avaliando a sua capacidade de gerar recomendações em dois cenários: o catálogo de oitocentos e sete cursos da Escola Virtual de Governo (EVG) e uma coleção de quase trinta mil livros da *Google Books API*. Os resultados evidenciaram que ambos os modelos produzem sugestões relevantes, sendo o baseado em *Sentence BERT* aquele que se destacou pela precisão semântica na captura de nuances contextuais (por exemplo, a distinção de polissemias e relações temáticas implícitas), enquanto o *Term Frequency - Inverse Document Frequency* mostrou-se preciso na identificação de correspondências lexicais exatas entre palavras-chave. A aplicação prática nestes domínios demonstrou a viabilidade de implementação, com recomendações adaptáveis a diferentes necessidades. Este trabalho demonstra que técnicas modernas de processamento de linguagem natural podem democratizar o acesso a sistemas de recomendação e oferecer soluções eficientes às mais diversas organizações, sem comprometer a qualidade das sugestões.

Palavras-chave: Processamento de Linguagem Natural, Sistemas de Recomendação, *Embeddings*, Similaridade Semântica.

Abstract

In a global context of massive information production, unstructured textual data — often overlooked or underutilized — represents an opportunity for organizations seeking to personalize user experience without investing heavily in costly data infrastructures. Since traditional recommendation systems, particularly those based on collaborative filtering, require large volumes of behavioral data and are therefore prohibitively expensive for smaller organizations, this study proposes an accessible and equally effective alternative: the development of recommendation systems entirely based on textual content. To achieve this, Natural Language Processing (NLP) techniques were applied to analyze semantic similarity between textual representation vectors, with minimal computational resource requirements.

Two prototypes were developed — one based on Term Frequency - Inverse Document Frequency (a lexical weighting approach), and another based on Sentence BERT (a transformers architecture using dense embeddings) — and their recommendation capabilities were evaluated in two scenarios: the catalog of eight hundred seven courses from the Escola Virtual de Governo (EVG), and a collection of almost thirty thousand books from the Google Books API. The results showed that both models generated relevant suggestions, with the Sentence BERT model standing out for its semantic precision in capturing contextual nuances (such as distinguishing polysemies and implicit thematic relationships), while the TF-IDF model proved accurate in identifying exact lexical matches between keywords. The practical application in these domains demonstrated the feasibility of implementation, with recommendations adaptable to different needs. This work demonstrates that modern NLP techniques can democratize access to recommendation systems, providing efficient solutions for a wide range of organizations without compromising suggestion quality.

Keywords: Natural Language Processing, Recommender Systems, Embeddings, Semantic Similarity.

Introdução

Esta secção tem por escopo apresentar de maneira estruturada o contexto e as motivações sob as quais repousa esta produção académica, de modo a expressar aos leitores e avaliadores os objetivos da investigação, sejam eles mediatos ou imediatos, a metodologia utilizada e a estrutura do relatório.

Contexto e motivação

É público e notório que estamos a viver a era do *big data*. Todos os dias uma quantidade massiva de dados é produzida e despejada na internet pelos mais diversos dispositivos eletrónicos existentes. Ressalta-se que a maior parte destes dados enquadram-se nos chamados "dados não estruturados", que são aqueles que não possuem forma ou esquema pré-definidos e não constituem necessariamente vetores de características numéricas ou categóricas, como por exemplo textos, imagens, áudios e vídeos.

Por conseguinte, uma das consequências mais evidentes deste novo tempo diz respeito à mudança de paradigma do comércio de mercadorias, bens e serviços. Se antes da adoção da internet pela maior parte da população mundial o canal de vendas principal da maioria das empresas era o presencial, com a massificação do acesso à internet e o crescimento exponencial do comércio eletrónico, o ambiente digital assumiu um papel central nas estratégias de vendas. Logo, haja vista que a presença *online* tem se mostrado cada vez mais determinante para o sucesso do negócio, as empresas passaram a investir significativamente em suas plataformas na internet, de modo a permitir aos consumidores a aquisição de mercadorias, bens e serviços a partir de qualquer lugar e a qualquer momento.

Ocorre que, além das inúmeras novas oportunidades criadas para as empresas, este novo paradigma trouxe também importantes desafios, sendo o aumento da concorrência um dos mais proeminentes. A partir deste momento a concorrência tornou-se global e trouxe consigo a necessidade de inovação constante e personalização de ofertas para atender a um público cada vez mais exigente e diverso.

Ademais, ao imaginarmos um comércio eletrónico com centenas ou até milhares de produtos, pode ser extremamente complexo para o cliente encontrar o item que está à procura, o que pode tornar a experiência de compra frustrante e desmotivadora e levá-lo à desistência.

Neste contexto é que surgem os sistemas de recomendação, os quais permitem a oferta personalizada de sugestões aos consumidores baseadas na similaridade entre os itens comprados ou pesquisados anteriormente, o que pode facilitar a experiência de compra *online* e impulsionar as vendas da empresa.

Há aqui entretanto uma questão: sistemas de recomendação tradicionais podem exigir investimentos volumosos para recolha, tratamento e armazenamento de grandes quantidades de dados quanto às preferências

dos clientes, o que pode inviabilizar a implementação de tais ferramentas por empresas de menor porte ou que estejam a iniciar suas atividades com poucos recursos.

Desta feita, este trabalho tem por motivação demonstrar que o desenvolvimento e aplicação de sistemas de recomendação (SR) baseados nos dados gerados diariamente no ambiente comercial pode apresentar-se como uma grande vantagem competitiva, sobretudo ao utilizar-se dos dados textuais (não estruturados), que são abundantes e ainda sub-utilizados no que diz respeito à geração de conhecimento e inteligência empresarial. Com isto, pretende-se demonstrar uma maneira de desenvolver SRs que possam ser implementados nos mais diversos contextos organizacionais, sobretudo em organizações que possuem poucos recursos para investir na aquisição de dados, diminuindo-se portanto a barreira de entrada para a adoção desta tecnologia e tornando-a mais acessível.

Objetivos

De acordo com o contexto retromencionado, o presente estudo tem por objetivos:

a) Investigar os avanços no campo do PLN, com ênfase na aplicação criativa destas técnicas para criação de sistemas de recomendação não supervisionados e totalmente baseados na similaridade textual entre os itens. O cerne neste ponto é analisar o quão eficaz pode ser essa abordagem, sem a necessidade à partida de recolha de dados comportamentais pelos utilizadores, de modo a comprovar a sua relevância para empresas que possuem orçamentos modestos ou limitados para investimentos em infraestrutura de recolha, armazenamento e processamento de dados;
b) Desenvolver dois protótipos de sistemas de recomendação baseados nas características textuais que descrevem os itens, utilizando-se para tanto técnicas de processamento de linguagem natural (PLN) e similaridade do cosseno entre os vetores de representação textual gerados por cada um dos sistemas. Esses protótipos devem ser projetados para utilizar uma quantidade mínima de dados, recorrendo apenas às descrições e demais dados textuais relevantes sobre os itens, de modo que sejam especialmente adequados para empresas em estágio inicial ou com recursos limitados, uma vez que, diferentemente dos sistemas de recomendação tradicionais, que frequentemente dependem de investimentos significativos para recolha e armazenamento de dados comportamentais dos utilizadores, essa abordagem servirá para reduzir a barreira de entrada, oferecendo uma alternativa eficiente e acessível para recomendar itens de forma eficaz. Os protótipos desenvolvidos devem seguir as seguintes diretrizes:
    - Um sistema de recomendação baseado na técnica TF-IDF para extração de vetores de representação textual esparsos, onde seja imputado como dado de entrada um determinado item e sejam emitidas ao menos duas recomendações de itens similares, tanto para a base de dados de

cursos à distância quanto para a base de livros;

- Um sistema de recomendação baseado em arquitetura de Transformers para extração de vetores de *embeddings* densos, onde seja imputado como dado de entrada um determinado item e sejam emitidas ao menos duas recomendações de itens similares, com uso das mesmas bases de dados anteriormente mencionadas;

c) Comparar o desempenho de ambos os sistemas no que diz respeito a:

- Qualidade das recomendações emitidas;
- Capacidade de perceção de contexto e nuances semânticas. O que interessa ao projeto neste ponto é perceber a diferença entre vetores de representação textual contextuais (SBERT) e não contextuais (TF-IDF) e como isso afeta o resultado final no que tange à qualidade das recomendações geradas;
- Capacidade de generalização em diferentes domínios;
- Capacidade de recomendação de sequências naturais de itens (Por exemplo, se lhe é imputado como dado de entrada o item "Harry Potter e a Câmara Secreta" o sistema deve ser capaz de recomendar naturalmente o próximo item da sequência, como "Harry Potter e o Prisioneiro de Azkaban", além de recomendar itens semelhantes, como outros livros de fantasia similares à série do exemplo);

d) Finalizar o projeto tendo avançado significativamente no entendimento do funcionamento e aplicação das principais técnicas de PLN.

Metodologia

A primeira etapa da elaboração do projeto diz respeito à revisão da literatura para expansão do conhecimento sobre as particularidades dos sistemas de recomendação e as técnicas de PLN a serem empregadas. De seguida, decorreu a fase de levantamento dos requisitos e necessidades para preparar o ambiente de desenvolvimento da maneira mais adequada à prototipagem dos sistemas de recomendação. Ainda nesta fase também foi realizada a extração e tratamento dos dados utilizados no estudo, os quais foram obtidos de fontes públicas consideradas como seguras e fiáveis. Por fim, procedeu-se à elaboração dos protótipos dos sistemas de recomendação, os quais diferem por meio das técnicas utilizadas para extração dos vetores de representação textual, bem como à avaliação e comparação dos resultados para elaboração das conclusões e entendimento quanto aos próximos passos.

Estrutura do relatório

O primeiro capítulo deste relatório de projeto traz em seu cerne um estudo a respeito da literatura especializada sobre o tema, com vistas a clarificar conceitos basilares, tais como os tipos de sistema de recomendação, as etapas do processo de mineração textual com ênfase na evolução dos vetores de representação textual e como a similaridade do cosseno aplica-se ao caso em tela, e estabelecer uma linha de raciocínio clara e concisa de modo a guiar o leitor ao entendimento das etapas a seguir. Neste ponto apresentou-se em apertada síntese a evolução das técnicas de processamento de linguagem natural durante as últimas décadas e o estágio em que se encontra atualmente, sendo este um dos campos da Inteligência Artificial que apresentou os maiores avanços na história recente.

No capítulo a seguir justificou-se a metodologia utilizada na prototipagem dos sistemas de recomendação e qual foi o método de avaliação de resultados utilizado.

O capítulo 3 diz respeito à apresentação dos resultados obtidos por meio da aplicação dos algoritmos desenvolvidos tanto à base de dados de oitocentos e sete cursos ofertados na plataforma da Escola Virtual de Governo (EVG), a qual possui notoriedade por ser a maior plataforma pública de cursos voltados à capacitação dos servidores públicos no Brasil, quanto à base dos mais de vinte e nove mil livros oriunda da *Google Books API*, de modo a avaliar o quão ajustadas e assertivas foram estas recomendações no que diz respeito à similaridade com os *inputs* apresentados a cada um dos sistemas.

Por último, apresentam-se no capítulo 4 as conclusões obtidas no estudo, bem como os desafios encontrados e os limites dos sistemas. Aqui encontram-se também as propostas de investigação futura.

1 Revisão de literatura

Após a introdução do projeto e exposição do contexto e motivação pelas quais fora produzido, o presente capítulo objetiva apresentar com maior profundidade as definições existentes na literatura especializada, tanto sobre os sistemas de recomendação, quanto sobre o processamento de linguagem natural enquanto sub-campo da inteligência artificial. Aqui serão apresentados os conceitos essenciais a respeito dos tipos de SRs existentes, as técnicas de vetorização e representação textual sob o prisma do PLN, a evolução do campo do PLN ao longo das últimas décadas, a intersecção entre as áreas do estudo e, finalmente, a fundamentação da métrica de similaridade escolhida para ser aplicada aos vetores de representação textual de modo a obter os resultados esperados.

1.1 Sistemas de recomendação

Segundo Ricci et al. (2011), os Sistemas de Recomendação (SR) são ferramentas e técnicas de software que fornecem sugestões de itens que podem ser úteis para um utilizador. Tais sugestões tem como objetivo principal apoiar os utilizadores no processo de tomada de decisão, como, por exemplo auxiliando-os sobre qual livro comprar numa biblioteca virtual, qual curso obter para expandir o conhecimento em determinada área ou quais notícias mais o apetecem em um sítio de notícias *online*.

Este tipo de sistema baseia-se principalmente nas interações entre:

- **Utilizador:** trata-se do sujeito que recebe as recomendações de itens por parte do fornecedor;
- **Fornecedor:** aquele que recomenda os itens ao utilizador em sua plataforma;
- **Item:** aquilo que está a ser recomendado ao utilizador;

Os SR podem desempenhar diferentes funções. Sob a ótica do fornecedor, um SR pode ser usado por exemplo para ampliar as vendas ou fidelizar os utilizadores da sua plataforma. Do ponto de vista do utilizador, o objetivo principal é encontrar itens relevantes, úteis e adequados às suas necessidades, sem precisar despender muito tempo nesta tarefa.

Por conseguinte, Herlocker et al. (2004) identificam tarefas principais em sistemas de recomendação (SR), das quais as três mais relevantes para este estudo são:

- **Recomendação de alguns itens bons:** o SR recomendará uma lista classificada de itens que possuem maior chance de satisfazer as necessidades do utilizador, com ou sem previsão explícita de avaliação.
- **Anotação em contexto:** o sistema destaca alguns itens de uma lista com base nas preferências de longo prazo do utilizador.

- **Recomendar uma sequência:** o sistema recomenda uma sequência de itens que seja agradável como um todo.

Ressalta-se que, apesar de as duas primeiras hipóteses elencadas anteriormente serem as abordagens mais populares em termos de aplicação de um SR, há de se ter em consideração que estas necessitam de um elevado esforço e investimento na recolha, armazenamento e tratamento de dados de *reviews* e *feedbacks* dos utilizadores, o que pode traduzir-se em custos significativos para a organização que está a implementar este recurso em sua plataforma comercial. Tal abordagem é chamada de Filtragem Colaborativa (Schafer et al., 2001).

Desta feita, **este estudo concentra-se na hipótese de recomendação de sequência**, em que recomenda-se ao utilizador os itens mais similares aos últimos itens consumidos ou pesquisados por ele. Desta maneira, a um utilizador que consumiu um livro de estatística para *machine learning* deve ser recomendado, por exemplo, um livro da linguagem R para Data Science. A esta abordagem denomina-se "Filtragem baseada no conteúdo" (Son & Kim, 2017), a qual será abordada no tópico a seguir.

1.1.1 Tipos de sistemas de recomendação

Quando estamos a falar em sistemas de recomendação, existem três tipos de técnicas que podem ser utilizadas em sua conceção (Burke, 2002; Ricci et al., 2011), as quais destacam-se a seguir:

a) ***Content-based*:**

Baseia-se na recomendação dos itens mais similares àqueles adquiridos ou pesquisados pelo utilizador anteriormente. Neste tipo de SR o que importa é a similaridade entre as características dos itens, de modo a utilizar unicamente o último ou o histórico dos últimos itens consumidos pelo utilizador para recomendar itens semelhantes, sem considerar contudo as variáveis relativas ao próprio utilizador.

Conforme demonstra a Figura 1, imagine-se que o utilizador João leu o livro "Os Segredos da Mente Milionária". Com base nessa interação, o sistema analisaria as características do livro, como género (desenvolvimento pessoal), temas abordados (educação financeira e mentalidade de riqueza) e descrição textual da obra e recomendaria livros como "Pai Rico, Pai Pobre" de Robert Kiyosaki ou "A Mente Acima do Dinheiro" de Ted Klontz, que partilham características similares e tendem a atrair o mesmo público-alvo.

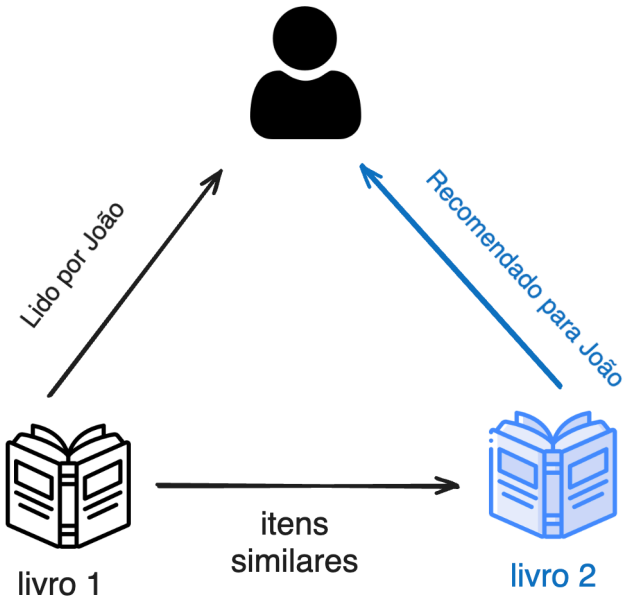

Figura 1: Exemplo de Sistema de Recomendação Baseado em Conteúdo.

Como principais vantagens deste tipo de sistema Aggarwal (2016) ressalta a sua simplicidade e menor necessidade de dados se comparado a outros sistemas de recomendação, haja vista não ser necessário a recolha de dados acerca do comportamento dos utilizadores para gerar recomendações, bastando para tanto o conhecimento a respeito de atributos descritivos dos itens consumidos pelo utilizador.

Em contrapartida, Ricci et al. (2011) descreve como desvantagem deste tipo de SR a limitação da descoberta de novas preferências pelos utilizadores, uma vez que ao receber apenas recomendações de itens similares dificilmente ocorreria uma exposição a itens diversos que poderiam expandir seus interesses, limitando portanto a serendipidade do sistema segundo o autor. Além disso, Azeroual & Koltay (2022) cita como um importante desafio enfrentado pelos sistemas baseados em conteúdo a elevada dependência da qualidade e granularidade das características que descrevem os itens para que se tenham boas recomendações, vez que apenas o conteúdo descritivo do item é levado em consideração para a recomendação.

b) ***Collaborative-filtering*:**

Se na técnica baseada em conteúdo os dados relativos ao utilizador não eram considerados, há aqui uma lógica de funcionamento diferente: A filtragem colaborativa utiliza os dados recolhidos de todos os utilizadores para recomendar a um determinado utilizador conteúdos que pessoas com o perfil semelhante ao dele consumiram ou avaliaram de maneira positiva anteriormente. Schafer et al. (2001) refere-se à filtragem

colaborativa como *"people-to-people correlation"*, uma vez que a similaridade entre os gostos de dois utilizadores é calculada com base no quão similares são as avaliações atribuídas a itens partilhados.

A Figura 2 traz um exemplo do funcionamento desta abordagem. Imagine-se que o utilizador Pedro assistiu e avaliou positivamente o filme "A Origem". O sistema deteta que outros utilizadores que também gostaram desse filme, como Maria, assistiram a "Blade Runner 2049" e o avaliaram positivamente. Com base nessa correlação de gostos, o sistema recomenda "Blade Runner 2049" para Maria, considerando que seus interesses são semelhantes aos de Pedro.

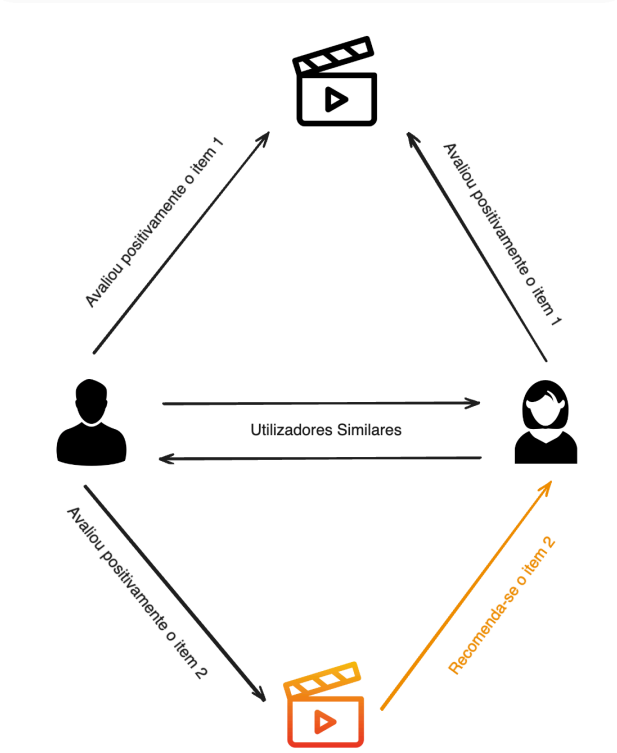

Figura 2: Exemplo de Sistema de Recomendação Baseado em Filtragem Colaborativa.

Como vantagens da filtragem colaborativa, Azeroual & Koltay (2022) cita a capacidade deste tipo de SR no reconhecimento de padrões complexos nas preferências dos utilizadores sem necessitar de informações detalhadas a respeito dos itens. Sua versatilidade permite aplicação em vários domínios, o que facilita a sugestão de novos itens que não partilham características explícitas, de modo a ampliar o escopo das recomendações. Além disso, para Aggarwal (2016) este tipo de SR promove descobertas inesperadas ao basear-se nas interações entre utilizadores, pois os expõe a opções que podem não estar diretamente ligadas às suas escolhas anteriores, mas que se alinham com as preferências de utilizadores semelhantes.

Contudo, a filtragem colaborativa também enfrenta desafios importantes. Um deles é o problema do *"cold start"*, onde a falta de dados históricos dificulta as recomendações iniciais para novos utilizadores

(Melville & Sindhwani, 2017; Schein et al., 2002). A eficácia do sistema também depende sobremaneira da quantidade e qualidade dos dados existentes, especialmente em cenários com poucas interações registadas (Azeroual & Koltay, 2022). Ademais, o processamento de grandes volumes de dados para calcular similaridades pode ser computacionalmente custoso.

c) **Sistema Híbrido:**

Ricci et al. (2011) definem sistemas híbridos como aqueles que combinam as técnicas mencionadas anteriormente. Segundo os autores, esses sistemas buscam aproveitar os pontos fortes de uma abordagem (A) para mitigar as desvantagens da outra (B).

Em complemento, Aggarwal (2016) pontua que os sistemas de recomendação híbridos foram desenvolvidos para explorar as possibilidades existentes no poder algorítmico de tipos diferentes de sistemas de recomendação para a produção de inferências mais robustas. Além disso, o autor sugere a utilização desse tipo de sistema como uma opção para lidar com o problema do *cold start* mencionado anteriormente.

Ocorre que, apesar de suas capacidades avançadas, este sistema traz consigo um importante desafio, sobretudo para organizações com recursos limitados, que diz respeito aos elevados custos computacionais. Kılıç (2023) aponta que a eficiência computacional é uma métrica crítica neste tipo de sistema de recomendação. Para o autor, as complexas interações entre múltiplos algoritmos nos SRs híbridos podem aumentar significativamente a carga computacional necessária para processá-las.

1.2 Processamento de linguagem natural

Kamath & Kanakaraj (2015) propuseram um sistema de recomendação de notícias eletrónicas (*e-news*) que utiliza técnicas de processamento de linguagem natural (PLN) para melhorar a experiência do utilizador em mecanismos de pesquisa. O sistema aborda a limitação das técnicas tradicionais baseadas em *bag-of-words*, que se baseiam na correspondência exata de palavras-chave, o que pode resultar na omissão de documentos potencialmente relevantes que não contêm as palavras-chave exatas da consulta do utilizador.

O algoritmo proposto pelos autores utiliza uma abordagem semântica *bag-of-words* aprimorada, incorporando sinónimos de *WordNet* (Miller et al., 1990) para aumentar a cobertura da pesquisa e capturar artigos de notícias potencialmente relevantes. Ao considerar sinónimos, o sistema expande o alcance da correspondência de palavras-chave, aumentando a probabilidade de recuperar documentos relevantes que podem usar termos diferentes para descrever o mesmo conceito, o que confere maior flexibilidade e alcance ao algoritmo. Por conseguinte, o sistema realiza agrupamento de documentos com base em vetores de representação textual ponderados de maneira proporcional à sua frequência no documento e inversamente proporcional à sua frequência na coleção de documentos.

O estudo destaca em seu cerne o potencial das técnicas de PLN para melhorar os sistemas de recomendação de notícias eletrónicas, fornecendo aos utilizadores resultados de pesquisa mais completos e precisos.

Apesar de ainda serem importantes e relevantes até os dias de hoje, técnicas como as mencionadas anteriormente baseadas em *bag-of-words* falham por não captarem a semântica — a qual pode ser percebida neste espectro como "a determinação do que uma sentença significa e as condições sob as quais ela é verdadeira" (Chowdhary, 2020) —, o que levou à ascenção de técnicas mais modernas, baseadas sobretudo em redes neurais e arquitetura de *transformers* para captação avançada de contexto por meio dos mecanismos de atenção (Vaswani, 2017).

As secções seguintes deste capítulo abordarão as nuances do processamento de linguagem natural de modo a demonstrar a relevância deste ramo da Inteligência Artificial no desenvolvimento dos sistemas de recomendação baseados em conteúdo ora propostos, preparando o leitor para compreender a metodologia utilizada, a qual será apresentada no capítulo 2.

1.2.1 **Problemas passíveis de serem solucionados com uso do PLN**

A primeira etapa de qualquer projeto que utilize PLN diz respeito à identificação do problema, ou seja, definir o tipo de tarefa a ser desenvolvida. Ressalta-se que cada uma dessas tarefas exige abordagens específicas de pré-processamento, com vistas a garantir que os modelos consigam extrair significado dos textos de forma eficiente para a consecução do objetivo.

No caso em tela, o foco está na similaridade semântica entre frases, essencial para a construção de sistemas de recomendação baseados em conteúdo, a qual pode ser mensurada por meio de métricas tradicionais de similaridade/distância em espaço vetorial, como a similaridade do cosseno, *Jaccard* dentre outras. Esse tipo de sistema compara textos e sugere itens com base na semelhança semântica entre seus textos de apresentação.

Alguns exemplos de tarefas em PLN podem ser encontradas em Sun et al. (2022), a saber:

- **Classificação Textual:** categorização automatizada de textos de acordo com as suas áreas temáticas;
- **Reconhecimento de Entidades Nomeadas (NER):** identificação de nomes de pessoas, locais, empresas dentre outros em um texto, útil sobretudo para extração automatizada de informações em notícias e documentos.
- **Sumarização textual:** geração de resumos concisos e informativos a partir de textos mais extensos;

1.2.2 **Pré-processamento de dados textuais**

Após a identificação do problema que se deseja resolver com o uso do PLN, a próxima etapa que deve ser considerada diz respeito ao pré-processamento, que também pode ser percebido como o estágio em que se realiza a limpeza e transformação dos dados textuais. Esta etapa é essencial para transformar o texto bruto em um formato adequado para análise e modelagem, visando garantir que ruídos e inconsistências não comprometam o desempenho do sistema.

A Figura 3 demonstra um fluxo com algumas das técnicas de pré-processamento mais proeminentes no âmbito da mineração textual, técnicas estas que serão expostas a seguir, com o objetivo de demonstrar os passos que devem ser seguidos no desenvolvimento de soluções eficazes no âmbito do PLN.

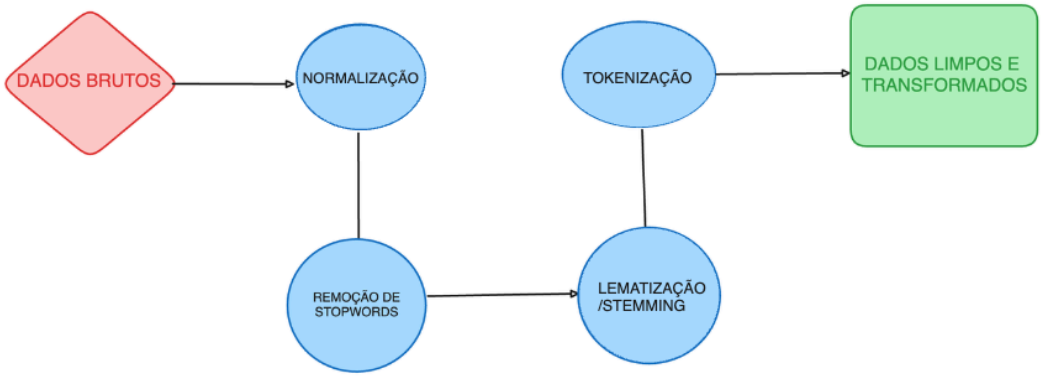

Figura 3: Fluxo de Pré-Processamento de Dados Textuais.

1.2.2.1 Normalização

De acordo com Jurafsky & Martin (2025), a normalização consiste em transformar as palavras ou *tokens* para um formato padrão. Neste ponto todo o texto pode ser convertido para letras minúsculas, naquilo que os autores retromencionados chamam de *case folding* ou maiúsculas (a depender do caso), podendo incluir-se ainda a remoção de caracteres especiais, sinais de pontuação e acentuação, para que variações desnecessárias sejam eliminadas.

O objetivo principal desta etapa é evitar que palavras com pequenas variações sejam tratadas como distintas, de modo a reduzir a dimensionalidade do vocabulário e tornar a representação dos textos mais eficiente, sendo muito útil para o desenvolvimento de tarefas de PLN como recuperação de informação, reconhecimento de fala, tradução automática, classificação textual dentre outras (Jurafsky & Martin, 2025).

1.2.2.2 Remoção de stopwords

Stopwords são palavras comuns e pouco informativas que aparecem diversas vezes ao longo de um texto, como os pronomes, artigos e preposições. A depender da tarefa pretendida, estas palavras devem ser removidas pois geralmente possuem pouco conteúdo lexical e podem gerar ruídos desnecessários no processamento (Bird et al., 2009).

1.2.2.3 Lematização e stemming

> *Lemmatization is the task of determining that two words have the same root, despite their surface differences. The words am, are, and is have the shared lemma be; the words dinner and dinners both have the lemma dinner. Lemmatizing each of these forms to the same lemma will let us find all mentions of words in Polish like Warsaw. The lemmatized form of a sentence like He is reading detective stories would thus be He be read detective story*. (Jurafsky & Martin, 2025, p. 23)

A lematização consiste em reduzir as palavras à sua forma base (lema), levando em conta o contexto e a gramática. Esta técnica de pré-processamento usa dicionários linguísticos para garantir que a palavra resultante seja válida. Por exemplo, as palavras "estudando", "estudou" e "estudam" são todas reduzidas a "estudar", pois essa é a **forma canónica do verbo.**

Ocorre que algoritmos de lematização podem ser muito complexos, devendo recorrer-se em alguns casos à aplicação de um método mais simples e rudimentar chamado *stemming* (Jurafsky & Martin, 2025). Assim sendo, *stemming* é uma técnica mais simples, que elimina sufixos para reduzir palavras à sua forma raiz, sem contudo considerar a gramática. Ele usa regras fixas para truncar as palavras, o que pode gerar formas sem sentido. No exemplo anterior, "estudando", "estudou" e "estudam" seriam reduzidos a "estud".

Embora o stemming reduza o vocabulário de forma eficiente, sua falta de precisão pode ser um problema em algumas aplicações. Conforme Krovetz (1993), *stemmers* simples podem cometer tanto erros de supergeneralização (transformar *policy* em *police*) como erros de subgeneralização (deixar de transformar *European* para *Europe*).

1.2.2.4 Tokenização

Trata-se do processo de divisão do texto em unidades menores, conhecidas como ***Tokens***, as quais podem ser frases, palavras, símbolos, números ou outros tipos de caracteres. Conforme Jurafsky & Martin (2025), há duas classes principais de algoritmos para essa tarefa: *top-down* e *bottom-up*.

Na estratégia *top-down*, define-se um conjunto de regras linguísticas e padrões bem definidos que

orientam a segmentação do texto. Já na tokenização *bottom-up*, o texto é dividido com base estatísticas simples sobre a frequência e coocorrência de sequências de caracteres, originando um vocabulário de sub-palavras, fragmentos que podem ser palavras inteiras, partes delas ou caracteres. Para os autores, esta última abordagem é comum em modelos modernos de PLN, por ser mais flexível e adaptável a diferentes domínios.

1.2.3 **Representação textual em formato vetorial: do *bag-of-words* à semântica contextual.**

Desde o advento dos mecanismos de atenção (Vaswani, 2017), tornou-se evidente o potencial da inteligência artificial generativa. No entanto, conforme demonstrado na Figura 4, é importante destacar que a Inteligência Artificial (IA) é um grande domínio do qual fazem parte a aprendizagem de máquina, a aprendizagem profunda e o processamento de linguagem natural, subdivisão de extrema importância na conceção das IAs generativas.

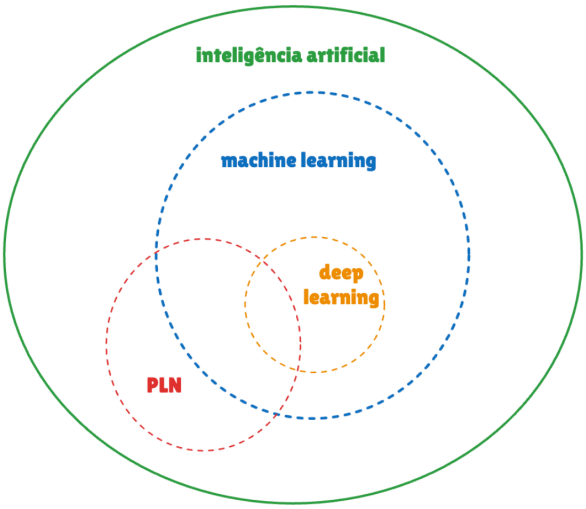

Figura 4: Classificação do Processamento de Linguagem Natural no campo da Inteligência Artificial.

O PLN engloba um conjunto de métodos computacionais que permitem que as máquinas compreendam, interpretem e manipulem a linguagem humana (Zohuri & McDaniel, 2022). Entretanto, a compreensão textual pelas máquinas não ocorre da mesma maneira que ocorre nos humanos, mas através de métodos computacionais que transformam as palavras ou frases em números, aos quais denomina-se vetores de representação textual, o que possibilita que os algoritmos interpretem e extraiam valor dos textos escritos em linguagem natural.

Desta forma, a vetorização textual desempenha um papel central no processamento de linguagem natural,

sendo responsável por transformar palavras ou frases em representações numéricas. Conforme demonstrado anteriormente, o processo inicia-se na definição do problema, passando então ao pré-processamento, no qual o texto é dividido em unidades menores chamadas *tokens*, que passam por limpeza, padronização, e pela remoção de palavras com baixo valor semântico (*stopwords*). Após essa preparação, são utilizados métodos que variam em complexidade e eficácia para geração dos vetores. Tais métodos evoluíram significativamente ao longo do tempo, a começar por abordagens simples, baseadas na contagem de palavras e vocabulário (Salton et al., 1975), passando por representações distribuídas (Mikolov et al., 2013) e culminando em modelos sofisticados baseados em mecanismos de atenção, que capturam nuances semânticas profundas e contextuais (Vaswani, 2017). A evolução destes métodos será detalhada a seguir.

1.2.3.1 CountVectorizer

Os métodos de vetorização de texto têm a sua génese no campo da Recuperação de Informação, a começar pela noção de "estrutura distribucional" proposta por Harris (1954), que já tratava o texto como um conjunto de palavras sem ordem fixa (*"bag of words"*) para análise estatística. Posteriormente, Luhn (1957) formalizou a codificação estatística de documentos ao definir um vocabulário e contar a frequência de cada termo para fins de indexação automática. Por conseguinte, Salton et al. (1975) introduziram o *Vector Space Model*, representando documentos como vetores em que cada dimensão correspondia a um termo distinto, consolidando portanto o paradigma *bag-of-words*, do qual derivaram algumas implementações modernas como o *CountVectorizer* do *scikit-learn*.

Ante o exposto, o *CountVectorizer* consiste na criação de um vocabulário com todas as palavras únicas existentes em um texto e, após isso, a contagem da frequência de cada uma daquelas palavras ao longo do texto formando-se uma matriz onde cada linha representa um documento e cada coluna representa uma palavra do vocabulário formado.

Apresenta-se, na Tabela 1 a seguir, um exemplo deste tipo de representação:

a) **"Eu Gosto de PLN"**

b) **"PLN é incrível"**

	Eu	gosto	de	PLN	é	incrível
Documento a	1	1	1	1	0	0
Documento b	0	0	0	1	1	1

Tabela 1: Resultado da vetorização com CountVectorizer

Apesar de ser simples e de fácil implementação, esta abordagem apresenta desvantagens nítidas: elevada

sensibilidade a ruídos e ortografia, não captação de contexto, ordem das frases e relação entre palavras, alta dimensionalidade dos dados a depender do tamanho do vocabulário, geração de matrizes esparsas e a importância exacerbada dada às palavras mais frequentes em detrimento das menos frequentes.

1.2.3.2 TF-IDF (Term Frequency - Inverse Document Frequency)

Assim como o *CountVectorizer*, o TF-IDF também é um método de transformação de textos em valores numéricos com o intuito de representar a linguagem natural junto às máquinas. Ocorre que, diferentemente da abordagem anteriormente mencionada segundo a qual apenas a frequência absoluta das palavras importava para a formação dos vetores, o TF-IDF tem por objetivo reduzir a importância de termos muito comuns e destacar aqueles que podem ser mais relevantes (Sparck Jones, 1972).

Aqui, as palavras com maior número de ocorrências ao longo do documento tendem a ser tratadas como menos relevantes e têm sua importância diluída, de modo a conferir maior destaque àquelas palavras com menor ocorrência ao longo do texto que está a ser vetorizado (Salton & Buckley, 1988).

Term frequency (TF)

A frequência de termo (TF) representa a frequência relativa do termo (t) dentro do documento (d):

$$\mathrm{tf}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

onde,

- $(f_{t,d})$ é o número de vezes que o termo (t) aparece no documento (d),
- $(\sum_{t' \in d} f_{t',d})$ é o número total de termos no documento (d).

Inverse document frequency (IDF)

A frequência inversa de documento (IDF) mede a importância de um termo em uma coleção de documentos, e é calculada no âmbito deste trabalho de acordo com a implementação oficial da biblioteca *Scikit-Learn*, a saber:

$$\mathrm{idf}(t) \;=\; \log\!\left(\frac{1+N}{1+|\{\, d \in D : t \in d\}|}\right) \;+\; 1$$

onde,

- N é o número total de documentos na coleção D;
- $|\{\, d \in D : t \in d\}|$ é o número de documentos que contêm o termo t;
- O "+1" no numerador e denominador vêm do parâmetro smooth_idf=True, e o "+1" fora do log garante que $\mathrm{idf}(t) > 0$ mesmo para termos presentes em todos os documentos.

A métrica TF-IDF é então o produto da frequência do termo pela frequência inversa de documento:

$$\text{tfidf}(t, d) = \text{tf}(t, d) \times \text{idf}(t)$$

Por fim, todo o vetor TF–IDF de cada documento é normalizado pela norma Euclidiana (L_2), de modo que:

$$v_{\text{norm}} = \frac{v}{\|v\|_2} = \frac{v}{\sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}}$$

A Tabela 2 a seguir apresenta um exemplo do resultado da aplicação desta técnica em três documentos[1] diferentes, quais sejam:

- **"1. Este jogador é muito rápido e habilidoso."**
- **"2. Este jogador não é rápido mas é habilidoso."**
- **"3. Este jogador é talentoso e não é rápido."**

	este	habilidoso	jogador	mas	muito	não	rápido	talentoso
doc1	0.364544	0.469417	0.364544	0.000000	0.617227	0.000000	0.364544	0.000000
doc2	0.329995	0.424929	0.329995	0.558731	0.000000	0.424929	0.329995	0.000000
doc3	0.364544	0.000000	0.364544	0.000000	0.000000	0.469417	0.364544	0.617227

Tabela 2: Resultado da vetorização com TF-IDF

Por ser uma técnica que apresentou importantes avanços se comparada ao *CountVectorizer*, decidiu-se aplicar esta abordagem no desenvolvimento de um dos sistemas de recomendação deste trabalho, o qual será apresentado no capítulo da Metodologia.

Entretanto, apesar de ser uma das técnicas de vetorização textual mais importantes para o PLN ao longo do tempo e de apresentar uma notável evolução no que diz respeito à captação da importância dos termos para o sentido do documento, vez que reduz sensivelmente a influência de termos corriqueiros como os artigos e preposições, o TF-IDF ainda possui desvantagens, dentre as quais destacam-se a **não captação de contexto**, uma vez que o modelo funciona por meio da correspondência exata entre termos; Ademais, possui **elevada sensibilidade a variações ortográficas**, pois assim como o *CountVectorizer* também depende da formação de um vocabulário de termos únicos o que leva a mais uma questão na aplicação desta técnica que diz respeito à geração de matrizes esparsas, conforme exemplo da Tabela 3, as quais podem exigir elevado poder computacional para processamento, a depender do tamanho do conjunto de textos (Zhang et al., 2011); Por fim, não foram encontradas ao longo das pesquisas que embasaram este trabalho evidências suficientes

[1] Documento neste contexto diz respeito à porção total de texto presente em determinada linha do *dataframe*

que sustentem a assunção de que os termos mais frequentes ao longo dos documentos serão sempre menos importantes que os demais.

	anexo	animados	animais	ano	anomalias	anormalidade
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
3	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
4	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
...	...	...	...	...	...	...
802	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
803	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
804	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
805	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
806	0.000000	0.000000	0.000000	0.000000	0.173172	0.000000

Tabela 3: Exemplo de matriz esparsa TF-IDF. Cada linha representa um documento do conjunto de dados, e cada coluna diz respeito a uma palavra do vocabulário.

1.2.4 ***Word2Vec***

Nas palavras de Mikolov et al. (2013) *"Many current NLP systems and techniques treat words as atomic units – there is no notion of similarity between words, as these are represented as indices in a vocabulary"*.

Conforme dito pelos criadores do *Word2Vec*, as abordagens de vetorização textual apresentadas até então possuíam a desvantagem inerente de criação de vetores esparsos e demasiado grandes, além de não capturar nenhum tipo de relação de proximidade de significado entre as palavras do vocabulário. Desta forma, o *Word2Vec* foi proposto como uma alternativa por Mikolov et al. no ano de 2013. Este método introduziu a ideia de representações distribuídas de palavras, onde cada termo é mapeado em um espaço vetorial contínuo, permitindo a captação de nuances semânticas e sintáticas entre termos relacionados.

Essa abordagem materializou-se por meio da utilização de um modelo de linguagem baseado em uma rede neural probabilística do tipo *feed forward* para treinamento dos *word embeddings* do *word2vec*, com vistas a aprender representações densas de palavras diretamente a partir de grandes volumes de texto de maneira eficiente.

Esta técnica baseia-se em duas arquiteturas principais, segundo os autores, conhecidas como *continuous bag-of-words (CBOW)* e *skip-gram*. Enquanto o *CBOW* prevê a palavra atual com base no contexto circundante, o *skip-gram* tenta prever as palavras ao redor com base em uma única palavra central. Ambos os modelos utilizam camadas de projeção compartilhadas para mapear palavras em vetores de dimensão fixa, o que resulta em representações compactas e informativas.

Como vantagens desse modelo de geração de *embeddings* é possível citar a sua elevada eficiência do ponto de vista computacional, sobretudo se comparado às abordagens anteriores, e a sua capacidade de captar relações semânticas entre as palavras do *corpus*.

Apesar de ser considerado como um marco importante no campo da vetorização textual, o *word2vec* também enfrenta limitações, dentre as quais destacam-se a geração de vetores estáticos (cada palavra possui um vetor fixo), necessidade de dados em demasia para treinamento, sendo este um fator determinante para a qualidade das saídas (Mikolov et al., 2013) e o desafio de lidar com a polissemia (Neelakantan et al., 2015) e termos raros ou novos que não foram inseridos no modelo durante a etapa de treinamento.

1.2.5 ***Transformers e self-attention***

Introduzida por Vaswani et al. (2017), a arquitetura de transformers é considerada como um dos grandes marcos da história do processamento de linguagem natural. Esta arquitetura superou as limitações de modelos anteriores, como *Recurrent Neural Networks (RNNs)* e *Long Short-Term Memory networks (LSTMs)*, ao utilizar mecanismos de auto-atenção para capturar relações e dependências globais entre os termos em uma frase. A partir desse avanço, os *transformers* passaram a permitir a formação de vetores de representação textual dinâmicos e contextualizados, resolvendo um dos maiores desafios do PLN até então: a captação eficiente de contexto.

Esta técnica de vetorização textual possibilita o processamento paralelo completo de frases, de modo a captar o contexto bidirecional. Ao contrário de modelos como *Word2Vec*, que geram representações estáticas e unidirecionais (da esquerda para a direita), a arquitetura de *transformers* considera tanto o contexto à esquerda quanto à direita simultaneamente, gerando representações dinâmicas e contextualizadas.

Os mecanismos de autoatenção em modelos com esta arquitetura permitem que cada palavra em uma frase "preste atenção" em todas as outras palavras, incluindo ela mesma, para capturar dependências e relações contextuais. Tal técnica baseia-se nos conceitos de *query*, *key* e *value*. Em apertada síntese, cada palavra gera uma *query*, que é comparada com as *keys* de todas as outras palavras para calcular pesos de relevância. Esses pesos são então aplicados às *values* correspondentes, resultando em uma representação contextualizada da palavra.

Além das vantagens citadas anteriormente, este novo paradigma também mostra-se muito relevante pela sua flexibilidade, vez que além da aplicação no desenvolvimento de grandes modelos de linguagem como

GPT (Radford et al., 2018) e BERT (Devlin et al., 2019), esta arquitetura pode ser adaptada a diversas tarefas no campo do PLN como tradução, busca semântica, classificação textual e, sobretudo, à tarefa de similaridade de frases a qual constitui o núcleo dos sistemas de recomendação desenvolvidos neste trabalho.

1.2.5.1 BERT e SBERT: a espinha dorsal do sistema de recomendação baseado em Transformers

Introduzido por Devlin et al. (2019), o BERT (*Bidirectional Encoder Representations from Transformers*) é um modelo de linguagem baseado na arquitetura de transformers. Este modelo foi pré-treinado em grandes coleções de documentos textuais usando duas tarefas principais: *masked language modeling (MLM)*, onde palavras são ocultadas e o modelo deve prevê-las com base no contexto, e *next sentence prediction (NSP)*, que ensina o modelo a entender relações entre pares de frases. Como vantagem desta abordagem, sobretudo em virtude do método de treinamento, Patwardhan et al. (2023) cita a rica representação de padrões contextuais aprendida que leva a uma aplicação efetiva do BERT em diversas tarefas no campo do PLN.

Ocorre que o BERT, apesar de suas vantagens, possui algumas limitações que o tornam menos eficiente para tarefas específicas, como o cálculo de similaridade entre frases (Reimers & Gurevych, 2019). Primeiramente, o BERT exige um grande poder de processamento devido à sua arquitetura complexa e ao mecanismo de *self-attention*. Além disso, o BERT não foi concebido originalmente para a tarefa de similaridade entre frases, uma vez que sua finalidade precípua é a previsão de palavras ocultas (*masked language modeling*) e a compreensão de relações entre pares de frases (*next sentence prediction*). Como resultado, para comparar duas frases, o BERT precisa processá-las em conjunto, o que é computacionalmente ineficiente e pode representar um risco para aplicações em larga escala, como sistemas de recomendação.

1.2.5.2 Mecanismo de funcionamento do SBERT

Com o objetivo de adaptar o BERT para a realização de tarefas relativas à similaridade entre frases e busca semântica, o SBERT (*Sentence-BERT*) foi desenvolvido. Apresentado em agosto de 2019 no artigo *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks* (Reimers & Gurevych, 2019), o SBERT é uma variação do BERT projetada especificamente para tarefas de comparação de frases. Diferente do BERT, que processa pares de frases em conjunto, o SBERT gera *embeddings* independentes e semanticamente densos para cada frase, que podem ser comparados de forma eficiente por meio de métricas como a similaridade do cosseno. Conforme explanação constante à Figura 5, essa abordagem é viabilizada por uma arquitetura siamesa, onde o BERT é ajustado para produzir representações vetoriais contextualizadas. Soma-se a isso uma camada de *mean pooling* que auxilia na criação de um único vetor de *embeddings* que representa toda a frase.

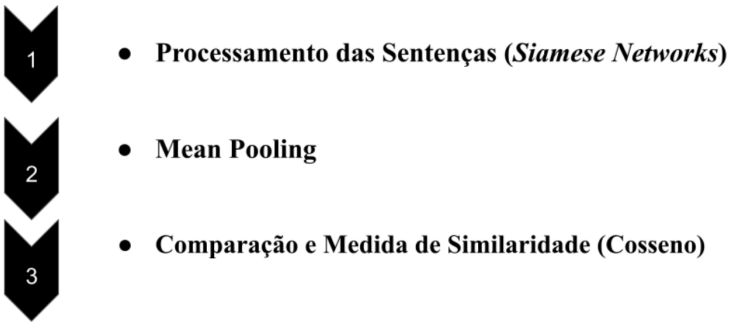

Figura 5: Funcionamento do SBERT.

No âmbito deste projeto, o SBERT é especialmente vantajoso, pois possui a capacidade de calcular a similaridade entre descrições de textos de maneira rápida e precisa, tornando-o interessante para aplicação em sistemas de recomendação baseados em conteúdo. Desta feita, justifica-se sua utilização que será demonstrada no capítulo da metodologia, comparando-o a um sistema com arquitetura mais simples baseado em *TF-IDF*.

1.3 Considerações finais sobre a revisão de literatura

A presente secção teve por escopo a consolidação dos principais fundamentos teóricos relacionados aos diferentes tipos de sistemas de recomendação e ao processamento de linguagem natural, com particular destaque para as etapas de pré-processamento textual e a evolução dos modelos de representação de texto em formato vetorial, indo desde as representações esparsas do paradigma *bag-of-words*, exemplificadas pelo **CountVectorizer** e pelo **TF-IDF (Term Frequency - Inverse Document Frequency)**, até abordagens modernas baseadas em *embeddings* densos e mecanismos de auto-atenção como o *BERT* e o *Sentence BERT*. Estes elementos revelaram-se essenciais para perceber de que forma diferentes técnicas influenciam a qualidade da informação extraída e, consequentemente, o desempenho dos sistemas de recomendação.

Deste enquadramento resulta, portanto, um fundamento sólido para o desenvolvimento das soluções propostas neste trabalho. Os conceitos explorados nesta secção orientam e justificam as decisões metodológicas adotadas no capítulo seguinte, onde são detalhados o desenho do sistema, as etapas de preparação dos dados, a implementação dos modelos testados e as medidas de similaridade que serão aplicadas aos vetores. A partir deste ponto, a metodologia descreve operacionalmente como os elementos teóricos identificados são integrados e aplicados aos sistemas de recomendação desenvolvidos.

2 Metodologia

Neste capítulo, apresentam-se com detalhes as estratégias metodológicas adotadas para a realização deste projeto. O trabalho tem como propósito desenvolver dois sistemas de recomendação de cursos utilizando técnicas de processamento de linguagem natural (PLN), buscando identificar com precisão a similaridade entre os itens. Para isso, são empregadas duas bases de dados: uma composta por cursos da plataforma Escola Virtual de Governo (EVG) com oitocentos e sete entradas e outra contendo livros em língua portuguesa extraídos da *Google Books API*, contendo quase trinta mil entradas.

Para isso, detalha-se a partir deste momento o processo de recolha, limpeza e estruturação dos dados, além dos métodos utilizados para a geração das recomendações. O estudo explora duas abordagens distintas para a vetorização textual: o TF-IDF, uma técnica mais tradicional, e o SBERT, baseado em redes neurais do tipo transformers. O objetivo é comparar o desempenho de ambas as metodologias e avaliar qual delas oferece recomendações mais precisas e relevantes.

Por conseguinte, apresenta-se o arcabouço técnico adotado, sobretudo no que concerne às ferramentas utilizadas para processamento dos dados e extração dos resultados. Por fim, são explicitados os critérios para avaliação do desempenho dos modelos e o prisma sob o qual os resultados são analisados.

2.1 Fluxo metodológico

A Figura 6 abaixo demonstra o fluxo metodológico utilizado ao longo deste projeto. Os passos serão detalhados a seguir em pontos específicos.

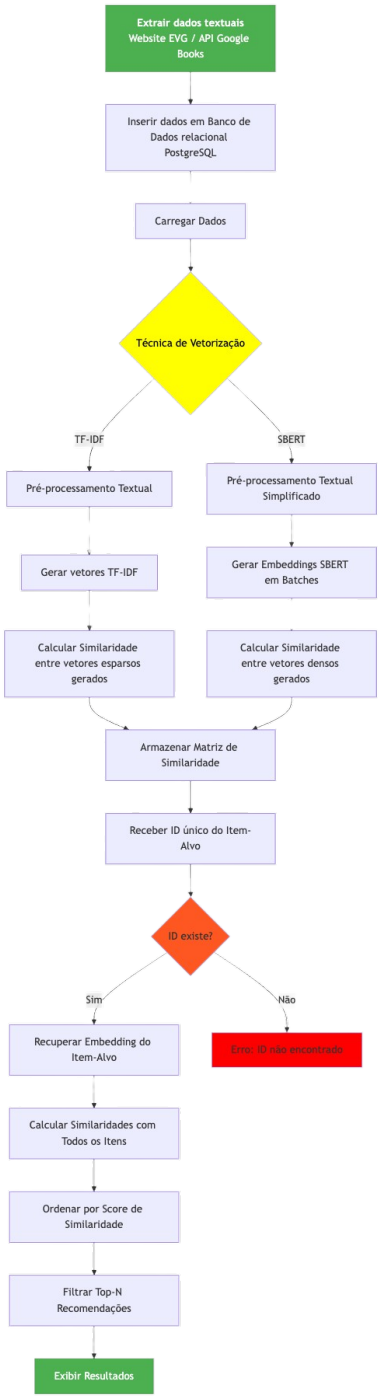

Figura 6: Fluxograma da metodologia utilizada

2.1.1 **Extração dos dados e *datasets* utilizados**

A etapa inicial do estudo envolve a extração e o armazenamento dos dados textuais necessários para o desenvolvimento e avaliação dos sistemas de recomendação. Para testar a eficácia dos modelos em

diferentes cenários, foram selecionados dois conjuntos de dados distintos: o primeiro, de menor volume, é composto pela matriz de cursos disponíveis no website da Escola Virtual de Governo (EVG); O segundo, significativamente maior, contém livros em língua portuguesa extraídos da *Google Books API*. A escolha desses *datasets* permite avaliar o desempenho dos sistemas tanto em bases mais pequenas quanto em conjuntos de dados mais extensos e diversos.

a) ***Dataset* "Cursos"**

A Escola Virtual de Governo (EVG) é uma plataforma de ensino à distância concebida e gerenciada pela Fundação Escola Nacional de Administração Pública (ENAP), vinculada ao Governo Federal brasileiro. Seu objetivo principal é oferecer capacitação gratuita e de qualidade para servidores públicos e cidadãos em geral, promovendo o desenvolvimento de competências relacionadas à gestão pública, políticas sociais, tecnologia e outras áreas do conhecimento estratégicas para o setor público.

Isto posto, o *dataset* da EVG foi escolhido devido à riqueza de informações textuais (*strings*) em suas descrições e a quantidade relativamente baixa de entradas, o que possibilitou uma análise mais controlada e detalhada das recomendações, tornando-o adequado para desenvolvimento dos protótipos dos sistemas. Esse conjunto de dados foi extraído em 05/02/2025, diretamente do *website* da EVG, em formato ".csv". Ele contém oitocentos e sete linhas e treze colunas. A seguir, apresenta-se o dicionário completo de variáveis deste conjunto de dados:

- id_curso: Inteiro, Identificador único do curso na plataforma;
- nome_curso: *String*, Título da capacitação;
- eixos_tematicos: *String*, Subdivisão do eixo ao qual a capacitação pertence;
- competencias: *String*, Competencias profissionais abrangidas pela capacitação;
- certificador: *String*, Ente responsável pela aplicação do curso;
- conteudista: *String*, Ente responsável pela elaboração dos conteúdos do curso;
- carga_horaria: Inteiro, Medida em horas. Carga do curso;
- disponibilidade_dias: Inteiro, Em quanto tempo o curso pode ser feito. Medido em dias;
- tipo_oferta: *String*, Quem pode se inscrever na capacitação;
- apresentacao: *String*, Texto de apresentação do curso na plataforma;
- publico_alvo: *String*, Público-Alvo do curso;
- conteudo_programatico: *String*, Módulos ou capítulos que compõem a capacitação;
- data_lancamento: *Datetime*, Data em que a capacitação foi lançada no site;

É importante destacar que, das 13 variáveis do conjunto de dados, apenas quatro foram utilizadas no desenvolvimento dos sistemas de recomendação, conforme apresentado na Tabela 4. A escolha dessas colunas se deve à necessidade de maximizar o uso de informações textuais disponíveis sobre cada curso, ampliando a base de conhecimento para a recomendação.

id curso	nome curso	apresentacao	conteudo programatico
1328	Processamento de Linguag...	Descubra como o Processa...	Módulo 1 - Análise de te...
1329	Treinamento Censo Superi...	O Censo da Educação Supe...	Módulo 1 - O Inep e os C...
1276	Armamento Institucional	Nesse curso você conhece...	Módulo 1 - Normas Gerais...
1231	Elaboração de editais em...	O curso capacita profiss...	Módulo Introdução ;Módu...
936	GTFS, GPS e bilhetagem e...	Neste curso, você conhec...	Módulo 1: Gestão de dado...
1030	Instrumentos de Desenvol...	Descubra quais são as at...	Módulo 1 - Fazendo as co...
1138	Primeiros passos com o M...	Neste curso, você conhec...	Como criar um documento ...
860	Compras sustentáveis e a...	Este curso tratará dos p...	Módulo 1: Sustentabilida...

Tabela 4: Amostra de 8 linhas e 4 colunas da tabela de cursos

Dado que o sistema de recomendação é inteiramente baseado em dados textuais (*strings*), colunas numéricas como carga horária, disponibilidade do curso e data de lançamento foram descartadas por não agregar relevância semântica. Além disso, seu uso poderia introduzir ruídos desnecessários no modelo, prejudicando a qualidade das recomendações.

Entre as variáveis textuais, foram selecionadas apenas aquelas que fornecem informações extensas e únicas sobre os cursos: nome, apresentação e conteúdo programático. Essas colunas contêm descrições detalhadas sobre os temas abordados, facilitando a identificação de palavras-chave e relações semânticas. A coluna nome_curso, por sua vez, foi utilizada apenas para identificar cada capacitação em conjunto com seu respectivo ID.

Por fim, visando aproximar-se ao máximo de um ambiente de produção real em que os dados de *input* utilizados na emissão das recomendações estarão armazenados em algum tipo de *database*, as informações dos cursos foram armazenadas em um banco de dados relacional *PostgreSQL*, tendo sido utilizada a linguagem *python* e as bibliotecas *psycopg2-binary* e pandas para conexão com o banco e ingestão dos dados.

b) ***Dataset* "Livros"**

O segundo e mais volumoso conjunto de dados utilizado para desenvolver e testar os sistemas de recomendação traz informações sobre aproximadamente trinta mil obras literárias em Português-BR e seus campos foram extraídos por meio de consumo da *API* Google Books com uso da linguagem *python* e das bibliotecas *requests* e pandas. Assim como o antecessor, este conjunto de dados traz consigo um campo textual em específico que possui grande relevância para o desenvolvimento deste projeto: a coluna de descrição do livro.

O processo para extração destes dados, no entanto, não fora tão simples e direto como aquele percorrido para o *dataset* anterior. A *Google Books API* tem suas requisições baseadas no código ISBN que identifica cada livro, conforme demonstrado na Figura 7. Logo, foi necessário um trabalho preliminar de pesquisa para localizar uma lista com todos os ISBNs disponíveis, a qual fora encontrada e extraída em formato .parquet na plataforma Hugging Face, para só então adaptar a função em *python* para iterar sobre esta lista e construir o *dataframe* com os dados consolidados.

Ocorre que mostrou-se inviável encaminhar requisições à *API Google Books* para todos os ISBNs listados, pois excederia o limite de requisições definida na documentação da ferramenta, além de gerar custos de processamento e armazenamento desnecessários, pois a maior parte daqueles dados não interessam ao projeto. Soma-se a isso o facto de que, conforme evidenciado no *output* da Figura 7, muitos dos identificadores presentes na lista retromencionada diziam respeito a obras em outras línguas que não a língua portuguesa, o que não era desejável uma vez que o cerne deste projeto é desenvolver sistemas de recomendação adaptados à língua portuguesa.

```python
## Assim funciona a requisição da API do google books:
import requests

isbn = '9780545010221'
url = 'https://www.googleapis.com/books/v1/volumes?q=isbn:'+isbn
r = requests.get(url)
data = r.json()

#verificar se há itens retornados na resposta
if 'items' in data:
    # verificar se existem dados sobre o ISBN apontado
    if 'volumeInfo' in data['items'][0]:
        title = data['items'][0]['volumeInfo']['title']
        authors = data['items'][0]['volumeInfo']['authors']
        description = data['items'][0]['volumeInfo'].get('description', 'No description available')
        print("Title:", title)
        print("Authors:", authors)
        print("Description:", description)
    else:
        print("No volume information available")
else:
    print("No items found in the response")
```

Python

```
Title: Harry Potter and the Deathly Hallows
Authors: ['J. K. Rowling']
Description: "The final adventure in J.K. Rowling's phenomenal, best-selling Harry Potter book series"--Provided by publisher.
```

Figura 7: Exemplo de função para extração da Google Books API com uso de python

Isto posto, a estratégia utilizada para solucionar os problemas apontados anteriormente foi utilizar a biblioteca *langdetect*, desenvolvida pelo Google originalmente em java e adaptada em sua integralidade para *python*, a qual detecta automaticamente, baseando-se no título textual do livro associado ao seu código ISBN, a linguagem a qual aquela obra dizia respeito. Após a sua aplicação, a lista de ISBNs que anteriormente possuía 28 milhões de entradas, passou a conter pouco mais de 100.000 entradas.

Por conseguinte, os dados foram efetivamente extraídos do google books apenas para os ISBNs de interesse tendo sido transformados e limpos com uso da linguagem *SQL* e inseridos em uma nova tabela denominada "livros", armazenada no mesmo *database* onde já estava a tabela de cursos, de modo a centralizar a fonte de dados e padronizar o uso do método "carrega_dados" para ambas as tabelas na classe Recommender, a qual será analisada ao fim deste capítulo. A Tabela 5 traz uma amostra da composição final da tabela de livros.

isbn	title	authors	description
9788500001710	177 maneiras de enlouquecer um	Margot Saint-Loup	177 dicas de Margot Sa...
9788500002533	Da cabeça aos pés	Marilda Castanha	'Da cabeça aos pés' é ...
9788500002540	Fada fofa, onça-fada!	Sylvia Orthof	Esta é mais uma divert...
9788500003042	Homem mais rico da Babilônia,O	George Samuel Clason	Traz orientações atrav...
9788500005626	Pai, Me Compra Um Amigo?	Pedro Bloch	Esta é a história de B...
9788500005763	Memórias de um sargento de milícias	Manuel António de Almeida	A linguagem popular e ...
9788500006593	No Final Do Seculo	Nathan P. Gardels	Uma coleção de ensaios...
9788500007385	Memórias de Ramses o grande	CLAIRE LALOUETTE	'As Memórias de Ramsés...
9788500007453	Viagens de Marco Polo, As	Carlos Heitor Cony	O veneziano Marco Polo...
9788500007606	Que cara tem o Brasil?	Mônica Pimenta Velloso	Por que será que o ver...

Tabela 5: Amostra de 10 linhas da tabela de livros

2.1.2 **Pré-processamento**

De seguida, procedeu-se à limpeza e transformação dos dados textuais, preparando-os para a vetorização e subsequente inserção nos modelos de recomendação. As minúcias técnicas utilizadas nesta etapa serão expostas na secção específica que versa sobre a construção dos sistemas de recomendação. No entanto, de forma geral, a metodologia aplicada na limpeza e transformação da coluna compilado_textual incluiu diversas etapas essenciais para garantir a qualidade e consistência dos dados antes da etapa de vetorização, dentre as quais destacam-se:

- Remoção de *stopwords* e extensão da lista pré-definida pela biblioteca nltk para abarcar casos específicos identificados durante o processo de exploração dos dados;
- Normalização para que todas as palavras estivessem em letras minúsculas;
- Remoção de acentuação gráfica;
- Remoção de sinais de pontuação e caracteres especiais;

Quanto aos demais métodos de pré-processamento comumente aplicados no âmbito do PLN, como a lematização e o stemming (Jurafsky & Martin, 2025), é importante destacar que, durante os testes exploratórios, tais abordagens não se mostraram eficazes para os conjuntos de dados utilizados. Por esse motivo, optou-se por não incorporá-las nas funções de limpeza dos dados aplicadas às versões finais dos sistemas de recomendação desenvolvidos.

2.1.3 **Compilação dos dados**

Após a recolha e processamento dos conjuntos de dados, o próximo passo envolve os processos de extração e engenharia de *features*. No contexto deste projeto, isso significa selecionar as variáveis mais relevantes para o desenvolvimento dos sistemas de recomendação, além de criar uma nova variável, chamada "compilado_textual", que reunirá as informações textuais mais importantes sobre cada item em ambos os conjuntos de dados.

A criação da coluna "compilado_textual", que centraliza as informações textuais essenciais para os sistemas de recomendação, envolveu a agregação das seguintes colunas[2] para cada conjunto de dados:

- **Cursos:** nome, apresentação e conteúdo programático;
- **Livros:** título e descrição;

2.1.4 **Medida de semelhança: a similaridade do cosseno e porquê utilizá-la.**

Optou-se por adotar no âmbito do presente projeto a similaridade do cosseno como métrica objetiva de aferição da semelhança entre os vetores dos itens, haja vista ser uma das métricas mais comuns e úteis no campo do PLN (Jurafsky & Martin, 2025) por ser indiferente à magnitude dos vetores, com foco na comparação da direção destes.

Este parâmetro de similaridade mede o ângulo entre dois vetores em um espaço multidimensional, indicando o grau de similaridade entre eles, ou seja, o quão próximos eles são semânticamente. Esta comparação é fundamental para que, dado determinado item, determine-se quais são os mais próximos a ele e, portanto, mais relevantes, os quais deverão ser recomendados ao utilizador.

[2]A descrição de cada coluna consta no dicionário de variáveis da secção Extração dos dados e datasets utilizados

Considerando os vetores associados às palavras z e w, a similaridade do cosseno entre elas pode ser percebida como:

$$\text{cos-similarity}(z, w) = \frac{z \cdot w}{\|z\|\|w\|} = \cos(\alpha)$$

onde, α é o ângulo entre os vetores z e w.

Ressalta-se ainda que o valor da similaridade do cosseno para o caso em tela varia entre 0 e 1, onde 1 indica identidade e 0 indica que os vetores são ortogonais(nenhuma relação de similaridade). Destaca-se ainda a natureza não-negativa dos vetores gerados no caso do TF-IDF, por exemplo, onde são representadas frequências de palavras ponderadas, o que impede que a similaridade do cosseno seja menor que 0.

No que tange ao modelo de *Sentence Embeddings* (SBERT), apesar de ser possível que os vetores de *embeddings* assumam valores negativos, eles são projetados para captar apenas relações semânticas positivas. Aqui o cálculo da similaridade do cosseno entre esses *embeddings* é feito em um espaço onde a semântica é representada por proximidade e não por oposição. Ademais, nesta técnica de geração de *embeddings* os vetores extraídos também são normalizados, a exemplo do que ocorre com o TF-IDF, para garantir que a magnitude dos vetores não terá influência sobre a similaridade.

Por conseguinte, a etapa de normalização dos vetores durante a extração destes também tem por escopo garantir que o ângulo entre eles nunca seja superior a 90°, de modo a manter o cosseno restrito ao intervalo entre [0,1], conforme abordagem outrora adotada para o TF-IDF.

A Figura 8 traz um exemplo claro do que fora explicitado até aqui: nela representa-se uma amostra de 10 entradas da matriz de similaridade do cosseno gerada ao comparar-se os vetores de todos os itens entre si, matriz esta que constitui o ponto central dos sistemas de recomendação ora propopostos. É possível observar que determinado item possui identidade (similaridade cosseno = 1) consigo mesmo, e que os valores das similaridade são menores à medida que os comparamos com os demais itens da matriz.

Figura 8: Heatmap das 10 primeiras entradas da matriz de similaridade do cosseno, que possui dimensão total de 807 × 807 para o dataset de cursos.

2.1.5 **Outras medidas de similaridade**

Embora a similaridade do cosseno tenha sido a métrica adotada no âmbito deste trabalho, outras medidas de distância, como a distância Euclidiana e a distância de Manhattan, também podem ser aplicadas ao caso em tela (Reimers & Gurevych, 2019). No entanto, essas métricas podem apresentar limitações significativas no presente contexto de sistemas de recomendação baseados em **vetores de alta dimensionalidade**.

No caso da matriz TF-IDF gerada pelo primeiro sistema de recomendação, estamos a lidar com uma matriz esparsa, conforme ilustrado na amostra da Tabela 3. Para o conjunto de dados menor (cursos), a matriz possui cerca de sete mil colunas, enquanto para o conjunto de dados maior (livros), ela ultrapassa as cem mil colunas. Isto ocorre porque cada palavra única no vocabulário se transforma numa coluna na matriz. Já para as recomendações baseadas em vetores de *embeddings* densos gerados pelo modelo SBERT *paraphrase-multilingual-MiniLM-L12-v2* (Reimers & Gurevych, 2019), o qual foi consumido diretamente da plataforma Hugging Face e aplicado em seus pesos originais e sem nenhum tipo de ajuste fino, o espaço vetorial é restrito a trezentos e oitenta e quatro dimensões. Embora isto seja consideravelmente menor que o espaço TF-IDF, ainda pode ser excessivo para métricas que consideram a distância absoluta entre os pontos, em vez de se focarem no ângulo e na direcção entre os vetores.

Isto posto, resta claro que um dos principais desafios no presente contexto diz respeito ao enfrentamento da elevada dimensionalidade do espaço vetorial, fenómeno que afeta métricas sensíveis à magnitude dos vetores, como a Euclidiana, mas tende a apresentar impacto reduzido na similaridade do cosseno, que se concentra na orientação relativa dos vetores, conforme discutido por Aggarwal et al. (2001).

Por fim, embora a similaridade do cosseno tenha sido escolhida para extração dos resultados deste trabalho, serão apresentados, no capítulo seguinte, os resultados dos testes comparativos ao se utilizar a similaridade do cosseno, a distância Euclidiana e a distância de Manhattan. O objetivo é analisar as principais diferenças nos resultados gerados por estas diferentes métricas.

2.1.6 Construção dos sistemas de recomendação: desenvolvimento das classes TfidfRecommender e SBERTRecommender

Após a obtenção e transformação dos dados, passou-se à construção efetiva dos sistemas de recomendação. Optou-se mais uma vez pelo uso da linguagem de programação *python* e do paradigma de orientação a objetos para facilitar a manutenção, organização e reutilização dos códigos fonte dos sistemas de recomendação. Foram implementadas duas classes distintas – TfidfRecommender e SBERTRecommender –, cada uma encapsulando o fluxo completo de seu respectivo algoritmo.

A partir deste momento, cada sistema de recomendação será analisado separadamente, dado que a natureza dos métodos utilizados para a extração e representação dos vetores difere significativamente entre eles. Inicialmente, será apresentada a implementação do modelo baseado em TF-IDF, uma abordagem mais tradicional e rudimentar para recomendação textual. Em seguida, será detalhada a construção do modelo baseado em SBERT e *transformers*, que utiliza representações semânticas mais avançadas para capturar significados contextuais nos textos.

- **TfidfRecommender**

A Figura 9 demonstra a estrutura da classe responsável pelo sistema de recomendação baseado em TF-IDF, a saber:

```python
import pandas as pd
import re
import nltk
from sqlalchemy.engine import create_engine
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import linear_kernel
from tqdm import tqdm

class TfidfRecommender:
    def __init__(self, DB_USER,
                 DB_PASS,
                 DB_NAME,
                 DB_HOST, emb_cols, id_col: str, item_name_col: str, stopwords_extra=None):
        self.DB_USER = DB_USER
        self.DB_PASS = DB_PASS
        self.DB_HOST = DB_HOST
        self.DB_NAME = DB_NAME
        self.engine = create_engine(f'postgresql://{self.DB_USER}:{self.DB_PASS}@{self.DB_HOST}:5432/{self.DB_NAME}')

        ##cols que serao usadas p/ extrair texto p/ vetores de embeddings
        self.emb_cols = emb_cols
        self.id_col = id_col
        self.item_name_col = item_name_col
        self.stop_words = set(stopwords.words('portuguese'))

        # add stopwords personalizadas se fornecidas como argumento
        if stopwords_extra is not None:
            self.stop_words.update(stopwords_extra)

        #Inicialização do Tfidf
        self.tfidf_vectorizer = TfidfVectorizer()

        # Dados e manipulação dos vetores
        self.df_text = None
        self.embeddings = None
        self.cosine_sim = None
        self.indice = None
        self.tf_matrix = None
```

Figura 9: Construtor da Classe TfidfRecommender

- Bibliotecas e versão do *python* utilizadas:
    - Python 3.12.3
    - Pandas: manipulação e estruturação dos dados em *DataFrames*.
    - re: expressões regulares para limpeza e pré-processamento de texto.
    - nltk: utilizado para processamento de linguagem natural, especialmente remoção de *stopwords* no caso em tela.
    - Sqlalchemy: criação de conexão com banco de dados *PostgreSQL* para consulta dos dados.
    - Sklearn: inicialização do TF-IDF, conversão de textos em vetores numéricos e cálculo da similaridade do cosseno via função linear_kernel.
    - tqdm: exibição de barra de progresso para acompanhar o processamento.
- Parâmetros ao instanciar a classe:
    - **DB_USER, DB_PASS, DB_NAME, DB_HOST:** dados relativos ao banco de dados onde as informações relevantes para desenvolvimento do modelo encontram-se.
    - **emb_cols:** colunas textuais que serão utilizadas como fonte de informação para o modelo. Este é o parâmetro mais importante, vez que aqui o operador deve apontar aquelas colunas de interesse no *dataset* que serão compiladas e servirão de base para extração dos vetores de representação textual.
    - **id_col:** coluna na base de dados que contém a informação a respeito do identificador único de cada item a ser processado e recomendado.
    - **item_name_col:** coluna que contém o título/nome do item a ser processado e recomendado adiante. Importante pois seu texto será adicionado às informações das colunas de emb_cols com vistas a maximizar os dados textuais sobre cada um dos itens.
    - **stopwords_extra:** faculta ao operador do sistema de recomendação a opção de adicionar ainda mais *stopwords* à etapa de limpeza textual que façam sentido para aquele domínio específico, as quais serão adicionadas às *stopwords* previamente definidas pela biblioteca nltk e aplicadas à etapa de limpeza textual.
- **Método “carrega_dados”:**
    - Responsável pela abertura do banco de dados relacional e carregamento das colunas textuais de interesse para o modelo. Utiliza-se para tanto os dados das variáveis inseridos pelo operador ao instanciar a classe TfidfRecommender (emb_cols + item_name_col).
- **Método “limpar_e_compilar_texto”:**
    - Executa a etapa de limpeza e preparação dos dados textuais das colunas que serão vetorizadas. Por se tratar do sistema de recomendação baseado em TF-IDF, o qual requer minuciosa etapa

de limpeza dos dados para evitar-se redundâncias e ruídos que sobrecarreguem o sistema ao estender em demasia o vocabulário[3], aplicam-se aos dados as seguintes transformações: remoção de acentos, sinais de pontuação e caracteres especiais; Alteração para que todas as palavras estejam em letras minúsculas; Remoção das *stopwords* pré-definidas pela biblioteca nltk e aquelas *stopwords* adicionais definidas pelo operador em *stopwords_extra*, se houverem; Junção do texto de todas as colunas textuais apontadas pelo operador em uma coluna chamada "compilado textual", a qual trará o texto do nome do item acrescido do texto das demais colunas referentes ao texto de apresentação, eventual conteúdo programático e descrição. Esta coluna será utilizada como base para extração dos vetores que descrevem aquele item na matriz TF-IDF. Ressalta-se ainda que a nova variável criada prescinde de qualquer tratamento por já ter sido formada pela agregação de colunas limpas e transformadas.

- Considerando o fragmento original do texto descritivo do livro "As Mil e Uma Noites", observa-se na Tabela 6 um exemplo da aplicação do método em comento à frase exemplificativa a seguir:

  "O livro apresenta as fábulas das 'Mil e uma Noites', com seu colorido oriental"

Etapa de Pré-Processamento	Resultado
Frase Original	O livro apresenta as fábulas das 'Mil e uma Noites', com seu colorido oriental
Minúsculas	o livro apresenta as fábulas das 'mil e uma noites', com seu colorido oriental
Remoção de Acentos	o livro apresenta as fabulas das 'mil e uma noites', com seu colorido oriental
Remoção de Stopwords	livro apresenta fabulas 'mil uma noites' colorido oriental
Remoção de Pontuação Extra	livro apresenta fabulas mil e uma noites colorido oriental

Tabela 6: Aplicação dos estágios de pré-processamento sobre a frase exemplificativa.

- **Método "gerar_vetores":**
  - Recebe os dados da coluna "compilado_textual" limpos e compilados e aplica sobre eles o método tfidf_vectorizer da biblioteca scikit-learn, de modo a transformá-los nos vetores de representação textual numéricos que serão utilizados como base para a etapa de aplicação da similaridade do

[3]Vocabulário refere-se ao conjunto de termos únicos extraídos de todos os textos analisados, onde cada termo corresponde a uma coluna na matriz TF-IDF.

cosseno e posterior extração das recomendações;

- Calcula a matriz TF-IDF, onde cada linha representa um item (livro ou curso) e cada coluna representa um termo do vocabulário, atribuindo pesos conforme a importância relativa do termo no conjunto de dados. Isso permite que textos com palavras frequentes, mas pouco discriminativas, tenham menos peso, enquanto termos mais raros e relevantes recebam maior influência na representação vetorial.
- Utiliza a técnica de *linear kernel* para calcular a similaridade do cosseno entre os vetores da matriz TF-IDF. Esse cálculo mede o grau de proximidade entre os itens, atribuindo valores entre 0 e 1, onde 1 indica máxima similaridade e 0 indica ausência de relação.
- Armazena a matriz de similaridade do cosseno para que ela possa ser utilizada posteriormente no método de recomendação, garantindo que as consultas sejam feitas de forma eficiente sem necessidade de recalcular as similaridades a cada requisição.
- Cria um índice (self.indice), que mapeia os identificadores únicos dos itens (id_col) aos seus respectivos índices na matriz de similaridade, permitindo consultas rápidas para recomendações futuras.
- Exibe informações sobre a geração dos *embeddings*, incluindo o *shape* da matriz TF-IDF e amostras das matrizes TF-IDF e de similaridade do cosseno, para fins de depuração e análise da qualidade das representações vetoriais.

- **Método "recomendar":**
  - Responsável por exibir as recomendações de itens mais similares ao item de entrada (*input*), utilizando a matriz de similaridade do cosseno previamente calculada. Ele recebe um ID de item e retorna os N itens mais similares, com base na proximidade vetorial entre os textos da matriz TF-IDF.
  - Em primeira medida, recebe um ID de item e consulta a matriz de similaridade do cosseno previamente gerada para encontrar os itens mais similares ao item de entrada.
  - A seguir, a partir do ID do item fornecido pelo utilizador, o método busca seu índice na matriz TF-IDF utilizando self.indice[id_item]. O nome do item original é recuperado da base de dados (self.df_text) para ser exibido na saída, facilitando a interpretação do utilizador.
  - Obtém a linha correspondente ao item de entrada na matriz de similaridade do cosseno, recuperando os valores de similaridade em relação a todos os outros itens.
  - Ordena os itens de maneira decrescente quanto à similaridade, garantindo que os mais relevantes apareçam primeiro (remove a própria entrada da lista para evitar que o item recomende a si mesmo devido a similaridade máxima).

- Seleciona os n itens mais similares, conforme definido pelo utilizador, garantindo que a recomendação seja ajustável às necessidades do sistema.
- Retorna um *DataFrame* contendo os IDs, nomes e *scores* de similaridade dos itens recomendados, de modo a facilitar a análise da qualidade daquilo que foi recomendado como sendo similar ao item de entrada.
- Exibe as recomendações de forma estruturada, apresentando o nome do item original e os itens sugeridos com seus respectivos *scores*.

```python
def recomendar(self, id_item, top_n=3):
    """Recomenda itens similares com base no ID do item de entrada."""
    if self.cosine_sim is None:
        raise ValueError("A similaridade ainda não foi calculada. Certifique-se de que os embeddings foram gerados.")

    idx = self.indice[id_item]
    item_name = self.df_text.loc[self.df_text[self.id_col] == id_item, self.item_name_col].values[0].title()

    # Calcular scores de similaridade
    sim_score = list(enumerate(self.cosine_sim[idx]))
    sim_score = sorted(sim_score, key=lambda x: x[1], reverse=True)[1:top_n+1]
    sim_index = [i[0] for i in sim_score]

    recomendacoes = pd.DataFrame({
        'Item Recomendado': self.df_text[self.item_name_col].iloc[sim_index],
        'Similaridade': [score[1] for score in sim_score]
    }).reset_index(drop=True)

    print(f"As recomendações mais similares ao item '{item_name}' são:\n")
    return print(recomendacoes)
```

Figura 10: Demonstração da função que extrai as recomendações

- **SBERTRecommender**

A Figura 11 apresenta a estrutura da classe responsável pelo sistema de recomendação baseado em SBERT, a saber:

```python
import pandas as pd
from sqlalchemy.engine import create_engine
from sentence_transformers import SentenceTransformer, util
import torch
import nltk
from nltk.corpus import stopwords
from tqdm import tqdm
import os
import warnings
from dotenv import load_dotenv

warnings.filterwarnings("ignore")
load_dotenv()

class SBERTRecommender:
    def __init__(self, DB_USER,
                 DB_PASS,
                 DB_NAME,
                 DB_HOST,
                 model_name: str,
                 emb_cols,
                 id_col: str,
                 item_name_col: str):
        self.DB_USER = DB_USER
        self.DB_PASS = DB_PASS
        self.DB_HOST = DB_HOST
        self.DB_NAME = DB_NAME
        self.engine = create_engine(f'postgresql://{self.DB_USER}:{self.DB_PASS}@{self.DB_HOST}:5432/{self.DB_NAME}')
        ##chama o modelo
        self.model = SentenceTransformer(model_name)
        ##chama cols que serao usadas p/ extrair texto p/ vetores de embeddings e stopwords em PT
        self.emb_cols = emb_cols
        self.id_col = id_col
        self.item_name_col = item_name_col
        self.stop_words = set(stopwords.words('portuguese'))
        self.embeddings = None
        self.df_text = None
```

Figura 11: Construtor da Classe SBERTRecommender

- Bibliotecas e versão do *python* utilizadas:
    - Python 3.12.3
    - Pandas: manipulação e estruturação dos dados em *DataFrames*.
    - Sentence-Transformers: geração de *embeddings* a partir de textos, utilizando modelos pré-treinados como SBERT.
    - nltk: utilizado para processamento de linguagem natural, especialmente remoção de *stopwords* no caso em tela.
    - Sqlalchemy: criação de conexão com banco de dados *PostgreSQL* para consulta dos dados.
    - Torch: suporte ao processamento vetorial e operações matriciais com *embeddings*.
    - tqdm: exibição de barra de progresso para acompanhar o processamento dos *embeddings*.
    - dotenv: carregamento seguro de variáveis de ambiente sensíveis, como credenciais do banco de dados.
    - warnings: supressão de mensagens de aviso durante a execução do código.
    - os: manipulação de variáveis de ambiente e configurações do sistema.
- Parâmetros ao instanciar a classe:
    - **DB_USER, DB_PASS, DB_NAME, DB_HOST:** dados relativos ao banco de dados onde as informações relevantes para desenvolvimento do modelo encontram-se.
    - **emb_cols:** colunas textuais que serão utilizadas como fonte de informação para o modelo. Aqui o operador deve apontar as variáveis de interesse no *dataset* que serão compiladas e servirão de base para extração dos vetores de representação textual.
    - **id_col:** coluna na base de dados que contém a informação a respeito do identificador único de cada item a ser processado e recomendado.
    - **item_name_col:** coluna que contém o título/nome do item a ser processado e recomendado adiante. Importante pois seu texto será adicionado às informações das colunas de emb_cols com vistas a maximizar os dados textuais sobre cada um dos itens.
    - **model_name:** atalho para o modelo a ser utilizado para geração dos *embeddings* textuais. Permite flexibilidade na escolha de diferentes versões do modelo, a depender do domínio da aplicação. **No caso em tela utilizou-se o modelo *"paraphrase-multilingual-MiniLM-L12-v2"* da biblioteca Sentence Transformers**, entretanto outros modelos poderiam ter sido aplicados.
- **Método "carrega_dados"**:
    - Responsável por estabelecer a conexão com o banco de dados *PostgreSQL* e carregar as colunas textuais de interesse definidas pelo operador ao instanciar a classe. Ele executa uma *query SQL* para extrair os dados das colunas especificadas em "emb_cols" e "item_name_col", e define "id_col" como o identificador único de cada item.

- **Método “limpa_dados”**:
    - Método responsável pela limpeza e pré-processamento dos textos antes da geração dos *embeddings* densos pelo modelo. Aplica técnicas padronizadas de tratamento textual, removendo ruídos e normalizando os dados para melhorar a qualidade das representações geradas. Destaca-se o facto de ser uma limpeza menos complexa que aquela realizada no sistema de recomendação anterior, haja vista a menor sensibilidade às nuances de padronização de vocabulário neste modelo.
- **Método “embeddings_extract”**:
    - Este método é essencial para o funcionamento do sistema, pois é responsável pela geração dos vetores numéricos densos a partir dos textos processados na etapa de anterior. Utiliza-se um modelo de *deep learning* para converter os textos em *embeddings* densos, de modo a permitir cálculos de similaridade eficientes.
    - Os vetores de *embeddings* resultantes são empilhados em uma matriz 2D usando “torch.stack”, consolidando-os em um formato adequado para cálculos de similaridade.
- **Método “recomendar_itens”**:
    - Método que realiza efetivamente a recomendação de itens relevantes com base na matriz de similaridade do cosseno entre os *embeddings* previamente gerados. Trata-se da etapa final do fluxo de recomendação baseado em vetores densos.

```python
def recomendar_itens(self, id_item, top_n: int = 3):
    """Recomenda itens similares com base no ID do item de entrada."""
    id_item = str(id_item)

    if id_item not in self.df_text[self.id_col].values:
        print(f"ID {id_item} não encontrado na base de dados.")
        return None

    df_reset = self.df_text.reset_index()
    indice = pd.Series(df_reset.index, index=df_reset[self.id_col].astype(str))
    idx = indice[id_item]

    sim_score = util.cos_sim(self.embeddings[idx], self.embeddings).squeeze()
    sim_score = sim_score.tolist()

    sim_score = sorted(enumerate(sim_score), key=lambda x: x[1], reverse=True)[1:top_n+1]
    sim_index = [i[0] for i in sim_score]

    recomendacao = pd.DataFrame({
        'Item Recomendado': self.df_text[self.item_name_col].iloc[sim_index],
        'Sim_Coss': [score[1] for score in sim_score]
    }).reset_index(drop=True)

    if recomendacao.empty:
        print("Nenhuma recomendação encontrada.")
    else:
        print(f"As recomendações mais similares ao item '{self.df_text.loc[idx, self.original_nome_item]}' são:\n")

    return recomendacao
```

Figura 12: Função responsável pela extração das recomendações

2.1.7 **Metodologia de avaliação das recomendações**

Para perceber de maneira clara o método a ser utilizado para avaliar as recomendações geradas no âmbito deste projeto, há de se delinear ainda que de maneira superficial as diferenças entre a aprendizagem de máquina supervisionada e não supervisionada.

Na aprendizagem de máquina supervisionada, os modelos são treinados com dados rotulados, o que permite a avaliação quantitativa acerca da qualidade dos *outputs* gerados (Sokolova & Lapalme, 2009). Neste tipo de abordagem utilizam-se métricas como sensibilidade e precisão para avaliar o desempenho dos modelos com base em respostas conhecidas.

Em contrapartida, na aprendizagem de máquina não supervisionada, não existem dados rotulados ou respostas pré-definidas para aplicar-se ao treinamento dos modelos. Desta feita, neste tipo de abordagem os modelos devem identificar padrões intrínsecos aos dados, como agrupamentos ou relações de similaridade entre as variáveis ou as observações (James et al., 2013). Ressalta-se que neste contexto a aplicação de métricas de avaliação quantitativas como aquelas citadas no parágrafo anterior torna-se inviável, pois não existem dados rotulados para treinamento dos modelos.

Ante o exposto, **os sistemas de recomendação propostos neste projeto adotam uma abordagem de avaliação qualitativa e não supervisionada**, dispensando interações utilizador-item rotuladas ou juízos explícitos de relevância em um primeiro momento. As recomendações são geradas através da comparação de representações vetoriais (*embeddings*) extraídas de descrições textuais, **utilizando a similaridade semântica como principal critério.** Esta abordagem visa garantir escalabilidade e facilitar a implementação em diferentes organizações, eliminando a necessidade de anotações dispendiosas. Diferentemente da filtragem colaborativa, que exige grandes volumes de dados históricos e *feedback* explícito dos utilizadores para produzir recomendações eficazes, a abordagem adotada baseia-se, à partida, exclusivamente no conteúdo textual, **permitindo a sua aplicação mesmo em cenários com dados limitados.**

A proposta deste projeto diz respeito a um ponto de partida eficiente para oferecer sugestões relevantes somente com uso de descrições textuais dos itens, sem depender da existência prévia de dados do utilizador para tal. No entanto, para mitigar a falta de monitorização de desempenho automatizada, recomenda-se que, numa fase posterior, o sistema capte ao menos o *feedback* implícito — cliques, tempo de visualização, taxas de conversão — conforme proposto por Jannach et al. (2018). Esses sinais poderão ser usados para calcular métricas de precisão e sensibilidade e para detetar desvios na qualidade das recomendações, permitindo que o sistema evolua gradativamente para um modelo híbrido leve, personalizando-se de acordo com as preferências e comportamentos de cada utilizador.

Por fim, no âmbito deste projeto, haja vista tratar-se de desenvolvimento de sistemas de recomendação não supervisionados, a avaliação qualitativa daquilo que está a ser recomendado priorizou os seguintes aspetos:

- Análise manual (haja vista o pressuposto da ausência à partida de dados históricos e de *feedback* do utilizador) da relevância temática existente entre o item base e aqueles recomendados a partir dele, considerando-se a relevância semântica e, principalmente, a coerência contextual;
- Capacidade de identificação pelos vetores das relações de dependência entre determinados itens (p.ex., capacidade de recomendação de sequências);
- Análise e discussão crítica de casos onde os modelos apresentam falhas, sobretudo no que diz respeito à polissemia e não captação de contexto;

3 Apresentação dos resultados

A avaliação proposta neste projeto encontra a sua validação prática na análise dos resultados obtidos pelos sistemas de recomendação baseados em *TF-IDF* e *SBERT*. O presente capítulo estrutura-se em duas partes principais, correspondentes aos *datasets* analisados: (1) um conjunto reduzido de cursos da plataforma EVG, que permite uma inspeção detalhada e interpretável das recomendações geradas; e (2) um *corpus* extenso de mais de vinte e nove mil livros extraídos da *Google Books API*, onde se testa a robustez e escalabilidade do método em larga escala. Em ambos os cenários, os resultados são apresentados em estruturas de *dataframe* que têm por objetivo apresentar os contrastes entre as sugestões dos dois modelos, apoiando-se para tanto nas métricas de avaliação apontadas no fim do capítulo anterior e nos objetivos estabelecidos previamente.

Adicionalmente, inclui-se uma breve análise comparativa de algumas métricas de similaridade (Cosseno, Euclidiana e *Manhattan*) aplicadas a cada recomendador em ambos os *datasets*. Este exercício, de natureza puramente exploratória, visa elucidar como a escolha da medida influencia a diversidade e precisão das recomendações, consolidando assim as opções metodológicas adotadas.

3.1 *Dataset* "Cursos"

As recomendações devem ser interpretadas da seguinte maneira: considera-se como "item base" o último item consumido/pesquisado/favoritado pelo utilizador. A partir deste item, recomendam-se os mais similares.

Por conseguinte, haja vista que a base de cursos possui apenas oitocentos e sete itens, serão extraídas e apresentadas apenas as três principais recomendações para cada modelo.

a) **Item Base: "Estatística Para Análise De Dados Na Administração Pública"**

- Texto descritivo: estatística de forma simples, objetiva e prática é o que você encontrará neste curso! Faça sua inscrição e aprenda a organizar, analisar dados, determinar suas correlações e a reconhecer os princípios da estatística descritiva em linguagem R para analisar dados na Administração Pública. Matricule-se e faça parte da estatística daqueles que buscam crescimento pessoal e profissional se aperfeiçoando com os cursos da EV.G.

Tfidf Recommender:

Item Recomendado	Similaridade Cosseno
Introdução à Ciência de Dados - Estatística Essencial	0.434109
Análise de Dados em Linguagem R	0.318801
Governança de Dados	0.244432

Tabela 7: Recomendações mais similares ao curso "Estatística Para Análise De Dados Na Administração Pública" - TF-IDF

SBERTrecommender:

Item Recomendado	Similaridade Cosseno
Análise de Dados em Linguagem R	0.767737
Estatística	0.743384
Introdução à Ciência de Dados - Estatística Essencial	0.729554

Tabela 8: Recomendações mais similares ao curso "Estatística Para Análise De Dados Na Administração Pública" - SBERT

- Ressalta-se preliminarmente que, ao analisar o contexto em que o curso base insere-se, todas as recomendações de ambos os sistemas estão ajustados à realidade e são assertivas no que diz respeito à trilha de aprendizado possivelmente pretendida pelo utilizador.
- Ademais, as recomendações de ambos os sistemas foram similares para este caso, o que demonstra que o sistema mais rudimentar baseado em TF-IDF demonstrou utilidade a depender da aplicação. A diferença, contudo, parece estar no grau de confiança. Enquanto o SBERT recomenda "Introdução à Ciência de Dados - Estatística Essencial" com um grau de proximidade elevado (0.729554), o sistema TF-IDF recomenda o mesmo curso com grau de similaridade consideravelmente menor entre os vetores (0.434109).
- Por último, é salutar ressaltar que a recomendação do curso "Governança de Dados" pelo primeiro sistema pode ser um indicativo da sua principal limitação: A não captação de contexto, reduzindo-se a coincidências lexicais entre palavras-chave. Por outro lado, o segundo sistema priorizou cursos mais alinhados ao foco do item-base, sobretudo no que tange à perspectiva quantitativa/estatística.

b) **Item Base: “S2Id - M2 - Usuário Federal - Solicitação De Recursos Para Ações De Resposta”**

- Texto descritivo: neste curso, o aluno vai compreender como utilizar o S2ID para cumprir as funções relacionadas à análise de solicitações de recursos federais para ações de reposta às situações de emergência ou de estado de calamidade pública , composta por procedimentos ligados às atribuições e competências dos perfis de Analista, Coordenador, Coordenador-geral e Diretor.

Tfidf Recommender:

Item Recomendado	Similaridade Cosseno
S2ID - M2 - Usuário Federal - Liberação de Recursos para Ações de Resposta	0.580242
S2ID - M2 - Usuário Estadual - Solicitação de Recursos para Ações de Resposta	0.478507
S2ID - M2 - Usuário Municipal - Solicitação de Recursos para Ações de Resposta	0.458379

Tabela 9: Recomendações mais similares ao curso “S2Id - M2 - Usuário Federal - Solicitação De Recursos Para Ações De Resposta” - TF-IDF

SBERTrecommender:

Item Recomendado	Similaridade Cosseno
S2ID - M2 - Usuário Federal - Liberação de Recursos para Ações de Resposta	0.936987
S2ID - M2 - Usuário Federal - Execução das Ações de Resposta	0.866323
S2ID - M2 - Usuário Estadual - Prestação de Contas das Ações de Resposta	0.849901

Tabela 10: Recomendações mais similares ao curso “S2Id - M2 - Usuário Federal - Solicitação De Recursos Para Ações De Resposta” - SBERT

- Ao analisar as recomendações emitidas para o curso em questão, demonstra-se a capacidade das duas abordagens em captar e recomendar cursos sequenciais à partida. Nota-se que todos os cursos recomendados fazem parte do mesmo ecossistema do item base, sendo portanto sequências lógicas e naturais prontamente identificadas pelos modelos;
- Chama atenção novamente uma sensível diferença no grau de similaridade do cosseno apontado por cada modelo entre o curso base e os recomendados. O SBERT chega a apontar uma relação de quase identidade (0.93) entre as capacitações “S2Id - M2 - Usuário Federal - Solicitação De Recursos Para Ações De Resposta” e “S2ID - M2 - Usuário Federal - Liberação de Recursos para Ações de Resposta”, enquanto o TF-IDF aponta apenas 0.58.

c) **Item Base: “Microeconomia”**

- Texto descritivo: o curso apresenta noções de microeconomia com o objetivo capacitar pessoas para atuar na avaliação socioeconômica de projetos. Essa avaliação inclui a compreensão das implicações sociais, econômicas e ambientais ao longo do ciclo de vida dos projetos e serve como subsídio à tomada de decisões..

Tfidf Recommender:

Item Recomendado	Similaridade Cosseno
Macroeconomia	0.200974
Estatística	0.198600
Introdução à Avaliação Econômica de EE das Energias Renováveis	0.143911

Tabela 11: Recomendações mais similares ao curso “Microeconomia” - TF-IDF

SBERTrecommender:

Item Recomendado	Similaridade Cosseno
Macroeconomia	0.827938
Princípios de Economia: Microeconomia	0.804596
Estatística	0.733347

Tabela 12: Recomendações mais similares ao curso “Microeconomia” - SBERT

- Novamente ambos os modelos performam de maneira satisfatória no que diz respeito à similaridade semântica, temática e contextual entre o item base e as recomendações;
- Há mais uma vez uma quase sobreposição das recomendações entre os dois modelos, o que pode ser explicado pelo tamanho diminuto da base, com apenas oitocentos e sete itens. Ainda assim, nota-se mais uma vez uma ligeira vantagem nas recomendações exaradas pelo modelo SBERT, pois os três itens apontados no caso em tela possuem elevada semelhança com a temática da microeconomia, enquanto o último item recomendado pelo TF-IDF pode não ser tão relevante ao utilizador que realizou a capacitação em Microeconomia.
- Por fim, nota-se outra vez uma importante diferença no valor da similaridade do cosseno apontado em ambos os sistemas. Se para o TF-IDF o curso Macroeconomia possui similaridade de apenas 0.21 ao item base, para o SBERT esta similaridade sobe para 0.83. Apesar de não constituir um problema à primeira vista, vez que ambos os sistemas recomendariam o curso, caso o gestor/operador optasse por definir um *threshold* mínimo de similaridade em 0.5, por exemplo, para evitar recomendações

abaixo deste limiar, o TF-IDF não traria nenhuma recomendação e perderia eficácia. **Estas diferenças serão abordadas mais adiante neste capítulo em secção própria, devido à repetição deste padrão observada ao longo de boa parte das amostras de recomendação extraídas.**

d) **Item Base: "Conceitos Essenciais Sobre Patologias Em Estruturas De Concreto"**

- Texto descritivo: este curso apresenta os principais conceitos aplicados ao desenvolvimento de projetos e estudos sobre patologias em estruturas de concreto armado. Você conhecerá os conceitos essenciais para se trabalhar no processo de identificação das principais manifestações patológicas, especialmente aquelas que ocorrem em Obras de Arte Especiais (OAE). Quer saber mais? Inscreva-se!.

Tfidf Recommender:

Item Recomendado	Similaridade Cosseno
Estruturas Organizacionais do Poder Executivo Federal - Siorg	0.100401
Conhecendo o PROARTE	0.097901
Educação Patrimonial e Arte	0.084762

Tabela 13: Recomendações mais similares ao curso "Conceitos Essenciais Sobre Patologias Em Estruturas De Concreto" - TF-IDF

SBERTrecommender:

Item Recomendado	Similaridade Cosseno
Reconhecimento de Riscos Químicos nos Ambientes de Trabalho	0.676298
Relatório de Segurança de Barragens: o que é e para que serve	0.607058
Conceitos Básicos de Hidrologia e Drenagem para Projetos Rodoviários	0.600677

Tabela 14: Recomendações mais similares ao curso "Conceitos Essenciais Sobre Patologias Em Estruturas De Concreto" - SBERT

- Aqui pela primeira vez o **TF-IDF demonstra de maneira clara aquela que pode ser sua principal limitação: a falta de captação de contexto semântico, e elevada sensibilidade à polissemia.** Isto fica evidente ao comparar-se a primeira recomendação, qual seja, "Estruturas Organizacionais do Poder Executivo Federal - Siorg" com o item base. O sistema TF-IDF não possui a capacidade de distinguir a polissemia existente no termo "estrutura", que diz respeito a uma construção física pertencente ao domínio da Engenharia Civil para o caso do item base e de uma organização hierárquica no caso do curso que está a ser recomendado como o mais similar.

- O modelo SBERT, por sua vez, demonstra-se capaz de perceber o significado contextual do item base ao recomendar capacitações que dizem respeito a riscos e segurança no âmbito da engenharia. Isto enfatiza a evolução que a arquitetura de *transformers* e a captação de contexto representam no campo do PLN, se comparado às abordagens anteriores, conforme o caso do TF-IDF e *CountVectorizer*.

e) **Item Base: "Aprendendo com Python"**

- Texto descritivo: este curso aprofunda os fundamentos da ciência da computação em termos de variáveis, condicionais, loops e funções usando a sintaxe de programação do Python. Aprenda como aplicar esta linguagem para resolver vários problemas e usar seus frameworks / bibliotecas / pacotes para diferentes contextos. Este curso também está disponível na versão inglês.

Tfidf Recommender:

Item Recomendado	Similaridade Cosseno
Design Thinking aplicado a Bibliotecas	0.171908
Learning with Python	0.136896
Linguagem simples aproxima o governo das pessoas. Como usar?	0.108672

Tabela 15: Recomendações mais similares ao curso "Aprendendo com Python" - TF-IDF

SBERTrecommender:

Item Recomendado	Similaridade Cosseno
Learning with Python	0.916468
Introdução à Ciência de Dados - Conceitos e Ferramentas	0.533505
Introdução à Ciência de Dados - Descoberta de Tópicos em Texto	0.512092

Tabela 16: Recomendações mais similares ao curso "Aprendendo com Python" - SBERT

- Ao examinar as similaridades emitidas pelos dois sistemas, percebe-se novamente a importância dada ao contexto pelo sistema SBERT. Enquanto este percebe o *python* como linguagem de programação e instrumento complementar ao domínio da ciência de dados, o outro sistema não foi capaz de reconhecer tais características, recomendando de maneira assertiva tão somente a mesma capacitação em língua inglesa por conta da coincidência lexical do termo *"python"* em seu título e texto de apresentação.
- Por conseguinte, apesar de a análise deste projeto concentrar-se na produção de sistemas de recomendação capazes de atuar sobre dados textuais em língua portuguesa, percebe-se ainda a polivalência do modelo SBERT ao lidar também com textos em língua inglesa. Isto se deve ao facto de o modelo que está a ser utilizado, qual seja o *"paraphrase-multilingual-MiniLM-L12-v2"*

(Reimers & Gurevych, 2019) da biblioteca *sentence transformers/HuggingFace*, ter sido treinado em cinquenta línguas diferentes, dentre elas o inglês. Assim sendo, conforme apontado no texto de descrição do curso base, existe a oferta do mesmo curso porém em língua inglesa na plataforma, e o modelo aponta uma similaridade do cosseno muito elevada entre eles (0.92), enquanto o recomender anterior apontou similaridade de apenas 0.14 entre os itens.

3.2 *Dataset* "Livros"

A exemplo do que foi feito anteriormente as recomendações desta secção continuam a ser intepretadas da mesma maneira: considera-se como "item base" o último item consumido/pesquisado/favoritado pelo utilizador. A partir deste item, recomendam-se os mais similares.

Neste ponto, por tratar-se de uma base com quase trinta mil itens, serão extraídas e apresentadas as seis principais recomendações para cada modelo.

a) **Item Base: "A bíblia do vinho"**

- Autor: Karen MacNeil
- Texto descritivo: o que torna um vinho grande? Qual é a razão das borbulhas do Champagne? Por que os vinhedos de solos muito ruins, e que fazem as videiras sofrerem, podem originar grandes vinhos? Este livro mostra os segredos dos degustadores profissionais e como aumentar seu vocabulário de degustação. A obra abrange todos os aspectos essenciais dos países produtores de vinho, descreve os maiores vinhedos de vinhos finos do mundo e apresenta informações sobre os vinhos de diversas regiões do mundo.

Tfidf Recommender:

Item Recomendado	Similaridade Cosseno
Vinhos - Degustação, Elaboração e Serviço	0.421860
Degustação de vinhos	0.335930
Conheça vinhos	0.304774
Vale dos Vinhedos	0.265985
O Guia Essencial do Vinho	0.263190
Vinhos e comida	0.259839

Tabela 17: Recomendações mais similares ao Livro "A bíblia do vinho" - TF-IDF

SBERTrecommender:

Item Recomendado	Similaridade Cosseno
Vinhos - Degustação, Elaboração e Serviço	0.769939
O Guia Essencial do Vinho	0.766739
Vinhos e comida	0.757899
Os ignorantes	0.747955
Uma breve história da bebedeira	0.728047
A alma do vinho	0.726168

Tabela 18: Recomendações mais similares ao Livro “A bíblia do vinho” - SBERT

- Partindo-se dos referenciais metodológicos propostos neste trabalho, os resultados de ambos os sistemas parecem consistentes, o que pode ser percebido ao comparar-se as nuances semânticas e contextuais entre o item base e todos os itens recomendados;
- Consoante tudo aquilo que fora mencionado nos resultados da aplicação dos sistemas à base de dados anterior (cursos), observa-se mais uma vez a mesma tendência: enquanto o TF-IDF direciona sua atuação à sobreposição lexical, ao indicar itens diretamente relacionados à palavra “vinho” e demais palavras-chave adjacentes constantes aos campos de descrição textual destas obras literárias, o SBERT demonstra capacidade satisfatória de captação de relações contextuais ao indicar não somente obras relativas às técnicas de degustação de vinhos mas também outros temas adjacentes como história e cultura da bebida, como por exemplo em “história da bebedeira” e “Os ignorantes”, este último escrito por Étienne Davodeau, que explora a intersecção entre a produção de vinhos e a criação artística em banda desenhada, destacando a dimensão humana e cultural partilhada entre essas práticas.
- Por fim, observa-se que os valores de similaridade do cosseno contiunam a variar em faixas significativamente mais altas no SBERT (0.73 – 0.77) se comparado ao TF-IDF (0.26 – 0.42).

b) **Item Base: “O homem mais rico da Babilônia”**

- Autor: George Samuel Clason
- Texto descritivo: traz orientações através de parábola ambientada na Babilônia sobre como ter sucesso financeiro, ganhando dinheiro, sabendo poupá-lo e tendo mais lucro.

Tfidf Recommender:

Item Recomendado	Similaridade Cosseno
Ontem não te vi em Babilónia	0.294382
Babilônia	0.282461
Perdidos na Babilônia (Vol. 2 As sete maravilhas)	0.241618
As práticas para a prosperidades extraídas de - O homem mais rico da Babilônia	0.172280
Nós Queremos Que Você Fique Rico	0.131465
A Selva do Dinheiro	0.128673

Tabela 19: Recomendações mais similares ao Livro "O homem mais rico da Babilônia" - TF-IDF

SBERTrecommender:

Item Recomendado	Similaridade Cosseno
As práticas para a prosperidades extraídas de - O homem mais rico da Babilônia	0.695494
Desperte o milionário que há em você	0.690192
Os segredos da mente milionária	0.649632
Inabalável	0.639082
Dinheiro - os segredos de quem tem	0.636238
Pai Rico, Pai Pobre	0.632371

Tabela 20: Recomendações mais similares ao Livro "O homem mais rico da Babilônia" - SBERT

- Apesar do título remeter à temática de romance ou história, percebe-se pelo texto descritivo que o livro em questão diz respeito a uma parábola centrada em educação financeira, mentalidade de riqueza e prosperidade. Desta feita, percebe-se que os diferentes contextos de aplicação do termo "Babilônia" fizeram com que o modelo baseado em TF-IDF recomendasse priortariamente itens que não possuem similaridade com a real temática do item base, como as obras de ficção/romance "Ontem não te vi em Babilónia", do autor português António Lobo Antunes e "Perdidos na Babilônia", de autoria de Peter Lerangis, sendo este parte de uma aclamada série de livros de ficção/aventura.
- Em constraste, o modelo baseado em *embeddings* densos demonstrou boa precisão ao recomendar obras alinhadas ao contexto do item base. Soma-se a isso o facto de que mesmo o item de referência possuindo um texto descritivo curto e objetivo, sobretudo se comparado às descrições dos demais itens apresentados neste trabalho até então, foi o suficiente para que o modelo percebesse o contexto e realizasse recomendações como "As práticas para a prosperidades extraídas de - O homem mais

rico da Babilônia", "Desperte o milionário que há em você" e "Os segredos da mente milionária". A elevada similaridade (valores entre 0,63 e 0,69) reflete a capacidade do SBERT de capturar nuances semânticas mesmo com descrições textuais curtas, interpretando o item base como símbolo de princípios financeiros atemporais, não como referência geográfica ou literária.

c) **Item Base: "O diário de Anne Frank"**

- Autor: Anne Frank
- Texto descritivo: 12 de junho de 1942 – 1º de agosto de 1944. Ao longo deste período, a jovem Anne Frank escreveu em seu diário toda a tensão que a família Frank sofreu durante a Segunda Guerra Mundial. Ao fim de longos dias de silêncio e medo aterrorizante, eles foram descobertos pelos nazistas e deportados para campos de concentração. Anne inicialmente foi para Auschwitz, e mais tarde para Bergen-Belsen. A força da narrativa de Anne, com impressionantes relatos das atrocidades e horrores cometidos contra os judeus, faz deste livro um precioso documento. Seu diário já foi traduzido para 67 línguas, e é um dos livros mais lidos do mundo. Ele destaca sentimentos, aflições e pequenas alegrias de uma vida incomum, problemas da transformação da menina em mulher, o despertar do amor, a fé inabalável na religião e, principalmente, revela a rara nobreza de um espírito amadurecido no sofrimento. Um retrato da menina por trás do mito.

Tfidf Recommender:

Item Recomendado	Similaridade Cosseno
Anne Frank: Obra Reunida	0.489303
O mundo de Anne Frank	0.446751
Os sete últimos meses de Anne Frank	0.418117
A HISTORIA DA FAMILIA DE ANNE FRANK	0.394489
Querida Kitty: edição bolso de luxo	0.313102
O menino que amava Anne Frank	0.304058

Tabela 21: Recomendações mais similares ao Livro "O diário de Anne Frank" - TF-IDF

SBERTrecommender:

Item Recomendado	Similaridade Cosseno
O mundo de Anne Frank	0.774378
Querida Kitty: edição bolso de luxo	0.760387
Diário de Hélène Berr, O	0.749785
É hora de falar	0.732856
Apátrida	0.721721
Anne Frank: Obra Reunida	0.717516

Tabela 22: Recomendações mais similares ao Livro "O diário de Anne Frank" - SBERT

- Ao analisar as recomendações emitidas pelo modelo TF-IDF para o livro "O diário de Anne Frank", não há nenhuma incoerência: o modelo mostra-se capaz de realizar recomendações precisas por apoiar-se na sobreposição de palavras-chave, sobretudo no próprio nome de Anne Frank que aparece diversas vezes ao longo dos textos descritivos das obras recomendadas.
- O modelo baseado em SBERT, por sua vez, além de privilegiar o alinhamento temático, demonstra novamente a característica de domínio contextual ao recomendar obras como "Diário de Hélène Berr" e "É hora de falar", que expandem o horizonte temático para além da figura central de Anne Frank. Essas obras, embora não mencionem explicitamente o nome "Anne Frank" em seus títulos, partilham o cerne temático de relatos autobiográficos do Holocausto, o que demonstra a capacidade do SBERT de mapear conexões semânticas mais profundas.
- Por fim, enquanto o TF-IDF apresenta similaridades moderadas (0.30–0.48), refletindo a correspondência linear baseada em repetição lexical, o SBERT alcança valores significativamente mais elevados (0.71–0.77). Essa diferença quantitativa traduz-se qualitativamente: o SBERT não apenas identifica a palavra-chave "Anne Frank", mas reconhece o contexto histórico e literário do livro base, capturando subtemas adjacentes.

d) **Item Base: "Harry Potter e a Pedra Filosofal"**

- Autor: J.K. Rowlling
- Texto descritivo: 'Harry Potter e a Pedra Filosofal' conta a história de um menino que dorme embaixo de uma escada na casa dos tios. Quando ainda bebê, Harry teve sua casa invadida por um terrível bruxo responsável pelo assassinato de seus pais e é o único sobrevivente. Porém, Harry não sabe disso, e acha que é apenas um garoto normal que às vezes parece fazer coisas estranhas acontecerem. Entretanto, no dia de seu aniversário de 11 anos, Harry recebe uma visita inesperada e descobre que é um bruxo, assim como seus pais foram, e que está convidado a ingressar na Escola de Magia e Bruxaria de Hogwarts.

Harry, então, vai para Hogwarts, onde irá aprender poções, feitiços e a jogar Quadribol, e se meter em aventurar que irão ensiná-lo sobre a vida.

Tfidf Recommender:

Item Recomendado	Similaridade Cosseno
Harry Potter e a Câmara Secreta	0.456347
Harry Potter - Guia cinematográfico	0.443397
Harry e seus fãs	0.414196
Harry Potter e a Ordem da Fênix	0.366933
Harry Potter e o cálice do fogo	0.365371
HARRY POTTER E A ORDEM DA FENIX	0.349288

Tabela 23: Recomendações mais similares ao Livro "Harry Potter e a Pedra Filosofal" - TF-IDF

SBERTrecommender:

Item Recomendado	Similaridade Cosseno
HARRY POTTER E O ENIGMA DO PRINCIPE	0.763356
Harry Potter - Guia cinematográfico	0.759967
HARRY POTTER E A ORDEM DA FENIX	0.742773
O livro dos personagens de Harry Potter	0.721316
HARRY POTTER E O PRISIONEIRO DE AZKABAN	0.718321
O universo de Harry Potter de A a Z	0.716731

Tabela 24: Recomendações mais similares ao Livro "Harry Potter e a Pedra Filosofal" - SBERT

- Novamente, o TF-IDF demonstra boa capacidade na recomendação de sequências lógicas. Ao deparar-se com o primeiro livro da série Harry Potter, o modelo sugere como próxima leitura o segundo volume, "Harry Potter e a Câmara Secreta", alinhando-se à expectativa de leitores que buscam continuidade narrativa. Essa abordagem, embora linear, é eficaz para se atingir aqueles utilizadores que priorizam a ordem cronológica das obras.
- Por outro lado, o SBERT ignora a sequência direta dos livros neste caso, e expande sua atuação para recomendações que exploram o universo expandido da série, como "O livro dos personagens de Harry Potter" e "O universo de Harry Potter de A a Z". Ambos os sistemas, portanto, apresentam recomendações aderentes ao contexto e à temática da obra de referência, demonstrando boa capacidade de associação entre itens com base em suas similaridades, ainda que com enfoques distintos: o TF-IDF

na linearidade narrativa e o SBERT na profundidade contextual.

e) **Item Base: “A revolução dos bichos”**

- Autor: George Orwell
- Texto descritivo: ‘A revolução dos bichos’ é uma fábula sobre o poder. Narra a insurreição dos animais de uma granja contra seus donos. Progressivamente, porém, a revolução degenera numa tirania ainda mais opressiva que a dos humanos. Escrita em plena Segunda Guerra Mundial e publicada em 1945 depois de ter sido rejeitada por várias editoras, essa pequena narrativa causou desconforto ao satirizar ferozmente a ditadura stalinista numa época em que os soviéticos ainda eram aliados do Ocidente na luta contra o eixo nazifascista.

Tfidf Recommender:

Item Recomendado	Similaridade Cosseno
História concisa da Revolução Russa (edição de bolso)	0.166325
A Fazenda dos Animais - Edição especial	0.162831
A revolução vietnamita	0.159904
O antigo regime e a Revolução	0.151463
A Quarta Revolução Industrial	0.145681
Almanaque Maluquinho - Bocão e os bichos	0.140626

Tabela 25: Recomendações mais similares ao Livro “A revolução dos bichos” - TF-IDF

SBERTrecommender:

Item Recomendado	Similaridade Cosseno
A Batalha Dos Livros	0.717866
A maldição de Stalin	0.705019
Tornando-se Hitler	0.704560
De volta do inferno	0.698116
Morangos mofados	0.696090
A Segunda Guerra Mundial	0.689571

Tabela 26: Recomendações mais similares ao Livro “A revolução dos bichos” - SBERT

- Trata-se neste caso de um cenário intencional de ambiguidade lexical, trazido à discussão para testar a capacidade dos modelos em lidar com polissemia e contexto implícito. A obra “A Revolução dos Bichos”, apesar do título, é uma sátira política que utiliza animais como alegoria para criticar o regime

stalinista na União Soviética. O desafio aqui reside em distinguir a camada superficial (termos como "bichos" e "revolução") da camada semântica profunda (crítica ao totalitarismo).

- Com baixos valores de similaridade apontados, o sistema TF-IDF focou mais uma vez na coincidência entre palavras-chave ao recomendar obras que contém as palavras "revolução" ou "bichos" em seus textos descritivos. Isto gerou tanto recomendações adequadas como "História Concisa da Revolução Russa" e "A Fazenda dos Animais - Edição Especial" (outra sátira de George Orwell) como recomendações irrelevantes para a temática pretendida, como "A quarta revolução industrial" e o livro infantil "Almanaque Maluquinho - Bocão e os Bichos", expondo portanto a limitação do sistema em ir além da literalidade dos termos.
- O mesmo não parece ocorrer ao modelo SBERT. Além de não debruçar-se sobre nenhuma das palavras-chave supramencionadas, o modelo demonstrou capacidade de generalização para domínios relacionados ao indicar ao utilizador, por exemplo, "A maldição de Stalin" (similaridade 0.705) e "Tornando-se Hitler" (similaridade 0.704), livros que exploram a história de figuras ditatoriais no contexto da Segunda Guerra Mundial, demonstrando ainda por meio das demais recomendações a perceção da intenção satírica e política da obra de referência.

f) **Item Base: "Homo Deus"**

- Autor: Yuval Noah Harari
- Texto descritivo: o futuro visto por um dos grandes pensadores da atualidade: isto é Homo Deus. Neste Homo Deus: uma breve história do amanhã, Yuval Noah Harari, autor do estrondoso best-seller Sapiens: uma breve história da humanidade, volta a combinar ciência, história e filosofia, desta vez para entender quem somos e descobrir para onde vamos. Sempre com um olhar no passado e nas nossas origens, Harari investiga o futuro da humanidade em busca de uma resposta tão difícil quanto essencial: depois de séculos de guerras, fome e pobreza, qual será nosso destino na Terra? A partir de uma visão absolutamente original de nossa história, ele combina pesquisas de ponta e os mais recentes avanços científicos à sua conhecida capacidade de observar o passado de uma maneira inteiramente nova. Assim, descobrir os próximos passos da evolução humana será também redescobrir quem fomos e quais caminhos tomamos para chegar até aqui.

Tfidf Recommender:

Item Recomendado	Similaridade Cosseno
21 lições para o século 21	0.288593
Sapiens (Edição em quadrinhos): O nascimento da humanidade	0.207909
Homo Ferox	0.205478
Na batalha contra o coronavírus, faltam líderes à humanidade	0.199605
Notas sobre a pandemia	0.195967
Homo biologicus	0.177440

Tabela 27: Recomendações mais similares ao Livro “Homo Deus” - TF-IDF

SBERTrecommender:

Item Recomendado	Similaridade Cosseno
21 lições para o século 21	0.734084
A curva do sonho	0.690204
Homo biologicus	0.688249
Para onde foi o futuro?	0.686418
Evolução	0.686074
História do futuro	0.677030

Tabela 28: Recomendações mais similares ao Livro “Homo Deus” - SBERT

- Ainda que se prenda mais uma vez às palavras-chave, o modelo TF-IDF apresentou em suas três primeiras recomendações livros relevantes e ajustados à temática do item base, ressaltando-se que o livro “21 lições para o século 21” fora escrito pelo mesmo autor de “Homo Deus” e aborda questões filosóficas semelhantes.
- Entretanto, o TF-IDF demonstrou limitações ao recomendar livros como “Na batalha contra o coronavírus, faltam líderes à humanidade” e “Notas sobre a pandemia”, que aparentemente possuem pouca correlação com a obra de referência.
- Ratificando a tendência apresentada até agora, houve boa performance do SBERT. Destaca-se o peso contextual da temática de futuro da humanidade que foi captada pelo sistema de recomendação e considerado no ato de emissão de recomendações como “Para Onde Foi o Futuro?”, “Evolução” e “História do Futuro”.

- É oportuno mencionar ainda as elevadas similaridades encontrado pelo SBERT, a variar no intervalo entre 0.68 e 0.73, diferente do TF-IDF que encontrou correlações com valores abaixo de 0.3.

3.3 Uso de métricas de dissimilaridade: a aplicação da distância de *Manhattan* e distância Euclidiana aos sistemas de recomendação em tela

No âmbito deste estudo, procederam-se ensaios adicionais com duas medidas de dissimilaridades - distância Euclidiana $(L2)$ e distância de *Manhattan* $(L1)$ - aplicadas aos vetores representativos dos textos descritivos dos itens nos dois sistemas de recomendação. O objetivo consistiu em verificar se estas métricas, convertidas em similaridades, produziam recomendações coerentes quando comparadas com a similaridade do cosseno.

3.3.1 Aplicação da distância Euclidiana

Ambos os sistemas (TF-IDF com *TfidfVectorizer* do *scikit-learn* e *embeddings* densos do modelo *paraphrase-multilingual-MiniLM-L12-v2*) utilizam, por defeito ou por pré-processamento, vetores normalizados à $norma - L2$, conforme documentações oficiais presentes em scikit-learn.org e sbert.net. Nesse contexto, a distância Euclidiana entre dois vetores normalizados satisfaz

$$\|\mathbf{u} - \mathbf{v}\|_2^2 \;=\; 2\,(1 - \cos(\mathbf{u},\, \mathbf{v}))$$

o que implica equivalência monotónica com a similaridade do cosseno. Na prática, as recomendações obtidas com ambas as métricas coincidiram ao longo dos experimentos, confirmando-se que, num espaço de vetores unitários, minimizar a distância Euclidiana mostrou-se equivalente à maximização do cosseno. **Todavia, a explicabilidade revelou-se mais intuitiva no caso da similaridade do cosseno, pois se associa diretamente a valores entre 0 e 1 (quanto maior, maior similaridade), ao passo que a interpretação de "distâncias mais pequenas indicam maior semelhança" tende a causar alguma confusão ou ser contraintuitiva para pessoas que possuem pouca ou nenhuma familiaridade com este tipo de conceitos.**

3.3.2 Aplicação da distância de *Manhattan*

A distância de *Manhattan* (Riesz, 1910) é uma métrica que mede a dissimilaridade entre dois vetores somando as diferenças absolutas de cada dimensão. A fórmula para cálculo desta medida entre dois vetores u e v pode ser representada como:

$$\|\mathbf{u} - \mathbf{v}\|_1 \; = \; \sum_{i=1}^{n} |u_i - v_i|$$

Onde,

- $\|\mathbf{u} - \mathbf{v}\|_1$ é a Distância de *Manhattan* entre os vetores $\mathbf{u}$ e $\mathbf{v}$,
- u_i e v_i correspondem às i-ésimas componentes dos vetores $\mathbf{u}$ e $\mathbf{v}$, respectivamente,
- $|u_i - v_i|$ denota o valor absoluto da diferença entre u_i e v_i na dimensão i,
- $\sum_{i=1}^{n}$ indica a soma das diferenças absolutas em todas as n dimensões do espaço vetorial.

No que concerne à aplicação da distância de *Manhattan* $(L1)$ ao caso em tela, o emprego desta métrica sobre vetores TF-IDF normalizados encontrou-se com elevada esparsidade: a distribuição assimétrica dos poucos valores não nulos culminou na geração de distâncias $L1$ desproporcionadas, o que detriorou a eficiência da medida e resultou em recomendações de qualidade baixa e/ou insuficiente, conforme análise comparativa entre a Tabela 29 e a Tabela 30.

Item Recomendado	Similaridade Cosseno
Introdução à Ciência de Dados - Descoberta de Tópicos em Texto	0.226282
Português - Interpretação de Texto e Emprego de Regras Gramaticais	0.224332
Visão Geral da IA no Azure	0.170644

Tabela 29: Recomendações mais similares ao curso “Processamento de Linguagem Natural” - TF-IDF (Similaridade do Cosseno)

Item Recomendado	Distância de Manhattan
Fundamentos da Integridade Pública: Prevenindo a Corrupção	8.305175
Análise de Condutas Unilaterais Restritivas à Concorrência	8.412200
Gestão da Mobilidade	8.412875

Tabela 30: Recomendações mais similares ao curso “Processamento de Linguagem Natural” - TF-IDF (Distância de Manhattan)

Entretanto, **o mesmo fenómeno não repetiu-se nos *embeddings* densos gerados pelo SBERT**, conforme análise das tabelas a seguir:

Item Recomendado	Similaridade Cosseno
Primeiros passos para uso de Linguagem Simples	0.692621
Visão Geral da IA no Azure	0.681358
Linguagem simples aproxima o governo das pessoas. Como usar?	0.680942

Tabela 31: Recomendações mais similares ao curso “Processamento de Linguagem Natural” - S-BERT (Similaridade do Cosseno)

Item Recomendado	Distância de Manhattan
Linguagem simples aproxima o governo das pessoas. Como usar?	12.252589
Primeiros passos para uso de Linguagem Simples	12.311651
Visão Geral da IA no Azure	12.508574

Tabela 32: Recomendações mais similares ao curso “Processamento de Linguagem Natural” - S-BERT (Distância de Manhattan)

Nota-se que tanto a similaridade do cosseno quanto a distância de *Manhattan* produziram os mesmos resultados em ordem ligeiramente diferente, o que continuou a ocorrer ao longo dos demais testes realizados. A principal justificativa para esta ocorrência diz respeito à alta densidade (ausência total de valores nulos) e baixa dimensionalidade (se comparado ao método anterior) dos *embeddings* gerados pelo modelo, sobretudo se comparado ao TF-IDF, suavizou a contribuição individual de cada componente e harmonizou a relação de $L1$ com a similaridade do cosseno.

Desta feita, os ensaios realizados no âmbito deste projeto demonstraram que a métrica de *Manhattan* pode ser viável em contextos específicos (ex.: vetores densos e normalizados), mas ainda carece de vantagens práticas sobre a similaridade do cosseno, vez que assim como a distância euclidiana também possui interpretação contraintuitiva.

Ante todo o exposto, as experiências demonstraram que, quando os vetores estão normalizados, a distância Euclidiana replica a ordem do Cosseno sem perda de informação, mas demonstra perdas no que diz respeito à clareza interpretativa. A distância de *Manhattan*, por sua vez, falha em captar de forma consistente a semelhança entre os vetores em espaços esparsos (TF-IDF) e performa de modo similar ao cosseno em espaços densos (SBERT), pelo que a similaridade do cosseno se mantém como a métrica de referência para sistemas de recomendação textual, de modo a ratificar a sua escolha para o desenvolvimento deste estudo.

3.4 Considerações finais sobre os resultados

Ante todo o exposto neste capítulo, resta claro que ambos os sistemas desenvolvidos demonstraram capacidade de emissão de recomendações assertivas e podem auxiliar organizações no atingimento dos seus objetivos, sem a necessidade de infraestrutura complexa e onerosa para recolha e processamento de grandes quantidades de dados.

Entretanto, há uma boa vantagem na adoção do sistema baseado em vetores de *embeddings* densos (*SBERT based*). Se de um lado o sistema TF-IDF demonstrou elevada capacidade para lidar com coincidências literais entre termos, o que o leva a ser uma boa opção para a recomendação de itens sequenciais ou com títulos muito semelhantes, por outro lado demonstrou dificuldades em perceber o contexto que permeia os itens que lhe foram apresentados, falhando portanto em emitir recomendações mais diversas e baseadas na temática ao deparar-se com domínios mais restritos.

O SBERT por sua vez demonstrou interessante capacidade de captar relações contextuais entre as descrições dos itens, ainda que os textos dos itens partilhassem poucas palavras-chave entre si. Neste sistema, termos como “riqueza”, “investimento” e “sucesso”, por exemplo, são mapeados para regiões próximas no espaço vetorial denso de trezentos e oitenta e quatro dimensões criado pelo modelo *paraphrase-multilingual-MiniLM-L12-v2* (Reimers & Gurevych, 2019), de modo que sejam associadas quando da emissão das recomendações.

Ademais, há mais uma vantagem em adotar-se o SBERT: os *embeddings* densos fazem com que os valores de similaridade do cosseno variem em intervalos superiores àqueles observados no TF-IDF, o que facilita sobremaneira o controlo de qualidade das recomendações após a implantação do modelo, vez que o responsável por esta etapa pode definir um limiar mínimo de similaridade mais elevado para que determinada recomendação seja entregue ao utilizador final.

Ponto reiteradamente observado ao longo dos experimentos, **a discrepância observada nos valores de similaridade do cosseno apresentados por ambos os sistemas para os mesmos itens pode ser explicada pela diferença entre as estruturas dos vetores de representação textual gerados.** Conforme demonstrado na Tabela 3, cada vetor criado pelo TF-IDF possui elevado número de dimensões (sendo uma dimensão correspondente a cada termo único identificado no vocabulário), as quais possuem valor zero em sua maioria, o que culmina na criação de uma matriz esparsa ao “empilhar-se” cada um desses vetores. Assim sendo, se por exemplo dois itens partilham apenas dez palavras dentre as cem mil existentes naquele vocabulário, há uma tendência de baixa similaridade identificada, pois os vetores neste caso serão quase ortogonais. Já no caso dos vetores de *embeddings* gerados pelo modelo baseado em *deep learning*, o número de dimensões é fixo (trezentos e oitenta e quatro para o caso do modelo em uso *paraphrase-multilingual-MiniLM-L12-v2*) e todas elas possuem valores não nulos, o que permite a maximização da similaridade entre textos

semanticamente relacionados e possibilita que múltiplas características semânticas contribuam para a identificação de similaridades. Isto está intimamente ligado ao processo de treinamento do modelo SBERT, onde os vetores são ajustados para que pares relevantes apresentem similaridade do cosseno elevada, mesmo em frases com termos diferentes.

O principal obstáculo ao sistema de recomendação SBERT reside na limitação do seu contexto aos dados utilizados na etapa de treinamento do modelo, entretanto não se mostrou como um empecilho para aplicação aos itens elencados neste trabalho. Uma forma de superar possíveis limitações nesse sentido seria adicionar os dados do domínio pretendido em uma etapa de ajuste do modelo, ressaltando-se que isto pode ser oneroso em termos computacionais e de infraestrutura.

4 *USE CASE:* aplicação do recomendador SBERT aos dados da política nacional de desenvolvimento de pessoas

Instituída no âmbito do ordenamento jurídico brasileiro pelo Decreto nº 9.991, de 28 de agosto de 2019, a política nacional de desenvolvimento de pessoas (PNDP) tem por objetivo promover e fomentar o desenvolvimento dos servidores do poder público brasileiro no que diz respeito às competências necessárias para o alcance da excelência na atuação dos Órgãos e Entidades da Administração Pública Federal, Direta, Autárquica e Fundacional.

Conforme a página oficial do governo brasileiro, o objetivo da PNDP é estabelecer a cultura de planeamento de ações de desenvolvimento entre as instituições federais, com base no alinhamento das necessidades de cada órgão e entidade, tendo como meta o aprimoramento da gestão pública, considerando as boas práticas do mercado de trabalho.

4.1 Materialização da política

Entre os instrumentos para a efetivação da PNDP está o plano de desenvolvimento de pessoas (PDP) que permite planear e orientar a execução da política. O plano é elaborado anualmente por todos os órgãos e entidades da administração pública federal, **a partir do levantamento das necessidades de desenvolvimento dos servidores para o alcance dos resultados organizacionais,** sendo a necessidade uma lacuna identificada entre o desempenho esperado e o desempenho atual.

Todo o processo de levantamento de necessidades e elaboração do PDP é realizado pelo órgão/entidade no portal do sistema de pessoal civil da administração federal (Sipec), consolidado e encaminhado via sistema, para análise do órgão central do Sipec.

Devem constar no PDP a descrição das necessidades de desenvolvimento que serão contempladas no exercício financeiro, com a respectiva carga horária estimada; o público-alvo de cada ação de desenvolvimento; e os custos estimados das ações de desenvolvimento.

O PDP tem que alinhar as necessidades de desenvolvimento com a estratégia do órgão/entidade e estabelecer objetivos e metas institucionais como referência para o planeamento das ações de desenvolvimento. Além disso, o PDP deve atender às necessidades operacionais, táticas e estratégicas,

vigentes e futuras, além de nortear o planeamento das ações de desenvolvimento de acordo com os princípios da economicidade e da eficiência.

4.2 Participação da Fundação Escola Nacional de Administração Pública (Enap)

Após o envio dos PDPs pelos órgãos e entidades, **os planos são consolidados e encaminhados para a Fundação Escola Nacional de Administração Pública (Enap), conforme exemplo da Figura 13. A Enap, por sua vez, apresenta recomendações de soluções de desenvolvimento/capacitação para as necessidades identificadas. Essas soluções são encaminhadas às instituições,** via Portal Sipec, junto com a Manifestação Técnica emitida pelo órgão central do Sipec. De posse da Manifestação, órgãos e entidades poderão iniciar a execução dos seus planos, a partir dos planeamentos realizados e das ações indicadas na Manifestação Técnica.

A primeira rodada de recomendações emitidas pela ENAP remonta ao ano de 2020. Após o recebimento da consolidação de cerca de 20.000 demandas de capacitação, a escola ficou responsável por avaliá-las e sugerir cursos de seu catálogo ou de outras escolas de governo correspondentes àquelas necessidades, o que fora feito inicialmente de maneira manual.

identificador	Órgão	necessidade_ifs1	tema_ifs3	recorte_tema
415327	INSTITUTO NAC.	Adquirir conhecimentos sobre a aplicação do Power BI para aprimoramento das atividades de auditoria.	Metodologia e Técnicas da Computação	Power BI
422275	INSTITUTO NAC.	Domínio do conhecimento em novas tecnologias	Medidas Elétricas, Magnéticas e Eletrônicas; Instrumentação	Instrumentação e Medida de grandezas físicas
424642	INSTITUTO NAC.	Inteligência de negócios com uso de linguagem de programação SQL, Phyton, HTML, etc.	Sistemas de Computação	linguagem de programação SQL, Phyton, HTML
421827	INSTITUTO NAC.	Entendimento do funcionamento do Docker; a gerenciar redes e plugins no Docker; a criar seu próprio registry; à criação de clusters e serviços; ao ger	Sistemas de Computação	Docker

Figura 13: Exemplo de entradas da planilha consolidada de necessidades da PNDP que chega à ENAP para emissão de recomendações

Dado o elevado volume e a diversidade das demandas, esse trabalho exigia esforço considerável das equipes envolvidas, sendo naturalmente mais suscetível a variações e inconsistências decorrentes da complexidade do processo.

Por conseguinte, no ano de 2021 a equipe da Coordenação-Geral de Ciência de Dados da ENAP, da qual faço parte, iniciou o desenvolvimento de soluções com vistas a automatizar o processo de emissão destas

recomendações com o objetivo de reduzir o esforço humano empreendido na tarefa. O primeiro algoritmo utilizado baseava-se em TF-IDF, com funcionamento similar ao proposto na Secção TF-IDFRecommender deste projeto.

Durante os próximos anos, foram aplicadas melhorias no pré-processamento dos dados imputados ao sistema como a pré-categorização das necessidades por áreas temáticas e a identificação de palavras-chave, além de testes com outras técnicas e métricas de similaridade entre frases como *latent dirichlet allocation* (LDA), *Jaccard* e *Word2Vec*, tudo isto com o objetivo de fazer a ponte entre os textos descritivos dos cursos existentes na base de dados e os textos descritivos das necessidades dos servidores públicos.

Todas estas melhorias foram importantes para o aperfeiçoamento das recomendações exaradas, mas ainda havia um obstáculo: a qualidade das recomendações geradas dependia sobremaneira de correspondências lexicais exatas entre palavras-chave; Ademais, as matrizes esparsas geradas pelas técnicas de vetorização textual empregadas demonstravam-se como um problema para a assertividade das recomendações.

4.3 Aplicação do algoritmo de recomendação *SBERT-Based* proposto neste projeto ao caso concreto da PNDP

Desta feita, fui responsável pela proposição de uma nova solução no ano de 2024, a qual apoiou-se no algoritmo SBERT demonstrado neste projeto, com fulcro em aprimorar a qualidade das correspondências entre demandas e ofertas de capacitação por meio do uso de arquitetura de *transformers* para a geração de *embeddings* densos que possibilitassem a captação de contexto e o controlo de qualidade das recomendações por meio do estabelecimento de *thresholds* mínimos de similaridade do cosseno entre os vetores.

Nesta etapa aplicou-se o modelo *sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2* (Reimers & Gurevych, 2019) aos dados tratados e transformados, reduzindo o espaço vetorial gerado à ordem de trezentos e oitenta e quatro dimensões e alcançando uma elevada precisão nas recomendações, haja vista que a partir deste momento o contexto semântico passa a ser percebido e considerado pelo sistema quando da emissão das recomendações.

A solução fora apresentada aos demandantes por meio do produto mínimo viável (PMV) consignado na Figura 14, onde o gestor do sistema realiza apenas o *upload* da planilha de necessidades de capacitação e determina o valor mínimo de similaridade do cosseno que deve ser utilizado pelo modelo como linha de base para emissão ou não de recomendações. Este sistema demonstrou-se capaz de emitir recomendações para uma planilha com mais de dez mil necessidades em pouco menos de quatro minutos.

Figura 14: Exemplo de produto mínimo viável utilizado na aplicação do sistema SBERT à recomendação de cursos para necessidades elencadas na PNDP

Conforme demonstrado na Figura 15, o sistema de recomendação analisa o texto de cada necessidade de capacitação, compara-o com os dados textuais da matriz de cursos disponíveis e recomenda os três cursos mais similares à necessidade, de acordo com o limite mínimo de similaridade do cosseno indicado pelo operador, que no exemplo em tela fora definido em 0,55. Nota-se ainda que caso não sejam encontrados cursos condizentes com a similaridade mínima definida, a *engine* não realiza nenhuma recomendação, uma vez que poderia incorrer em falhas ao recomendar cursos com pouca ou nenhuma relevância para a necessidade elencada.

Identificador	Necessidade Treinamento	sim_coss	Curso(s) Recomendado(s)	Escola(s)	Link(s)
397578	Ferramentas do Microsoft Office 365 Metodologia e Técnicas da Computação Microsoft Office	[0.7441552877426147, 0.6662893295288086, 0.6617662906646729]	primeiros passos com o microsoft 365; otimização de trabalho com o microsoft 365; colaboração com o microsoft 365	EVG(ENAP); EVG(ENAP); EVG(ENAP)	https://www.escolavirtual.gov.br/curso/1138; https://www.escolavirtual.gov.br/curso/1156; https://www.escolavirtual.gov.br/curso/1139
398097	Técnicas de programação em HTML 5 com CSS 3 Metodologia e Técnicas da Computação Programação em HTML 5	[]	Não existem cursos que correspondam à essa necessidade.	[]	[]
380433	Sensoriamento Remoto Florestal Metodologia e Técnicas da Computação Sensoriamento Remoto	[]	Não existem cursos que correspondam à essa necessidade.	[]	[]
372773	Compreender os fundamentos, aplicações de segurança da informação e segurança cibernética, no contexto técnico, estratégico e de políticas públicas. Metodologia e Técnicas da Computação Segurança da Informacão e cibernética	[0.8151251673698425, 0.8140789270401001, 0.7589436769485474]	primeiros passos em cibersegurança no serviço público; fundamentos da segurança cibernética - introdução ao cis controls; segurança cibernética: controles 1 a 6 do cis controls	SUAP(ENAP); EVG(ENAP); EVG(ENAP)	https://suap.enap.gov.br/vitrine/curso/2395/; https://www.escolavirtual.gov.br/curso/1153; https://www.escolavirtual.gov.br/curso/1073
388133	Capacitação de técnicas e métodos de tecnologias emergentes como Inteligência Artificial, Computação Quântica e Robótica aplicada a Ciência Espacial Metodologia e Técnicas da Computação Metodologia e Técnicas da Computação	[0.5763888359069824, 0.5749669075012207, 0.567435026168823 2]	inteligência artificial para simplificar o dia a dia; automação de processos através da rpa para transformação digital; inteligência artificial no contexto do serviço público	EVG(ENAP); EVG(ENAP); EVG(ENAP)	https://www.escolavirtual.gov.br/curso/861; https://www.escolavirtual.gov.br/curso/797; https://www.escolavirtual.gov.br/curso/377
392486	Desenvolver maior habilidade com o Excel Metodologia e Técnicas da Computação Ferramentas e funcionalidade do Excel	[0.7635459899902344, 0.6023586988449097, 0.5733567476272583]	introdução ao excel; praticando business intelligence; gerenciar dados com o microsoft 365	EVG(ENAP); SUAP(ENAP); EVG(ENAP)	https://www.escolavirtual.gov.br/curso/459; https://suap.enap.gov.br/portaldoaluno/curso/2016/?area=2; https://www.escolavirtual.gov.br/curso/1160
398132	Técnicas de programação em softwares estatísticos Metodologia e Técnicas da Computação Softwares estatísticos	[0.6687991619110107, 0.6284525394439697, 0.5969482660293579]	análise estatística descritiva com uso de r; estatística para análise de dados na administração pública; análise estatística inferencial com uso de r	SUAP(ENAP); EVG(ENAP); SUAP(ENAP)	https://suap.enap.gov.br/portaldoaluno/curso/2046/?area=2; https://www.escolavirtual.gov.br/curso/930; https://suap.enap.gov.br/portaldoaluno/curso/2047/?area=2

Figura 15: Exemplo de output emitido pelo algoritmo SBERT para a planilha de necessidades da PNDP

Se antes era necessário que uma equipe inteira fosse mobilizada durante dezenas de dias para analisar cada solicitação de capacitação, compará-la visualmente à matriz de cursos disponíveis e montar a base de recomendações em um novo documento, após o advento desta solução é necessário apenas que uma pessoa revise a planilha de recomendações emitida pelo sistema baseado em SBERT, o que demonstra a utilidade e versatilidade do algoritmo desenvolvido, bem como a sua capacidade de gerar economia e eficiência no âmbito organizacional.

5 Conclusão e investigação futura

Neste capítulo serão elencadas as conclusões obtidas após o término do projeto, bem como os pontos que podem ser trabalhados em investigação futura neste campo de estudo.

5.1 Conclusões

A primeira conclusão que pode ser obtida com base nas análises e resultados deste trabalho diz respeito à importância dos dados não estruturados. Em um contexto onde este tipo de informação é cada vez mais abundante, boa parte das organizações ainda focam suas atenções no tratamento e análise apenas de dados estruturados para extração de insights e geração de inteligência negocial.

Desta feita, o trabalho ora em comento demonstrou por meio de resultados concretos o potencial existente na utilização dos dados não estruturados e o valor que pode ser obtido ao analisar-se da maneira correta este tipo de informação.

No que diz respeito ao alcance dos objetivos definidos ao longo da introdução deste trabalho, ressalta-se em primeira medida a demonstração da evolução no campo do processamento de linguagem natural ao longo do tempo, sobretudo nos últimos vinte anos, fazendo com que este sub-domínio da Inteligência Artificial receba cada vez mais atenção no âmbito empresarial e da ciência de dados enquanto área do conhecimento. Para tanto, o projeto buscou ao máximo empregar e demonstrar desde técnicas mais rudimentares a técnicas contemporâneas e tidas como avançadas na seara do PLN e da Inteligência Artificial.

Ademais, o presente projeto demonstrou que ao aplicar-se as ferramentas e conhecimentos corretos à extração e transformação de dados não estruturados, estes podem apresentar importantes resultados para a área negocial sem que haja a necessidade de investimentos substanciais em capacidade computacional para processamento e armazenamento de grandes volumes de dados, o que corrobora a tese de que podem ser relevantes para empresas que possuem orçamentos limitados para investimentos em infraestrutura de dados.

Quanto aos sistemas de recomendação desenvolvidos neste estudo, resta claro que os objetivos traçados foram cumpridos com êxito. Em primeira medida destacam-se os dois sistemas de recomendação desenvolvidos e testados: um baseado em vetorização textual por meio de TF-IDF e o outro baseado em vetores de *embeddings* densos e arquitetura de *transformers*.

Neste ponto foram elencadas as qualidades e limitações de cada abordagem, bem como oportunidades de melhoria e possíveis aplicações em âmbito organizacional, destacando-se ainda a simplicidade na implementação de cada um deles e a necessidade de poucos dados sobre os itens para um funcionamento

satisfatório, ressaltando ainda que tanto o sistema de recomendação baseado em TF-IDF quanto aquele baseado em SBERT comprovaram capacidade para emitir ao menos duas recomendações de itens similares ao item base para cada conjunto de dados testado.

O destaque dentre os modelos de recomendação desenvolvidos fica por conta do algoritmo SBERT. Além de demonstrar-se mais preciso que o seu predecessor, este modelo provou ainda sua versatilidade e eficiência ao considerar o contexto semântico, algo tão preponderante na língua portuguesa. Com isto, ao não deixar-se levar apenas por coincidências lexicais entre palavras-chave, este modelo mostrou-se relevante e fiável, além de possibilitar ao seu operador o controlo de qualidade das recomendações por meio da análise dos valores de similaridade entre as frases, conforme caso de aplicação do modelo exposto no Capítulo 4.

As diferenças de desempenho entre os modelos no que diz respeito à qualidade das recomendações emitidas, capacidade de perceção de contexto e generalização em diferentes domínios e aptidão para recomendação de itens sequenciais foram apresentadas com mais detalhes na seção de considerações finais sobre os resultados do capítulo 3.

Ressalta-se ainda que, a despeito dos resultados positivos e satisfatórios obtidos ao longo da construção deste projeto, o principal objetivo fora atingido: houve um avanço significativo do aluno no que diz respeito à aquisição de conhecimento sobre técnicas de análise de dados não estruturados, bem como no que tange ao funcionamento e aplicação das técnicas de PLN ao caso concreto e utilização dos métodos da ciência de dados na resolução de problemas organizacionais reais.

Por fim, importa realçar o caso prático descrito na secção 4 — USE CASE: aplicação do recomendador SBERT aos dados da política nacional de desenvolvimento de pessoas — que demonstra a transposição concreta do desenvolvimento académico para o ambiente organizacional. No caso em tela, o protótipo implementado permitiu automatizar a correspondência entre necessidades de formação e oferta de cursos, reduzindo um processo anteriormente moroso, que exigia equipas inteiras e dias de trabalho, para uma verificação humana final sobre um ficheiro com recomendações gerado automaticamente, com tempos de processamento da ordem dos minutos para milhares de registos analisados. A escolha do modelo SBERT, o refinamento do pré-processamento textual e a definição de *thresholds* de similaridade revelaram-se decisivos para conferir fiabilidade, economia de recursos e escalabilidade da solução. Este exemplo corrobora a hipótese de que o trabalho académico desenvolvido não se limitou à validação teórica: forneceu métodos, parâmetros e um protótipo operacional que foram essenciais para a aplicação eficiente e fiável no contexto real da PNDP/ENAP, de modo a confirmar a relevância prática e o impacto potencial da investigação apresentada neste projeto.

5.1.1 **Desafios e limitações**

Dentre os desafios encontrados ao longo do estudo, destacam-se a dificuldade em encontrar uma base consolidada com dados a respeito de livros em Língua Portuguesa (europeia ou brasileira), o que foi contornado por meio da exploração de *datasets* com ISBNs e sua vinculação à *Google Books API*, e a dificuldade na aplicação de métodos objetivos de mensuração da qualidade das recomendações emitidas pelos sistemas desenvolvidos, haja vista tratar-se de abordagem não supervisionada onde não existem rótulos para mensuração de precisão, *recall* ou *f1-score*.

Quanto às limitações identificadas nos sistemas de recomendação desenvolvidos, a elevada esparsidade da matriz de vetores mostrou-se como um fator limitante para o desempenho do sistema TF-IDF, bem como a falta de capacidade de perceber relações semânticas e contextuais entre os itens. Estas limitações foram superadas com uso do SBERT para geração dos vetores de *embeddings*, demonstrando-se como alternativa para mitigação das inconsistências do seu predecessor. Entretanto, no que diz respeito à capacidade de recomendação de itens em sequência, observou-se uma limitação do sistema SBERT se comparado ao TF-IDF.

5.2 Investigação futura

Como oportunidade de investigação em trabalhos futuros, destacam-se a implementação de um banco de dados vetorial para armazenamento dos *embeddings* gerados pelo modelo SBERT com fito de aumentar-se a eficiência na emissão das recomendações, a possível utilização de outros modelos da biblioteca *Sentence Transformers* para além do *paraphrase-multilingual-MiniLM-L12-v2*, de modo a elevar o número de dimensões do espaço vetorial gerado e captar ainda mais nuances contextuais e explorar métodos híbridos que combinem os pontos fortes do TF-IDF (como a recomendação de itens sequenciais) às vantagens do SBERT (captura de contexto semântico aprofundado) para aproximar-se ainda mais da máxima eficácia na produção das recomendações baseadas em texto.

Por fim, todos os códigos desenvolvidos ao longo deste estudo encontram-se disponíveis para consulta em repositório próprio do Github.

6 Bibliografia

Aggarwal, C. C. (2016). *Recommender Systems*. Springer International Publishing. https://doi.org/10.1007/978-3-319-29659-3

Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001). On the Surprising Behavior of Distance Metrics in High Dimensional Space. *Proceedings of the 8th International Conference on Database Theory (ICDT)*, 420–434. https://doi.org/10.1007/3-540-44503-X_27

Azeroual, O., & Koltay, T. (2022). RecSys Pertaining to Research Information with Collaborative Filtering Methods: Characteristics and Challenges. *Publications*, *10*(2), 17. https://doi.org/10.3390/publications10020017

Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python* (1st ed.). O'Reilly Media, Inc.

Burke, R. (2002). *User Modeling and User-Adapted Interaction*, *12*(4), 331–370. https://doi.org/10.1023/a:1021240730564

Chowdhary, K. R. (2020). Natural Language Processing. Em *Fundamentals of Artificial Intelligence* (pp. 603–649). Springer India. https://doi.org/10.1007/978-81-322-3972-7_19

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. https://arxiv.org/abs/1810.04805

Harris, Z. S. (1954). Distributional Structure. *<i>WORD</i>*, *10*(2–3), 146–162. https://doi.org/10.1080/00437956.1954.11659520

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, *22*(1), 5–53. https://doi.org/10.1145/963770.963772

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning. Em *Springer Texts in Statistics*. Springer New York. https://doi.org/10.1007/978-1-4614-7138-7

Jannach, D., Lerche, L., & Zanker, M. (2018). Recommending Based on Implicit Feedback. Em *Social Information Access* (pp. 510–569). Springer International Publishing. https://doi.org/10.1007/978-3-319-90092-6_14

Jurafsky, D., & Martin, J. H. (2025). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models* (3rd ed.). https://web.stanford.edu/~jurafsky/slp3/

Kamath, S. S., & Kanakaraj, M. (2015). Natural language processing-based e-news recommender system using information extraction and domain clustering. *International Journal of Image Mining*, *1*(1), 112.

https://doi.org/10.1504/ijim.2015.070031

Kılıç, E. (2023). *Hybrid Approaches to Improve Scalability in Collaborative Filtering Algorithms*.

Krovetz, R. (1993). Viewing morphology as an inference process. *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '93*, 191–202. https://doi.org/10.1145/160688.160718

Luhn, H. P. (1957). A Statistical Approach to Mechanized Encoding and Searching of Literary Information. *IBM Journal of Research and Development*, *1*(4), 309–317. https://doi.org/10.1147/rd.14.0309

Melville, P., & Sindhwani, V. (2017). Recommender Systems. Em *Encyclopedia of Machine Learning and Data Mining* (pp. 1056–1066). Springer US. https://doi.org/10.1007/978-1-4899-7687-1_964

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space*. https://arxiv.org/abs/1301.3781

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to WordNet: An on-line lexical database. *International journal of lexicography*, *3*(4), 235–244.

Neelakantan, A., Shankar, J., Passos, A., & McCallum, A. (2015). *Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space*. https://arxiv.org/abs/1504.06654

Patwardhan, N., Marrone, S., & Sansone, C. (2023). Transformers in the Real World: A Survey on NLP Applications. *Information*, *14*(4), 242. https://doi.org/10.3390/info14040242

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). *Improving language understanding by generative pre-training*.

Reimers, N., & Gurevych, I. (2019, novembro). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. http://arxiv.org/abs/1908.10084

Ricci, F., Rokach, L., & Shapira, B. (2011). *Introduction to Recommender Systems Handbook* (F. Ricci, L. Rokach, & B. Shapira, Eds.). Springer. https://doi.org/10.1007/978-0-387-85820-3

Riesz, F. (1910). Untersuchungen □ber Systeme integrierbarer Funktionen. *Mathematische Annalen*, *69*(4), 449–497. https://doi.org/10.1007/bf01457637

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, *24*(5), 513–523. https://doi.org/10.1016/0306-4573(88)90021-0

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613–620. https://doi.org/10.1145/361219.361220

Schafer, J. B., Konstan, J. A., & Riedl, J. (2001). *Data Mining and Knowledge Discovery*, *5*(1/2), 115–153. https://doi.org/10.1023/a:1009804230409

Schein, A. I., Popescul, A., Ungar, L. H., & Pennock, D. M. (2002, agosto). Methods and metrics for cold-start recommendations. *Proceedings of the 25th annual international ACM SIGIR conference on Research*

and development in information retrieval. https://doi.org/10.1145/564376.564421

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Son, J., & Kim, S. B. (2017). Content-based filtering for recommendation systems using multiattribute networks. *Expert Systems with Applications*, *89*, 404–412. https://doi.org/10.1016/j.eswa.2017.08.008

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, *28*(1), 11–21. https://doi.org/10.1108/eb026526

Sun, T.-X., Liu, X.-Y., Qiu, X.-P., & Huang, X.-J. (2022). Paradigm Shift in Natural Language Processing. *Machine Intelligence Research*, *19*(3), 169–183. https://doi.org/10.1007/s11633-022-1331-6

Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

Zhang, W., Yoshida, T., & Tang, X. (2011). A comparative study of TF*IDF, LSI and multi-words for text classification. *Expert Systems with Applications*, *38*(3), 2758–2765. https://doi.org/10.1016/j.eswa.2010.08.066

Zohuri, B., & McDaniel, P. (2022). Artificial intelligence driven by machine learning & deep learning. Em *Transcranial Magnetic and Electrical Brain Stimulation for Neurological Disorders* (pp. 317–334). Elsevier. https://doi.org/10.1016/b978-0-323-95416-7.00011-0