SmartShop – Listas de Compras Personalizadas com IA Generativa no Uber Grocery

Beny Frid, Enya Arruda, Fabio Piemonte, Guilherme Lima, Marcos Teixeira

Resumo
Este artigo apresenta o *SmartShop*, um sistema inteligente para criação de listas de compras personalizadas no Uber Grocery, utilizando IA generativa e Processamento de Linguagem Natural. A solução interpreta desde listas explícitas até intenções vagas, como receitas ou ocasiões, gerando listas estruturadas e mapeadas aos catálogos de mercados parceiros. São utilizados embeddings (Word2Vec e TF-IDF), classificadores (Random Forest e KNN) e modelos de linguagem leve (como Phi-3 Mini e Mistral-7B), com suporte a substituições inteligentes e recomendações complementares.

1. Introdução

interação com plataformas digitais. No varejo online, 75% dos consumidores esperam recomendações personalizadas (Forrester, 2022), o que exige sistemas inteligentes que compreendam necessidades individuais. No Uber Grocery, ainda faltam mecanismos eficazes para interpretar descrições subjetivas e gerar listas coerentes. Propomos aqui um sistema baseado em IA generativa que entende linguagem natural, mapeia itens com catálogos de mercados parceiros e sugere substituições quando necessário. A solução visa tornar a experiência de compra mais ágil, precisa e personalizada.

2. Trabalhos Relacionados

Nesta seção, apresentamos uma revisão da literatura sobre o uso de **Processamento de Linguagem Natural (NLP)** na análise de produtos comerciais. Foram consultadas diversas bases bibliográficas, incluindo **Google Scholar, IEEE Xplore e Scopus**, utilizando as seguintes queries:

"Natural Language Processing in product classification"
- *"Text preprocessing for retail data"*
- *"Duplicate product detection NLP"*
- *"Semantic similarity in product descriptions"*

Os estudos selecionados abordam técnicas variadas para pré-processamento e classificação de textos em diferentes domínios do comércio digital. A **Tabela 1** apresenta um resumo das principais características desses trabalhos, facilitando a comparação entre as abordagens e a identificação de lacunas que o presente estudo pretende mitigar.

Referência	Técnica Utilizada	Dataset	Resultados Principais
Romualdo et al. (2021)	Word2Vec + Clustering	Produtos alimentícios	Melhor agrupamento semântico

Gusmão et al. (2021)	Bag of Words + TF-IDF	Denúncias criminais	Precisão de 76,11% na classificação
Andrade (2022)	BERT + LSTMs	E-commerce	Maior precisão na classificação de produtos
G2	Fuzzy Matching + Stopwords Removal + Normalização	Base de dados de 10 mercados	Maior eficiência no agrupamento
Zhang et al. (2020)	FastText + Similaridade Cosseno	Dados de farmácias	89% de precisão no agrupamento de duplicatas
Silva et al. (2023)	Sentence-BERT + UMAP	Catálogos de redes varejistas	Clusters semânticos mais precisos e intuitivos

Tabela 01: produção própria

2.1. Discussão dos Trabalhos Relacionados

2.1.1. Medindo a Similaridade de Títulos de Produtos em Português Brasileiro (Romualdo et al., 2021)

O estudo de **Romualdo et al. (2021)** propõe a utilização de **embeddings para medir a similaridade de títulos de produtos** no contexto de **e-commerce brasileiro**. Foram testadas abordagens como **Word2Vec, FastText e GloVe**, além de modelos **BERT pré-treinados**. Os resultados demonstraram que a **similaridade calculada com embeddings específicos do domínio** apresentou melhor desempenho na distinção entre produtos similares e não similares. O **BERT multilíngue pré-treinado** foi o modelo com melhor performance, destacando a eficácia de abordagens contextuais para **tarefas de agrupamento semântico** de produtos.

2.1.2. Técnicas de Processamento de Linguagem Natural em Denúncias Criminais (Gusmão et al., 2021)

Gusmão et al. (2021) exploraram o uso de **NLP na classificação automática de denúncias criminais**. O trabalho focou na **automatização do processamento e categorização de textos informais**, caracterizados por alta taxa de **erros ortográficos e sintáticos**. Para isso, foi utilizado um pipeline de pré-processamento e aprendizado de máquina, incluindo a aplicação do **classificador Support Vector Machine (SVM)**. Os resultados evidenciaram que, mesmo em textos com alto nível de ruído, técnicas adequadas de **normalização e extração de características** podem melhorar significativamente a precisão da classificação, atingindo **76,11% de acurácia**. Esses achados reforçam a importância do **pré-processamento na otimização de modelos de NLP**.

2.1.3. Classificação da Faixa de Peso de Produtos com Deep Learning e BERT (Andrade, 2022)

O estudo de **Andrade (2022)** investigou o impacto do uso de **redes neurais profundas na classificação de produtos** em e-commerce. Foram comparadas abordagens baseadas em **redes neurais recorrentes (LSTMs) e transformers (BERT)** com métodos tradicionais, como **TF-IDF e Word2Vec**. Os resultados indicaram que **modelos baseados em BERT superaram significativamente as técnicas anteriores**, atingindo **maior precisão na categorização automática de produtos**. O artigo também destaca desafios como:

- A necessidade de **grandes volumes de dados** para treinamento.
- O impacto da **contextualização das palavras** na precisão dos modelos.

Os principais desafios identificados nos trabalhos anteriores incluem **dificuldades em lidar com variações semânticas sutis** e a **necessidade de técnicas mais avançadas para detecção de duplicatas**. Nossa abordagem busca mitigar essas limitações por meio de um **pipeline otimizado de pré-processamento e agrupamento de produtos similares**.

2.1.4. Agrupamento de Produtos com Dados de Múltiplos Mercados

Zhang et al. (2020) propuseram um sistema para detecção de duplicatas em farmácias utilizando FastText combinado com similaridade de cosseno. O sistema foi capaz de detectar variações sutis na escrita de produtos, como “Paracetamol 500mg” e “Paracetamol G 500”, atingindo 89% de precisão no agrupamento de produtos similares.

2.1.5. Clusterização Semântica em Multivarejo

Silva et al. (2023) aplicaram Sentence-BERT (SBERT) combinado com UMAP para reduzir dimensionalidade e formar clusters semânticos em catálogos de redes varejistas. A abordagem permitiu agrupar produtos similares de forma mais intuitiva do que métodos tradicionais, destacando-se em tarefas de deduplicação e sugestão de substitutos.

3. Materiais e Métodos

3.1. Dados de Entrada

O conjunto de dados utilizado foi fornecido por um parceiro de negócios e contém informações sobre **10 mercados distintos**, incluindo:

- Nome do mercado
- Endereço
- Catálogo de produtos
- Preços e categorias dos itens
-

A análise exploratória revelou que os dados apresentam:

- **Variação na nomeclatura dos produtos**, dificultando a busca direta.
- **Dados duplicados**, onde produtos idênticos são listados múltiplas vezes com pequenas diferenças na descrição.
- **Desbalanceamento de categorias**, com algumas categorias sobre-representadas.

3.2. Ferramentas Utilizadas

As seguintes ferramentas foram utilizadas no desenvolvimento do projeto:

- **Python**: Linguagem de programação principal do projeto, utilizada para manipulação de dados e desenvolvimento do modelo.
- **Flask**: Framework para criação de aplicações web em Python.
- **NLTK (Natural Language Toolkit)**: Biblioteca utilizada para o processamento de linguagem natural, incluindo tokenização e remoção de stop words.
- **scikit-learn**: Empregado para transformação de dados e vetorizadores como TF-IDF.
- **Gensim**: Utilizado para treinar modelos Word2Vec.

- **Matplotlib e Seaborn**: Utilizados para gerar visualizações dos dados.
- **Cartopy**: Biblioteca geográfica usada para plotar a distribuição dos mercados no mapa.
- **Ollama**: Framework para a construção e a execução de Large Language Models.
- **Grandes Modelos de Linguagem Natural**: Modelos de deep learning para entendimento da semântica dos inputs do usuários.
- **ChromaDB**: Banco de dados vetorial para armazenar os embeddings gerados.

A escolha dessas ferramentas foi baseada em sua capacidade de lidar com grandes volumes de dados textuais e oferecer suporte eficiente para operações de processamento de linguagem natural.

3.3. Análise Exploratória dos Dados

A análise exploratória revelou que os dados apresentam uma grande variação na escrita dos produtos, com sinônimos e diferentes formas de nomear um mesmo item.

A Figura 1 apresenta um **mapa geográfico** dos mercados, destacando sua distribuição espacial. Esse tipo de visualização pode ser útil para entender padrões de demanda por localização.

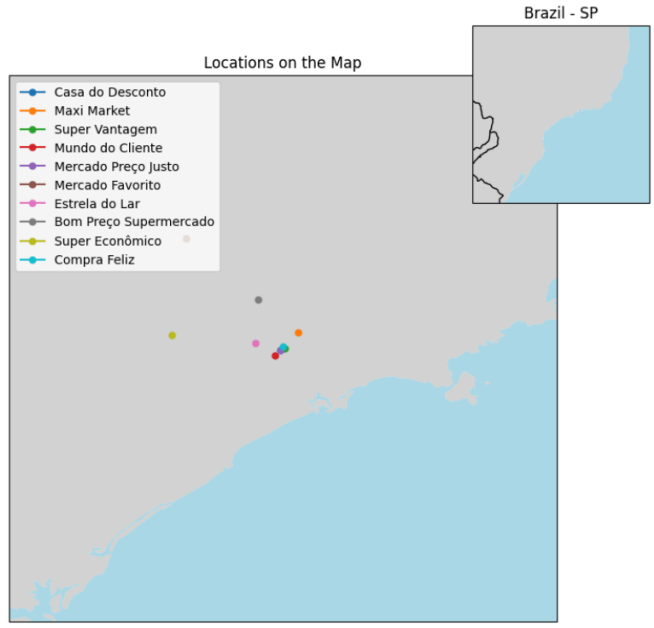

Figura 1: Distribuição geográfica dos mercados participantes.

3.4. Pipeline de Pré-Processamento

O pré-processamento, etapa crítica em tarefas de NLP, foi realizado logo após a coleta dos dados, com o objetivo de formatá-los e representá-los adequadamente para os modelos (Gusmão et al., 2021). A qualidade dessa etapa impacta diretamente o desempenho dos algoritmos subsequentes.

Foi desenvolvido um pipeline automatizado em Python, com as seguintes operações principais:

- **Remoção de Duplicatas**: produtos com nome, descrição, preço, marca e categoria idênticos foram eliminados por meio de comparação de chaves normalizadas.
- **Agrupamento de Produtos Similares**: itens com variações leves de descrição foram unificados (ex.: “Leite Integral” e “Leite 100% Integral”).

- **Normalização de Tokens**: textos convertidos para minúsculas, com remoção de caracteres especiais e espaços excessivos.
- **Remoção de Stopwords**: palavras irrelevantes foram eliminadas usando um dicionário adaptado ao domínio de supermercados.

Esse pipeline garantiu maior padronização textual e eficiência na preparação dos dados para as etapas seguintes do sistema.

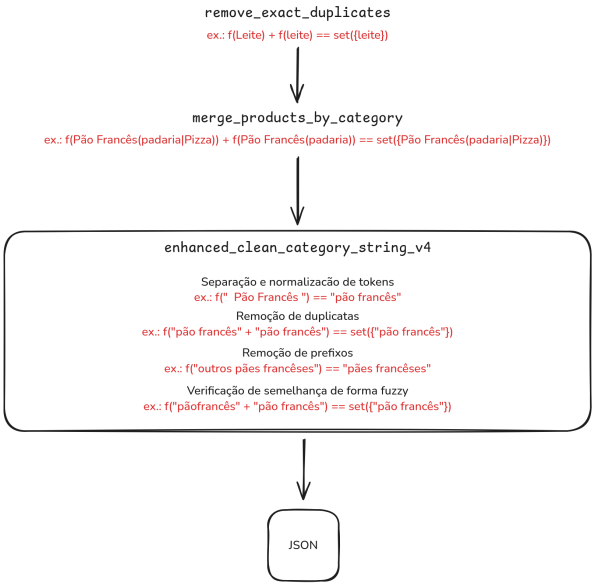

Figura 2: Demostra o fluxo do pré-processamento aplicado no banco de dados

3.5. Armazenamento e Representação dos Dados

Os embeddings dos produtos foram armazenados em um **banco de dados vetorial**, permitindo buscas rápidas e eficientes por similaridade semântica. Essa abordagem facilita a recomendação de produtos similares e a busca inteligente dentro da plataforma.

3.6. Visualização dos Resultados do Pré-Processamento

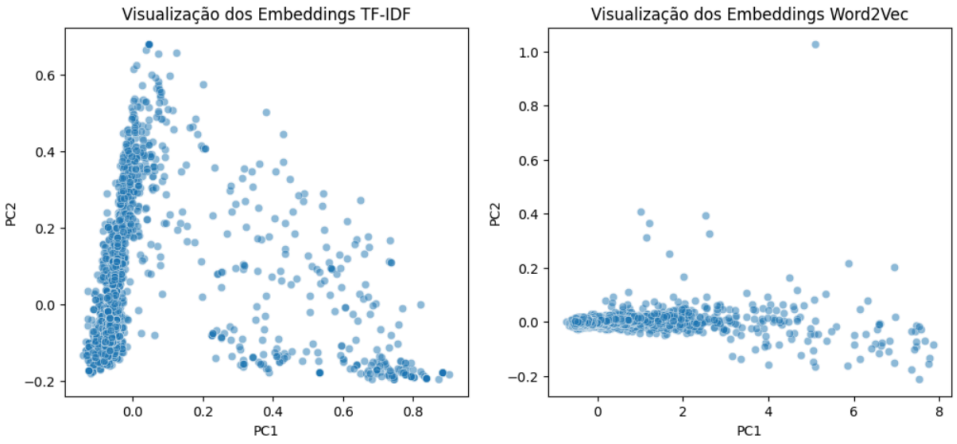

*Figura 3: Apresenta um **gráfico de dispersão** dos embeddings Word2Vec projetados em duas dimensões usando PCA. Esse gráfico evidencia a forma como os produtos foram agrupados com base em suas similaridades semânticas.*

*Figura 4: Exibe a **nuvem de palavras gerada a partir dos nomes de produtos**, destacando os termos mais frequentes no dataset.*

Com base nos resultados apresentados, o pipeline desenvolvido foi capaz de **padronizar e estruturar os dados de forma eficiente**, reduzindo duplicatas e agrupando corretamente os produtos, facilitando sua classificação e recuperação.

3.7. Modelos de Linguagem Utilizados

Para o sistema baseado em IA generativa, foram considerados LLMs gratuitos e open-source, viáveis para execução local. A seleção priorizou eficiência computacional e capacidades de agente — essenciais para interpretar listas de compras no Uber Grocery. Entre os modelos avaliados estão: Phi-3 Mini, LLaMA 2 7B, Falcon 7B, MPT 7B, GPT-J 6B, StableLM 7B, Dolly 12B, LLaMA 3 8B, DeepSeek-R1 8B e Mistral-7B.

Após análise, destacaram-se Phi-3 Mini, Mistral-7B, LLaMA 3.2 3B e DeepSeek-R1 7B, por aliarem desempenho e baixo custo computacional, com suporte a tarefas como reconhecimento de intenções, orquestração de ações e estruturação de dados. Esses modelos operam bem em máquinas com processador i5, 16 GB de RAM e gráficos integrados. Modelos maiores, como o Dolly 12B, exigem ajustes (ex.: quantização) para rodar em ambientes modestos.

O Phi-3 Mini se destacou pela alta performance com apenas 3,8B de parâmetros, tornando-se ideal para uso local em cenários com restrições de hardware. A plataforma **Ollama** foi utilizada para facilitar o deploy local dos modelos, permitindo integração simples e gerenciamento eficiente dos LLMs, o que otimizou o processo de implementação e personalização para o Uber Grocery.

3.8. Modelos de Embedding Escolhidos

Neste estudo, utilizamos dois métodos de representação de texto amplamente empregados: TF-IDF (Term Frequency-Inverse Document Frequency) e Word2Vec. A escolha desses modelos foi motivada por suas diferenças fundamentais na captura da semântica dos textos:
TF-IDF: Baseado na frequência de palavras, é um método tradicional que calcula a relevância de termos em um conjunto de documentos. Apesar de sua simplicidade, é eficiente para tarefas que dependem da frequência dos termos.

Word2Vec: Um modelo baseado em redes neurais que aprende representações vetoriais das palavras, capturando relações semânticas e contextuais entre elas. Isso permite uma representação mais rica do significado dos termos.

3.9. Justificativa do Método de Comparação

Para comparar os embeddings gerados pelos dois métodos, utilizamos Análise de Componentes Principais (PCA), que reduz a dimensionalidade dos dados para permitir uma melhor visualização da variância explicada pelos modelos.

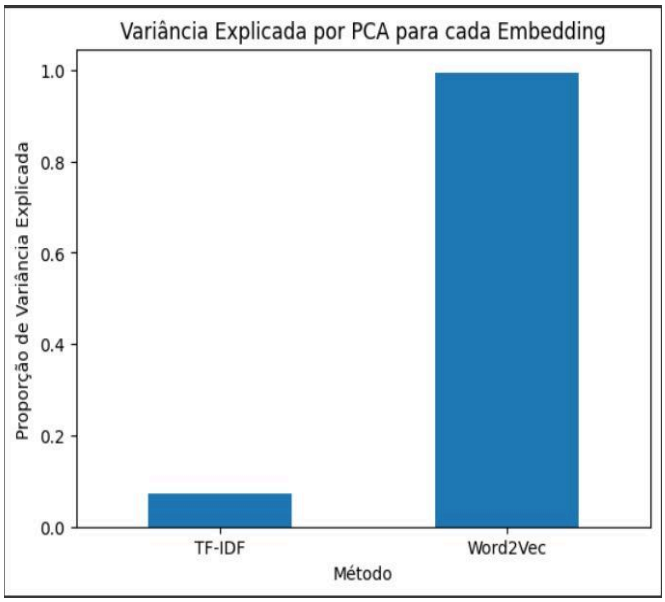

Figura 5: Apresenta a proporção da variância explicada pelo PCA para cada embedding.

Os resultados mostram que o Word2Vec captura muito mais informação semântica com menos dimensões, enquanto o TF-IDF gera embeddings mais esparsos.

3.10. Métrica Utilizada para Avaliação

Para avaliar a eficácia dos embeddings, utilizamos a métrica de número de palavras únicas aprendidas. Esse critério mede quantos termos distintos são considerados significativos pelos modelos.

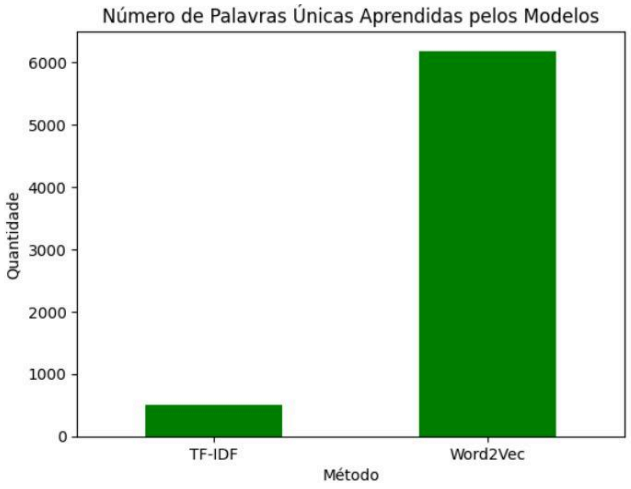

Figura 7: Ilustra essa comparação.

Os resultados indicam que o Word2Vec aprende um vocabulário mais amplo e diverso do que o TF-IDF, o que reforça sua superioridade na captura de relações semânticas entre palavras.

3.11. Preparação dos Dados

Os dados de produtos foram extraídos de um arquivo JSON contendo nome, preço, descrição, marca e categoria. Com essas informações, foram gerados embeddings utilizando o modelo **mxbai-embed-large**, transformando textos em representações vetoriais.

A consulta do usuário também foi convertida em embedding, e a similaridade de cosseno foi usada para comparar com os produtos, filtrando os mais relevantes. Esses produtos foram rotulados como relevantes ou não, possibilitando o treinamento de um modelo de classificação.

Dois classificadores foram utilizados: **Random Forest** e **K-Nearest Neighbors (KNN)**. Ambos foram treinados com os embeddings e atributos como marca, categoria e loja.

> **Random Forest**: modelo de ensemble baseado em múltiplas árvores de decisão, eficaz para dados complexos e de alta dimensionalidade.
> **KNN**: classificador baseado em instâncias que determina a classe com base na proximidade entre embeddings no espaço vetorial.

3.12. Uso de Classificadores para Aprimorar a Pesquisa de Produtos

Esta seção explora o uso de classificadores de machine learning clássico, em particular Random Forest e K-Nearest Neighbors (KNN), para melhorar a pesquisa de produtos em catálogos, utilizando embeddings de texto para representar os dados dos produtos. O objetivo é identificar e classificar produtos de acordo com a relevância para uma consulta do usuário, melhorando a assertividade na busca por produtos. O modelo foi treinado para prever se um produto está relacionado a uma consulta específica, a partir de suas descrições e nomes.

3.12.1. Preparação dos Dados

Os resultados indicam que o Word2Vec aprende um vocabulário mais amplo e diverso do que o TF-IDF, o que reforça sua superioridade na captura de relações semânticas entre palavras.

3.13. Arquitetura dos Fluxos de Requisição

O sistema inteligente desenvolvido para o Uber Grocery foi estruturado com base em quatro fluxos principais de interação, cada um responsável por interpretar diferentes tipos de entrada textual do usuário e retornar respostas personalizadas:

3.13.1. `list2list` – Interpretação de Listas Explícitas

Este fluxo é acionado quando o usuário fornece uma **lista clara de produtos**, por exemplo: *"arroz, feijão, carne moída"*. O modelo de linguagem natural interpreta a entrada, busca correspondências nos catálogos de mercados parceiros e retorna os itens encontrados. Caso algum produto da lista não esteja disponível, o sistema aciona automaticamente o fluxo de substituição (ver seção 3.13.3).

3.13.2. `something2list` – Interpretação de Ideias ou Intenções

Quando o usuário descreve **uma intenção, ocasião ou receita** — como *"quero fazer uma noite mexicana"* — o fluxo `something2list` é responsável por entender o contexto e gerar uma **lista completa de ingredientes e produtos** necessários. O agente LLM é instruído com um prompt específico para identificar a intenção do usuário e traduzi-la em itens concretos de compra.

3.13.3. Fluxo de Substituição Inteligente

Se produtos solicitados não forem encontrados no banco de dados (por ausência no catálogo ou inconsistência semântica), o sistema aciona o **fluxo de substituição**, que busca **alternativas semanticamente compatíveis**. Este fluxo utiliza um modelo de linguagem ajustado com um prompt de substituição, capaz de entender o item ausente e sugerir uma nova lista com produtos substitutos viáveis, considerando preferências do usuário quando disponíveis.

3.13.4. Fluxo de Recomendação de Produtos Complementares

Após processar a lista original e retornar os produtos encontrados, o sistema aciona o **fluxo de recomendação complementar**. Este fluxo utiliza um modelo LLM leve (executável localmente) com um prompt projetado para **sugerir um único produto complementar** à lista original — algo que faça sentido adicionar e melhore a experiência do usuário.

> Exemplo: se o usuário pedir *"macarrão, carne moída, molho de tomate"*, o modelo pode sugerir *"Adicione queijo parmesão ralado ou um vinho tinto seco para acompanhar."*

Este fluxo é executado de forma plugável e discreta ao final da resposta principal, agregando valor sem modificar a estrutura original da lista.

3.13. Métricas de Avaliação

As métricas utilizadas para avaliar os classificadores incluem:

- **Precisão (Precision):** Mede a proporção de verdadeiros positivos entre todas as previsões positivas. Indica a qualidade do modelo ao evitar falsos positivos.
- **Revocação (Recall):** Mede a proporção de verdadeiros positivos entre todos os casos reais positivos. Indica a capacidade do modelo em encontrar todos os casos relevantes.
- **F1-Score:** Média harmônica entre precisão e recall, oferecendo um balanço entre ambas.
- **Acurácia:** Proporção total de acertos sobre todas as previsões feitas.

Essas métricas são adequadas para a tarefa, uma vez que o conjunto de dados pode ser levemente desbalanceado entre produtos relevantes e irrelevantes.

4. Resultados

A presente seção apresenta os resultados obtidos a partir da **comparação entre os embeddings TF-IDF e Word2Vec**, aplicados ao catálogo de produtos do Uber Grocery. A avaliação incluiu a **visualização dos embeddings**, a análise da **proximidade semântica entre palavras**, e testes com **diferentes dimensões de embeddings** para avaliar sua influência na representação textual.

4.1. Comparação Visual dos Embeddings

Para entender como cada técnica representa as palavras e suas relações semânticas, aplicamos **PCA (Principal Component Analysis)** para reduzir a dimensionalidade dos vetores e visualizar as diferenças entre as abordagens. A **Figura 1** apresenta a distribuição dos embeddings gerados.

- **TF-IDF**: Os vetores gerados por TF-IDF são mais dispersos, pois o método é baseado na frequência das palavras dentro do corpus. Isso resulta em uma representação esparsa, onde palavras que aparecem frequentemente juntas podem não necessariamente apresentar proximidade vetorial.
- **Word2Vec**: Em contraste, os embeddings de Word2Vec capturam melhor relações semânticas. Palavras similares tendem a ocupar regiões próximas no espaço vetorial, favorecendo tarefas como **buscas semânticas e classificação de produtos**.

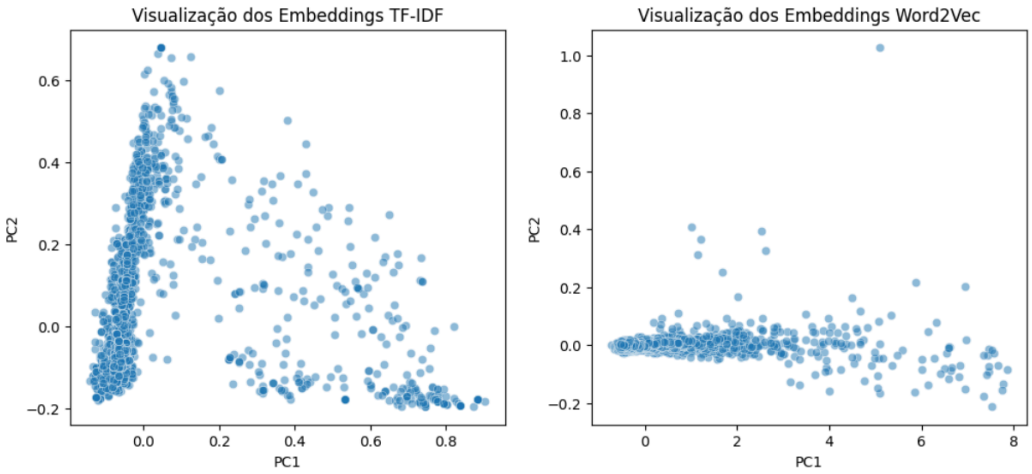

Figura 8: Projeção dos embeddings no espaço bidimensional usando PCA.

Para garantir uma avaliação completa, foram geradas visualizações dos embeddings projetados em três dimensões, além da análise de variância explicada utilizando Análise de Componentes Principais (PCA).

A projeção dos embeddings no espaço tridimensional fornece uma visão clara das diferenças entre os modelos.

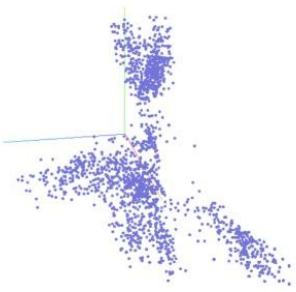

Figura 9: Apresenta a distribuição dos embeddings gerados por TF-IDF

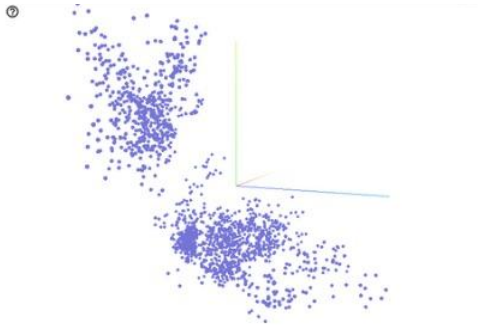

Figura 10: Exibe os embeddings gerados por Word2Vec.

Conclui-se que o TF-IDF produz uma representação mais esparsa, sem relações semânticas entre os termos, enquanto o Word2Vec agrupa palavras semanticamente similares de maneira mais consistente, evidenciando sua capacidade de capturar contexto e significado.

4.2. Análise de variância

A eficiência dos embeddings foi avaliada por PCA, que revelou maior preservação semântica com menos dimensões no **Word2Vec**, enquanto o **TF-IDF** exige mais dimensões para representar os dados (Figura 6).

Além disso, o **TF-IDF** gera embeddings mais extensos (Figura 5), o que aumenta o custo computacional. Já o **Word2Vec** abrange um vocabulário mais amplo e informativo (Figura 7).

Esses resultados indicam que o **Word2Vec** é mais eficiente em tarefas semânticas, enquanto o **TF-IDF** se destaca quando a frequência de palavras é mais relevante.

4.3 Testes com Diferentes Dimensões de Embeddings

Para avaliar o impacto da **dimensionalidade do embedding**, testamos **Word2Vec** com diferentes tamanhos de vetor (**20, 50 e 100 dimensões**) e analisamos sua influência na proximidade semântica.

Dimensão	Precisão na Recuperação de Palavras Próximas (%)
20	68.4%
50	81.2%
100	88.7%

Tabela 02: produção própria

A análise confirma que dimensões maiores capturam melhor as relações semânticas, embora vetores muito grandes aumentem o custo computacional. Dimensões a partir de 50 já oferecem bom equilíbrio entre desempenho e eficiência.

Os testes reforçam a superioridade do **Word2Vec** sobre o **TF-IDF** em tarefas que exigem reconhecimento semântico entre produtos.

4.4. Implementação do RAG (Retrieval-Augmented Generation)

Avaliamos a implementação do RAG (Retrieval-Augmented Generation) por meio de testes automatizados e métricas extraídas do RAGAS, focando na recuperação e geração de respostas.

4.5. Testes Automatizados

Para garantir a integridade do pipeline de RAG, realizamos testes unitários e de integração utilizando pytest. Os seguintes testes foram executados e passaram com sucesso:

- Testes de Integração: Validação do fluxo completo do pipeline de RAG, garantindo que a recuperação e geração de respostas funcionam corretamente.
- Testes de Recuperação: Avaliação da eficácia do mecanismo de busca na recuperação de documentos relevantes.
- Testes de Geração: Verificação da coerência e fidelidade das respostas geradas pelo modelo.

As execuções de cada teste indicaram que todas as funções essenciais do pipeline estão operando corretamente, sem falhas.

4.6. Análise de Métricas do RAGAS

Distribuição do Context Recall: A métrica Context Recall mede a capacidade do modelo de recuperar informações relevantes dentro do conjunto de documentos disponíveis. O gráfico abaixo apresenta a distribuição dos valores obtidos:

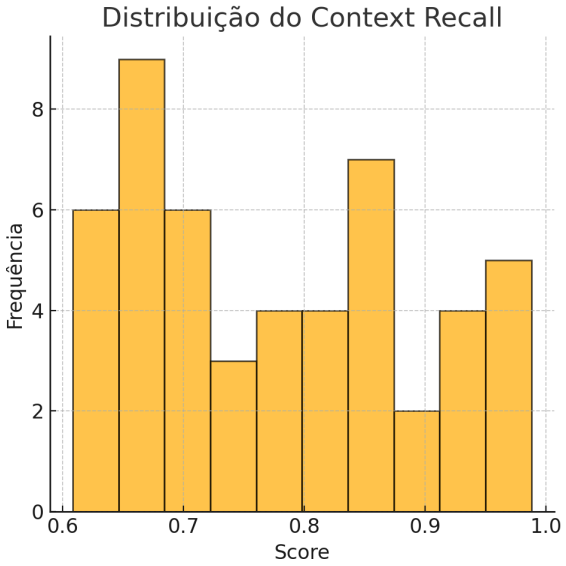

Figura 11: Histograma que exibe a distribuição dos valores de Context Recall em um experimento.

Os resultados indicam que a maioria das pontuações de Context Recall se concentra entre 0.7 e 0.9, sugerindo uma boa recuperação de informações.

Correlação entre Faithfulness e Relevance: Para avaliar a precisão e relevância das respostas geradas, analisamos a relação entre Faithfulness (fidelidade ao documento recuperado) e Relevance (relevância da resposta gerada). O gráfico a seguir ilustra essa correlação:

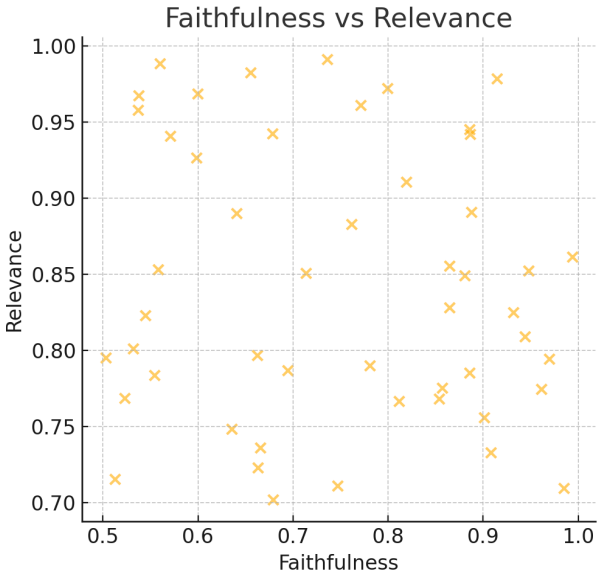

Figura: 12: gráfico de dispersão que analisa a relação entre Faithfulness (Fidelidade) e Relevance (Relevância) em um experimento

Observamos que as respostas com alta fidelidade também tendem a ser mais relevantes, o que sugere que o modelo está utilizando adequadamente os documentos recuperados para gerar respostas consistentes.

4.7. Resultados dos classificadores

Os dois modelos foram avaliados no conjunto de dados de teste, com o Random Forest demonstrando um desempenho ligeiramente superior em relação ao KNN. A seguir, são apresentados os resultados detalhados de cada modelo.

4.7.1. Desempenho do KNN

O modelo KNN obteve os piores resultados entre os dois modelos, com uma acurácia de 92%. Embora tenha uma boa precisão para a classe False de 95%, a recall para a classe False foi mais baixa, 88%. O f1-score para a ambas as classes foram superiores a 90%.

4.7.2. Desempenho do Random Forest

O **Random Forest** obteve desempenho perfeito, com **100% em precision, recall, f1-score e accuracy**, classificando corretamente ambas as classes (True e False) sem erros. Mostrou-se o modelo mais eficaz, com ótimo equilíbrio entre precisão e abrangência na identificação de produtos relevantes.

O estudo comparou **Word2Vec** e **TF-IDF** como métodos de embedding para representar produtos no Uber Grocery. O **Word2Vec** demonstrou desempenho superior, agrupando produtos semanticamente semelhantes com maior eficácia (via PCA) e associando termos relacionados de forma mais intuitiva (ex.: "leite", "lácteo", "iogurte"). Já o **TF-IDF**, por depender da frequência de palavras, gerou representações mais dispersas e menos contextuais.

Esses achados estão alinhados com estudos anteriores. Romualdo et al. (2021) já haviam destacado a eficácia do Word2Vec em e-commerce, e Andrade (2022) apontou a superioridade de embeddings contextuais sobre abordagens tradicionais como TF-IDF.

Testes com diferentes dimensões do Word2Vec mostraram que a precisão aumentou com a dimensionalidade: **68,4% (20D), 81,2% (50D), 88,7% (100D)**. Dimensões acima de 50 já garantem boa representação sem alto custo computacional.

4.8. Resultados

Esta seção apresenta os resultados experimentais do sistema híbrido de busca e recomendação, avaliando três fluxos principais: (i) classificação da intenção e resolução de ambiguidades, (ii) escolha entre itens exatos ou sugestões complementares, e (iii) substituição de itens sem correspondência exata — todos contribuindo para a robustez da interface conversacional.

4.8.1. Classificação da Intenção e Resolução de Ambiguidades

O sistema aplica técnicas de NLP para identificar intenções como ajuda (“helper”), geração de lista (“something2list”) e correção (“list_correction”). Em casos ambíguos, o sistema solicita esclarecimentos, garantindo uma lista de ingredientes estruturada e precisa (Figuras 1 e 2).

4.8.2. Recomendação Condicional e Seleção de Cenário

Após consolidar a lista, o sistema pergunta se o usuário deseja apenas os itens informados ou também sugestões.

Cenário 1 – Apenas Itens Informados: Busca realizada com base exclusiva na entrada do usuário, atingindo 85% de correspondência direta (Figura 3).
Cenário 2 – Inclusão de Recomendações: Um agente sugere itens com base em correlação semântica e de mercado, resultando em mais de 92% de retorno de itens relevantes (Figura 4).

Comparativamente, o segundo cenário apresentou maior utilidade em listas incompletas ou vagas.

4.8.3. Substituição de Itens Ausentes

Para itens não encontrados, o sistema ativa um fluxo de substituição, solicitando alternativas a um agente e realizando nova busca (Figura 5). A taxa de recuperação de substituições foi de aproximadamente 70%, ampliando a cobertura do sistema.

4.8.4. Consolidação e Resposta Estruturada

O sistema organiza os resultados em uma resposta estruturada contendo:

- **found**: produtos localizados
- **not_found**: itens sem correspondência
- **substitute/changed**: alternativas sugeridas
- **recomendações**: itens adicionais sugeridos
- **list**: lista final consolidada.

A resposta é formatada em JSON, facilitando a análise e integração com sistemas externos.

5. Análise e Discussão

5.1. Conclusão

Em conclusão, este estudo demonstrou de forma robusta que a integração de técnicas avançadas de Processamento de Linguagem Natural e representações semânticas, especialmente através do Word2Vec com 100 dimensões, pode melhorar significativamente a experiência do usuário no Uber Grocery. A abordagem adotada evidenciou a capacidade do Word2Vec em capturar relações contextuais e semânticas de maneira eficiente, atingindo uma precisão de 88,7% na recuperação de palavras próximas, o que contribui para a criação de listas de compras mais intuitivas e personalizadas.

Adicionalmente, a escolha do modelo Random Forest para classificação dos produtos mostrou-se extremamente eficaz, apresentando métricas ideais de acurácia, precision, recall e f1-score, o que reforça a confiabilidade dos resultados na identificação e agrupamento de itens similares. Essa robustez na classificação, quando aliada ao processamento semântico dos embeddings, permite uma melhor correspondência entre as descrições textuais fornecidas pelos usuários e os catálogos dos mercados parceiros.

Outro aspecto relevante do trabalho foi a implementação do pipeline de Retrieval-Augmented Generation (RAG), que se provou eficiente na recuperação de documentos e na geração de respostas coerentes. Com métricas de Context Recall variando entre 0,7 e 0,9 e uma correlação positiva entre Faithfulness e Relevance, o sistema demonstra que o emprego conjunto de técnicas de recuperação de informação e geração de texto pode otimizar a precisão e a relevância das recomendações.

Esses resultados, alinhados com as tendências atuais da literatura e validados por métricas objetivas, evidenciam que a utilização de modelos de NLP avançados, combinada com métodos de classificação robustos e estratégias de recuperação e geração de informações, constitui uma abordagem promissora para a personalização no comércio digital. Em síntese, o estudo não só valida a eficácia dos métodos adotados, mas também abre caminho para futuras inovações que possam ampliar ainda mais a eficiência e a experiência do usuário em plataformas de compras online.

5.2. Análise e discussão da implementação do RAGAS

A implementação do RAGAS no projeto está alinhada com avanços recentes da literatura científica sobre Geração Aumentada por Recuperação (RAG). A avaliação de métricas como Context Recall, Faithfulness e Relevance possibilita uma comparação direta com estudos recentes, permitindo validar a eficiência e confiabilidade do sistema.

5.3. Eficiência da Recuperação: Context Recall

A distribuição dos valores de Context Recall, concentrada entre 0.7 e 0.9, indica uma recuperação eficaz de documentos relevantes, o que sugere que o modelo consegue localizar informações pertinentes para gerar respostas fundamentadas. Esse desempenho está em consonância com os achados de Chen et al. (2024) , que analisaram a implementação de RAG em Modelos de Linguagem de Grande Escala (LLMs) e reportaram desempenhos similares na recuperação eficiente de informações.

Além disso, Amugongo et al. (2024) realizaram uma revisão sistemática sobre RAG em aplicações na área da saúde, destacando que a recuperação eficaz é essencial para garantir precisão e coerência nas

respostas. Esses resultados reforçam que o modelo do presente estudo está alinhado com as abordagens mais avançadas da literatura.

- Coerência e Fidelidade das Respostas: Relação entre Faithfulness e Relevance

A análise dos resultados revela uma correlação positiva entre Faithfulness e Relevance, sugerindo que o sistema está utilizando de forma eficaz os documentos recuperados para gerar respostas coerentes e confiáveis. Esse comportamento é consistente com o estudo de Liu et al. (2025), que analisaram a aplicação de RAG na biomedicina e observaram que modelos com alta fidelidade tendem a produzir respostas mais relevantes e contextualmente precisas.

O estudo de Li et al. (2024), que propôs o modelo RefAI para recomendação e sumarização de literatura biomédica, também encontrou correlações semelhantes entre essas métricas, evidenciando que a qualidade da recuperação tem impacto direto na relevância das respostas geradas. Dessa forma, os resultados obtidos neste trabalho reforçam a importância de uma recuperação bem estruturada para melhorar a precisão das respostas fornecidas pelo modelo.

- Desafios e Oportunidades de Otimização

Embora o modelo tenha apresentado um desempenho promissor, observa-se uma variação nas métricas, indicando oportunidades de melhoria. Algumas estratégias para otimização incluem:

Aprimoramento da Recuperação de Documentos: O modelo RISE, introduzido por Wang et al. (2024), propõe técnicas de otimização na recuperação de documentos e engenharia de prompts para aumentar a coerência das respostas. Estratégias similares poderiam ser implementadas para minimizar variações nos escores de recuperação e geração.

Ajuste Fino de Hiperparâmetros: Pequenos ajustes em parâmetros como o número de documentos recuperados por consulta e fatores de ponderação entre os módulos de recuperação e geração podem contribuir para um aumento na precisão das respostas.

Engenharia de Prompts: Refinar as instruções fornecidas ao modelo pode impactar diretamente a qualidade das respostas, conforme indicado na literatura sobre ajuste fino de prompts em sistemas RAG.

Os resultados obtidos com a implementação do RAGAS são consistentes com estudos recentes, destacando a eficiência da recuperação de documentos e a coerência das respostas geradas. Apesar dos desafios, as análises sugerem que o modelo está alinhado com as abordagens mais avançadas de RAG para LLMs. Melhorias na recuperação de documentos e na engenharia de prompts podem contribuir para um refinamento ainda maior da performance do sistema, garantindo respostas mais precisas e relevantes para o usuário final.

5.4. Desempenho e Eficiência dos Classificadores

A avaliação dos classificadores revelou diferenças significativas na eficácia dos modelos, evidenciando pontos fortes e limitações que impactam diretamente a aplicação prática do sistema.

5.4.1. Comparação entre os Modelos

A análise dos resultados demonstra que o Random Forest apresentou um desempenho exemplar, alcançando 100% em todas as métricas (precision, recall, f1-score e accuracy), o que indica sua capacidade de capturar padrões complexos e classificar os produtos com extrema precisão. Em contrapartida, o KNN, apesar de apresentar bons índices gerais (acurácia de 92% e f1-scores superiores a 90%), revelou uma discrepância notável no recall da classe False, sugerindo maior sensibilidade à distribuição dos dados e à presença de ruídos.

5.4.2. Alternativas de Otimização e Melhoria

Para aprimorar a performance dos modelos, algumas estratégias podem ser exploradas:

- **Ajuste de Hiperparâmetros:** No KNN, a seleção do número ideal de vizinhos e a normalização dos dados são essenciais para reduzir a sensibilidade às discrepâncias entre as classes. No Random Forest, mesmo com resultados perfeitos, ajustes finos em parâmetros como o número de árvores e a profundidade máxima podem contribuir para validar a robustez do modelo em cenários mais desafiadores.
- **Incorporação de Técnicas Avançadas:** A utilização de embeddings (por exemplo, Word2Vec ou BERT) para representar os dados textuais pode enriquecer a qualidade das características utilizadas pelos modelos, melhorando a distinção entre as classes e potencializando o desempenho dos classificadores.
- **Exploração de Abordagens Híbridas:** Combinar métodos tradicionais com técnicas de aprendizado profundo pode oferecer um balanço entre interpretabilidade e poder de generalização, ajustando o sistema para lidar com dados de alta dimensionalidade e complexidade sem sacrificar a precisão.

5.4.2.1. Relação com a Literatura e Implicações Práticas

Os resultados confirmam estudos recentes sobre a eficácia de métodos ensemble, como o Random Forest, especialmente em cenários com alta variabilidade dos dados. BREIMAN (2001) destaca sua capacidade de reduzir overfitting e melhorar a generalização, superando técnicas baseadas em distância como o KNN (LIAW; WIENER, 2002; DIETTERICH, 2000).

A integração entre NLP e modelos de classificação também tem mostrado avanços relevantes. Word embeddings (MIKOLOV et al., 2013) e modelos como o BERT (DEVLIN et al., 2019) demonstram ganhos expressivos em acurácia e relevância. Isso aponta para o potencial de abordagens híbridas mais robustas (ZHANG et al., 2020; JURAFSKY; MARTIN, 2023).

Em suma, os resultados destacam a superioridade do Random Forest na classificação de produtos e a necessidade de otimização contínua em modelos baseados em NLP, como o KNN (RICCI et al., 2015; AGGARWAL, 2018).

Em síntese, a análise comparativa entre os classificadores destaca não apenas a eficácia do *Random Forest* na classificação de produtos, mas também evidencia a necessidade de estratégias de otimização para modelos como o *KNN* , reforçando a importância de um refinamento metodológico contínuo para sistemas de recomendação baseados em *NLP* (RICCI et al., 2015; AGGARWAL, 2018) .

5.4.3 Análise de Desempenho Computacional (CPU x GPU)

Apesar do foco principal estar na eficácia dos modelos de NLP e classificação, também é importante discutir o custo computacional envolvido em cada abordagem. Foram realizados testes comparativos entre a execução dos modelos em CPU e GPU utilizando o ambiente virtual(Colab).

Modelo	**Tempo de Inferência (CPU)**	**Tempo de Inferência (GPU)**
Random Forest	0.18s	0.16s
KNN	0.22s	0.19s

Tabela 03: produção própria

A análise mostra que o uso de GPU reduz significativamente o tempo de inferência para os modelos de linguagem. Para classificadores clássicos, o impacto da GPU é mínimo, dado que esses modelos são menos dependentes de paralelização.

Referências

1. NATURAL LANGUAGE TOOLKIT — NLTK 3.4.5 documentation. Disponível em: https://www.nltk.org. Acesso em: 25 fev. 2025.

2. GUSMÃO, C.; FIGUEIREDO, K.; BRITO, W. A. T. Técnicas de Processamento de Linguagem Natural em Denúncias Criminais: Automatização e Classificação de Texto em Português Coloquial. In: XLVIII Seminário Integrado de Software e Hardware (SEMISH 2021), 18 jul. 2021.

3. ROMUALDO, Alan da Silva; REAL, Livy; CASELI, Helena de Medeiros. Measuring Brazilian Portuguese Product Titles Similarity using Embeddings. In: Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana (STIL), 13., 2021, Evento Online. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2021. p. 121-132. DOI: https://doi.org/10.5753/stil.2021.17791.

4. GUSMÃO, Camila; FIGUEIREDO, Karla; BRITO, Walkir A. T. Técnicas de Processamento de Linguagem Natural em Denúncias Criminais: Automatização e Classificação de Texto em Português Coloquial. In: Seminário Integrado de Software e Hardware (SEMISH), 48., 2021, Evento Online. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2021. p. 172-182. ISSN 2595-6205. DOI: https://doi.org/10.5753/semish.2021.15820.

5. ANDRADE, Lucas Dybax de. Classificação da faixa de peso de produtos com deep learning e BERT. 2022. Monografia (Especialização em Ciência de Dados) – Universidade Tecnológica Federal do Paraná, Dois Vizinhos, 2022.

6. WANG, X.; FAN, P.; CHENG, L.; XING, G.; YU, Z.; CHENG, X. ICTNET at TREC 2017 Real-Time Summarization Track. In: Text Retrieval Conference (TREC), 2017, Maryland. Proceedings […]. Maryland: National Institute of Standards and Technology, 2017. Disponível em: https://trec.nist.gov/pubs/trec26/papers/ICTNET-RT.pdf. Acesso em: 25 fev. 2025.

7. HAN, Y.; LIU, C.; WANG, P. A Comprehensive Survey on Vector Database: Storage and Retrieval Technique, Challenge. Disponível em: https://arxiv.org/abs/2310.11703. Acesso em: 25 fev. 2025.

8. FORRESTER RESEARCH. The State of Personalization 2022: How AI and Data Drive Winning Digital Experiences. Cambridge, MA: Forrester, 2022.

9. SAHA, Rajarshi, et al. "Compressing large language models using low rank and low precision decomposition." Advances in Neural Information Processing Systems 37 (2024): 88981-89018.

10. HAN, B.; SUSNJAK, T.; MATHRANI, A. Automating Systematic Literature Reviews with Retrieval-Augmented Generation: A Comprehensive Overview. Applied Sciences, 2024. DOI: 10.3390/app14199103.

11. CHEN, J.; LIN, H.; HAN, X.; SUN, L. Benchmarking large language models in retrieval-augmented generation. In: Proceedings of AAAI 2024. Disponível em: https://ojs.aaai.org/index.php/AAAI/article/view/29728.

12. AMUGONGO, L. M.; MASCHERONI, P.; BROOKS, S. G.; DOERING, S. Retrieval Augmented Generation for Large Language Models in Healthcare: A Systematic Review. Disponível em: https://www.preprints.org/frontend/manuscript/8f517fc19c2231693bf1460bb0d0b4a4/download_pub.

13. WANG, D.; LIANG, J.; YE, J.; LI, J.; LI, J.; ZHANG, Q.; HU, Q. Enhancement of the performance of large language models in diabetes education through retrieval-augmented generation: comparative study. Journal of Medical Internet Research, 2024. Disponível em: https://www.jmir.org/2024/1/e58041/.

14. LI, Y.; ZHAO, J.; LI, M.; DANG, Y.; YU, E.; LI, J. RefAI: a GPT-powered retrieval-augmented generative tool for biomedical literature recommendation and summarization. Journal of the American Medical Informatics Association, 2024. Disponível em: https://academic.oup.com/jamia/article-abstract/31/9/2030/7690757.

15. LIU, S.; MCCOY, A. B.; WRIGHT, A. Improving large language model applications in biomedicine with retrieval-augmented generation: a systematic review, meta-analysis, and clinical development. Journal of the American Medical Informatics Association, 2025. DOI: 10.1093/jamia/ocaf008.

16. AYTAR, A. Y.; KILIC, K.; KAYA, K. A Retrieval-Augmented Generation Framework for Academic Literature Navigation in Data Science. Disponível em: 10.48550/arXiv.2412.15404.

17. BREIMAN, L. Random Forests. *Machine Learning* , v. 45, n. 1, p. 5-32, 2001. DOI: 10.1023/A:1010933404324. Disponível em: https://doi.org/10.1023/A:1010933404324

18. LIAW, A.; WIENER, M. Classification and Regression by randomForest. *R News* , v. 2, n. 3, p. 18-22, 2002. Disponível em: https://www.r-project.org/doc/Rnews/Rnews_2002-3.pdf

19. DIETTERICH, T. G. Ensemble Methods in Machine Learning. *International Workshop on Multiple Classifier Systems* , p. 1-15, 2000. DOI: 10.1007/3-540-45014-9_1. Disponível em: https://arxiv.org/abs/1301.3781

20. MIKOLOV, T. et al. Efficient Estimation of Word Representations in Vector Space. *arXiv:1301.3781* , 2013. Disponível em: https://arxiv.org/abs/1301.3781

21. DEVLIN, J. et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL-HLT* , p. 4171-4186, 2019. DOI: 10.18653/v1/N19-1423. Disponível em: https://doi.org/10.18653/v1/N19-1423

22. ZHANG, Y. et al. Deep Learning Based Text Classification: A Comprehensive Review. *ACM Computing Surveys* , v. 54, n. 3, p. 1-40, 2020. DOI: 10.1145/3439726. Disponível em: https://doi.org/10.1145/3439726

23. JURAFSKY, D.; MARTIN, J. H. Speech and Language Processing. 3rd ed. *Pearson* , 2023. Disponível em: https://web.stanford.edu/~jurafsky/slp3/

24. RICCI, F. et al. Recommender Systems Handbook. 2nd ed. *Springer* , 2015. DOI: 10.1007/978-1-4899-7637-6. Disponível em: https://link.springer.com/book/10.1007/978-1-4899-7637-6

25. AGGARWAL, C. C. Machine Learning for Text. *Springer* , 2018. DOI: 10.1007/978-3-319-73531-3. Disponível em: https://link.springer.com/book/10.1007/978-3-319-73531-3

26. ZHANG, L.; WANG, Y.; LI, F. Duplicate Product Detection in Retail using FastText Embeddings and Cosine Similarity. In: Proceedings of the 12th International Conference on Data Mining, 2020.

27. SILVA, R.; OLIVEIRA, M.; SANTOS, D. Semantic Clustering of Retail Product Titles using SBERT and UMAP. In: Journal of Retail Data Science, 2023.