

GEORGE LUCAS LOPES DA SILVA GOMES

**VALIDAÇÃO DE PRESCRIÇÃO MÉDICA COM INTELIGÊNCIA ARTIFICIAL
GENERATIVA: ANÁLISE DE INTERAÇÕES MEDICAMENTOSAS BASEADA EM
BULA**

Trabalho de Conclusão de Curso apresentado
à banca avaliadora do Curso de Engenharia
de Computação, da Escola Superior de
Tecnologia, da Universidade do Estado do
Amazonas, como pré-requisito para obtenção
do título de Engenheiro de Computação.

Orientador(a): Prof. Dr. Raimundo Corrêa de Oliveira

Manaus – Março – 2025

Ficha Catalográfica

1. Estas instruções **não** devem ser entregues aos avaliadores do trabalho nas ocasiões dos TCCs 1 e 2. Portanto, comente a linha que importa este arquivo;
2. A ficha catalográfica deve ser elaborada após a defesa do TCC2, quando você concluir as correções sugeridas pela banca e validar a versão final com o/a professor(a) orientador(a);
3. Acessar <http://repositorioinstitucional.uea.edu.br:8080/ficha/ficha_catalografica.php> e preencher as informações requeridas para elaborar a ficha catalográfica;
4. Incluir o pdf gerado na subpasta *source* com o nome `ficha.pdf`;
5. Apagar todo o conteúdo deste arquivo e deixar apenas o comando a seguir:

```
\includepdf [pages=-]{./source/ficha.pdf}
```

Folha de Aprovação

1. Estas instruções **não** devem ser entregues aos avaliadores do trabalho nas ocasiões dos TCCs 1 e 2. Portanto, comente a linha que importa este arquivo;
2. Após a defesa do TCC2, efetue as correções indicadas pela banca;
3. Valide a versão final do trabalho com seu/sua orientador(a). Ele/Ela irá entregar uma carta de anuência ao coordenador da disciplina de TCC2 indicando estar de acordo com a versão final produzida;
4. O responsável pela disciplina de TCC2 irá lhe entregar digitalmente um documento PDF com as assinaturas e aprovação dos integrantes da banca;
5. O referido arquivo deve ser adicionado na pasta source com o nome `assinaturas.pdf`;
6. Todo o conteúdo deste arquivo deve ser retirado e substituído pelo seguinte comando:

```
\includepdf[pages=-]{./source/assinaturas.pdf}
```

Resumo

Deve constar de um parágrafo único cujo objetivo é prover uma versão condensada e concentrada do trabalho completo. De maneira geral, deve ser organizado em uma sequência que inclui: (1) o propósito geral do problema que está sendo pesquisado; (2) o design básico do trabalho realizado; (3) os principais resultados e evidências resultantes de sua análise, e (4) um breve sumário das suas interpretações e conclusões, evidenciando as contribuições da pesquisa. Para saber mais como escrever um bom resumo, sugere-se consultar a literatura sobre escrita científica, tais como: <<https://doi.org/10.4103%2F0019-5545.82558>>, <<https://plos.org/resource/how-to-write-a-great-abstract/>>, dentre outras.

Palavras-Chave: Palavra1; Palavra2; Palavra3.

Abstract

Follows the same instructions for its portuguese counterpart, except that it must not be translated with automatic tools. Put some effort on this part or hire a professional to help you with it.

Keywords: Keyword1; Keyword2; Keyword3.

Sumário

Lista de Tabelas	vii
Lista de Figuras	ix
Lista de Códigos	ix
1 Introdução	1
1.1 Justificativa	1
1.2 Objetivo Geral	2
1.3 Objetivos Específicos	2
1.4 Metodologia	2
1.4.1 Desenvolvimento do Sistema	2
1.5 Etapas do Projeto	3
1.5.1 Etapa 1: Criar o Crawler da Anvisa	3
1.5.2 Etapa 2: Upload e Armazenamento de Arquivos	4
1.5.3 Etapa 3: Embeddings das Bulas	4
1.5.4 Etapa 4: Cadastro de Usuários	5
1.5.5 Etapa 5: Upload de Arquivos de Usuários	5
1.5.6 Etapa 6: Embeddings dos Documentos do Usuário	5
1.5.7 Etapa 7: Análise de Interações Medicamentosas e Validação de Prescrição	6
1.6 Próximo Passo	6
2 Fundamentação Teórica	7

2.1	Inteligência Artificial	7
2.2	Inteligência Artificial Generativa	7
2.3	Modelos de Linguagem de Grande Escala (LLMs)	8
2.4	Recuperação Aumentada por Geração (RAG)	8
2.5	Embeddings e Representação de Texto	8
2.6	Bancos de Dados Vetoriais	9
2.7	Busca em Documentos: BM25 e KNN	9
2.7.1	BM25: Busca Baseada em Palavras-Chave	9
2.7.2	KNN: Busca por Similaridade Vetorial	10
3	Exemplo	12
3.1	Tabela com o Pacote Booktabs	12
3.2	Inserção de Figuras	12
3.3	Referências Bibliográficas no Padrão ABNT	13
4	Título do Quarto Capítulo	15
4.1	Algoritmos	15

Lista de Tabelas

3.1 Esta é uma tabela básica em L^AT_EX com o pacote booktabs. 12

Lista de Figuras

3.1	Um exemplo de figura JPG inserida no \LaTeX	13
3.2	Canal clássico cuja obtenção da capacidade erro-zero é não-trivial.	13

Lista de Códigos

Capítulo 1

Introdução

1.1 Justificativa

A prescrição de medicamentos é uma prática essencial na área da saúde, mas erros de prescrição podem levar a complicações médicas graves, incluindo interações medicamentosas prejudiciais, reações adversas e falhas terapêuticas. Médicos e farmacêuticos enfrentam desafios diários ao garantir que uma prescrição seja segura e eficaz para o paciente, considerando variáveis como idade, peso, condições clínicas, alergias e medicamentos em uso.

A tecnologia pode desempenhar um papel fundamental na redução desses erros, proporcionando ferramentas inteligentes para análise e validação de prescrições. A aplicação proposta utilizará inteligência artificial e processamento de linguagem natural para analisar bulas de medicamentos em conjunto com os dados do paciente, gerando alertas automáticos sobre possíveis incompatibilidades.

Ao desenvolver essa solução, buscamos contribuir para a segurança do paciente, reduzir a carga de trabalho dos profissionais de saúde e oferecer uma ferramenta de suporte à decisão baseada em evidências.

1.2 Objetivo Geral

Desenvolver uma aplicação baseada em inteligência artificial que auxilie médicos e farmacêuticos na validação de prescrições médicas, analisando bulas de medicamentos e dados clínicos do paciente para identificar possíveis interações medicamentosas e alertar sobre incompatibilidades.

1.3 Objetivos Específicos

- Criar um crawler para obter automaticamente bulas de medicamentos do site da Anvisa.
- Permitir upload e armazenamento de bulas, tanto as baixadas da Anvisa quanto as enviadas pelo usuário.
- Implementar um sistema de cadastro de pacientes, permitindo a inserção de informações médicas relevantes (idade, peso, histórico de doenças, alergias, medicamentos em uso, exames laboratoriais, etc.).
- Desenvolver um mecanismo de análise semântica das bulas utilizando Langchain.js e Elasticsearch para embeddings.
- Criar um mecanismo de análise automática de interações medicamentosas, validando prescrições médicas com base nos dados do paciente.
- Implementar alertas e recomendações baseadas em inteligência artificial, fornecendo insights para médicos e farmacêuticos no momento da prescrição.

1.4 Metodologia

1.4.1 Desenvolvimento do Sistema

O projeto será desenvolvido utilizando uma arquitetura modular, dividida nos seguintes componentes principais:

- **Backend (NestJS + Prisma + PostgreSQL)**

- CRUD para cadastro de pacientes (nome, idade, peso, histórico clínico, alergias, medicamentos em uso).
- CRUD para cadastro de medicamentos e prescrições.
- Integração com MinIO para armazenamento de documentos médicos.
- API para comunicação com o frontend e com o módulo de inteligência artificial.

- **Inteligência Artificial (Langchain.js + Elasticsearch)**

- Extração de informações de bulas de medicamentos.
- Indexação e busca semântica utilizando embeddings de IA.
- Geração de relatórios e validações por meio do modelo Claude Sonnet 3.5.
- Sugestões de ajustes nas prescrições médicas.

- **Frontend (AngularJS + MaterialUI/Bootstrap)**

- Tela de cadastro de paciente.
- Interface para upload de documentos e pesquisa de bulas.
- Visualização de alertas e recomendações personalizadas.

1.5 Etapas do Projeto

O desenvolvimento será dividido em sete etapas, cada uma com suas atividades e entregáveis.

1.5.1 Etapa 1: Criar o Crawler da Anvisa

Objetivo: Obter bulas de medicamentos automaticamente do site da Anvisa.

Tarefas:

- Identificar as URLs relevantes e a estrutura do site da Anvisa.
- Definir a tecnologia do crawler (Scrapy, BeautifulSoup ou Selenium).

- Implementar o script para buscar e baixar bulas automaticamente.
- Criar uma base de metadados para armazenar informações extraídas.
- Lidar com possíveis bloqueios (CAPTCHA, rate limits).

Previsão: 1-2 semanas

1.5.2 Etapa 2: Upload e Armazenamento de Arquivos

Objetivo: Criar um sistema para armazenar bulas baixadas e arquivos enviados pelo usuário.

Tarefas:

- Criar endpoints no backend (NestJS) para upload e consulta de arquivos.
- Configurar MinIO para armazenar documentos.
- Estruturar um banco de dados (PostgreSQL) para associar bulas aos usuários.
- Criar mecanismos para recuperação e listagem de arquivos.

Previsão: 1-2 semanas

1.5.3 Etapa 3: Embeddings das Bulas

Objetivo: Processar documentos para permitir consultas semânticas via IA.

Tarefas:

- Criar um pipeline de processamento utilizando Langchain.js e Elasticsearch.
- Indexar bulas automaticamente para permitir pesquisa inteligente.
- Criar endpoints para busca por similaridade.

Previsão: 1 semana

1.5.4 Etapa 4: Cadastro de Usuários

Objetivo: Criar um sistema para cadastrar médicos e permitir autenticação.

Tarefas:

- Criar a estrutura de usuários no PostgreSQL.
- Implementar autenticação JWT e gerenciamento de permissões.
- Criar interface de login e perfil no frontend.

Previsão: 1 semana

1.5.5 Etapa 5: Upload de Arquivos de Usuários

Objetivo: Permitir que médicos adicionem documentos médicos dos pacientes.

Tarefas:

- Criar endpoints no backend para upload e armazenamento de arquivos.
- Associar documentos a pacientes no banco de dados.
- Criar a interface de gerenciamento de arquivos no frontend.

Previsão: 1-2 semanas

1.5.6 Etapa 6: Embeddings dos Documentos do Usuário

Objetivo: Permitir consultas inteligentes nos documentos médicos do paciente.

Tarefas:

- Criar um pipeline de embeddings para os arquivos médicos.
- Indexar automaticamente os documentos ao serem enviados.
- Criar endpoints para busca semântica nos arquivos médicos.

Previsão: 1 semana

1.5.7 Etapa 7: Análise de Interações Medicamentosas e Validação de Prescrição

Objetivo: Criar alertas automáticos para identificar interações medicamentosas e validar prescrições.

Tarefas:

- Criar prompts específicos para o Claude Sonnet 3.5 gerar relatórios.
- Implementar endpoints no backend para validação de interações medicamentosas.
- Criar uma interface no frontend para exibir os alertas ao médico.

Previsão: 2 semanas

1.6 Próximo Passo

Com base nesse planejamento, o primeiro passo será a implementação do crawler para obter bulas do site da Anvisa.

Capítulo 2

Fundamentação Teórica

2.1 Inteligência Artificial

A **Inteligência Artificial** (IA) é um campo da ciência da computação que busca desenvolver sistemas capazes de simular a capacidade humana de raciocínio, aprendizado e tomada de decisões (RUSSELL; NORVIG, 2016). A IA é amplamente utilizada em diversas áreas, como reconhecimento de padrões, processamento de linguagem natural, automação de tarefas e suporte à decisão médica.

Dentre as técnicas de IA, destacam-se os algoritmos de aprendizado de máquina (*Machine Learning*) e aprendizado profundo (*Deep Learning*), que permitem aos sistemas aprenderem a partir de grandes volumes de dados e aprimorarem seu desempenho ao longo do tempo (GOODFELLOW; BENGIO; COURVILLE, 2016).

2.2 Inteligência Artificial Generativa

A **Inteligência Artificial Generativa** (*Generative AI*) refere-se a uma classe de algoritmos de IA capazes de criar novos conteúdos a partir de exemplos previamente aprendidos. Esses modelos são frequentemente baseados em redes neurais profundas, como *transformers*, *autoencoders* variacionais (VAE) e redes adversárias generativas (GANs) (BROWN et al., 2020).

Na área da saúde, modelos de IA generativa podem ser utilizados para gerar relatórios

médicos, sumarizar informações clínicas e até mesmo criar imagens médicas sintéticas para treinamento de modelos diagnósticos (SHEN; WU; SUK, 2021).

2.3 Modelos de Linguagem de Grande Escala (LLMs)

Os **Modelos de Linguagem de Grande Escala** (*Large Language Models* - LLMs) são modelos de IA treinados com grandes volumes de dados textuais para compreender e gerar linguagem natural. Exemplos populares incluem o GPT (Generative Pre-trained Transformer), BERT (Bidirectional Encoder Representations from Transformers) e Claude (DEVLIN et al., 2018).

Esses modelos são amplamente utilizados em aplicações como chatbots, geração de texto, tradução automática e suporte à tomada de decisão baseada em linguagem (BOMMASANI et al., 2021).

2.4 Recuperação Aumentada por Geração (RAG)

A **Recuperação Aumentada por Geração** (*Retrieval-Augmented Generation* - RAG) combina técnicas de recuperação de informações com modelos de linguagem generativa para aprimorar a precisão e relevância das respostas (LEWIS, 2020). O RAG primeiro busca informações relevantes em uma base de dados (por exemplo, documentos médicos ou bulas de medicamentos) e, em seguida, utiliza um LLM para gerar respostas com base nesses dados.

Essa abordagem é especialmente útil em aplicações médicas, onde a confiabilidade da informação é fundamental. Em vez de confiar apenas no conhecimento interno do modelo de IA, o RAG permite que a IA consulte fontes externas, garantindo maior precisão e contextualização nas respostas (MIALON; SCIALOM et al., 2023).

2.5 Embeddings e Representação de Texto

Os **embeddings** são representações vetoriais de palavras, frases ou documentos, que permitem ao computador entender relações semânticas entre diferentes textos. Modelos como Word2Vec,

GloVe, FastText e Transformers são amplamente utilizados para gerar embeddings (MIKOLOV et al., 2013).

Na área médica, embeddings podem ser aplicados para indexação de documentos clínicos, agrupamento de termos semelhantes e recuperação semântica de informações em prontuários eletrônicos (PENNINGTON; SOCHER; MANNING, 2014).

2.6 Bancos de Dados Vetoriais

Os **bancos de dados vetoriais** são sistemas especializados no armazenamento e recuperação de embeddings. Diferentemente dos bancos de dados relacionais tradicionais, que operam com tabelas e consultas SQL, os bancos de dados vetoriais utilizam técnicas de indexação aproximada para permitir buscas eficientes em espaços de alta dimensão (JOHNSON; DOUZE; JÉGOU, 2019).

Exemplos de bancos de dados vetoriais populares incluem:

- **FAISS (Facebook AI Similarity Search)** – Otimizado para grandes volumes de embeddings e busca rápida (JOHNSON; DOUZE; JÉGOU, 2019).
- **Pinecone** – Solução gerenciada na nuvem para indexação vetorial escalável.
- **Milvus** – Um banco de dados vetorial de código aberto amplamente utilizado em aplicações de IA.

Esses bancos de dados são essenciais para sistemas que utilizam RAG, pois permitem recuperar documentos relevantes antes da geração da resposta (JIANG et al., 2021).

2.7 Busca em Documentos: BM25 e KNN

2.7.1 BM25: Busca Baseada em Palavras-Chave

O **BM25** (*Best Matching 25*) é um modelo estatístico amplamente utilizado para recuperação de documentos baseando-se na frequência dos termos presentes em um texto (ROBERT-

SON; ZARAGOZA, 2009). Ele pertence à família dos modelos de recuperação de informações baseados na frequência de palavras, sendo uma evolução do **TF-IDF** (Term Frequency - Inverse Document Frequency).

O BM25 utiliza dois fatores principais para calcular a relevância de um documento em relação a uma consulta:

- **Frequência do termo** (*Term Frequency - TF*): A quantidade de vezes que um termo aparece no documento.
- **Frequência inversa no conjunto de documentos** (*Inverse Document Frequency - IDF*): Mede a importância do termo em relação ao conjunto de documentos, penalizando termos muito comuns.

Dessa forma, BM25 é eficiente para encontrar documentos onde os termos da consulta aparecem com alta relevância, sendo amplamente utilizado em motores de busca, como Elasticsearch e Apache Solr (MANNING; RAGHAVAN; SCHÜTZE, 2008).

2.7.2 KNN: Busca por Similaridade Vetorial

O **k-Nearest Neighbors (KNN)** é um algoritmo de aprendizado de máquina não supervisionado usado para busca de similaridade entre documentos (FIX; JR, 1951). Ele funciona identificando os **k vizinhos mais próximos** de um ponto de consulta dentro de um espaço vetorial.

No contexto de recuperação de informações, o KNN pode ser aplicado da seguinte forma:

- Cada documento é convertido em um vetor numérico usando **embeddings**.
- Quando uma nova consulta é feita, seu embedding é comparado com os embeddings dos documentos armazenados.
- Os **k documentos mais próximos** da consulta são retornados como os mais relevantes.

A busca vetorial baseada em KNN é essencial para sistemas que utilizam **busca semântica**, pois permite encontrar textos com significados semelhantes, mesmo que não compartilhem as mesmas palavras-chave (VASWANI et al., 2017).

Capítulo 3

Exemplo

3.1 Tabela com o Pacote Booktabs

Um exemplo de tabela com o pacote booktabs pode ser visto na Tabela 3.1. A explicação das tabelas sempre vem em cima e esse padrão deve ser respeitado. A numeração é automática e a inserção no índice também. Legal né? (BENNETT; SHOR, 1998)

Basta quebrar uma linha para criar um novo parágrafo. Neste parágrafo vou contar que tabelas no \LaTeX dão um pouco de trabalho, mas nada que com paciência não se resolva. Veja os links com dicas que coloquei nos comentários do arquivo `index.tex`.

Tabela 3.1: Esta é uma tabela básica em \LaTeX com o pacote booktabs.

Parte 1	Parte 2	Parte 3	Parte 4
0,415	1,365	1,98	2,05
1,36	45,5	7,98	3,01
2,36	1,35	0,15	5,32

3.2 Inserção de Figuras

Você pode inserir figuras JPG no \LaTeX ! Veja o caso da Figura 3.1. Se você quiser outras configurações e dicas, veja o seguinte endereço: <<http://en.wikibooks.org/wiki/LaTeX/Floats>,

_Figures_and_Captions>.

A explicação da figura sempre vem embaixo da mesma. Isto aqui é um novo parágrafo apenas para ilustrar a idéia geral de como escrever.

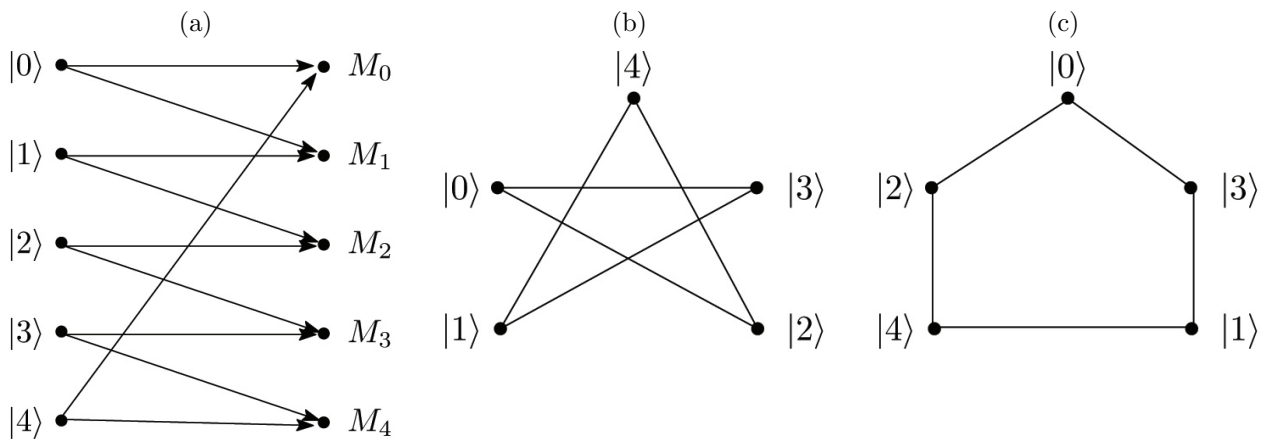
Figura 3.1: Um exemplo de figura JPG inserida no \LaTeX .



Elaborado pelo autor.

Pode inserir várias figuras lado a lado também. Um exemplo está reproduzido a seguir.

Figura 3.2: Canal clássico cuja obtenção da capacidade erro-zero é não-trivial.



Elaborado por Bacon

3.3 Referências Bibliográficas no Padrão ABNT

Para gerar versões corretas do \BibTeX das suas referências, recomenda-se usar o DOI no caso de artigos e o ISBN no caso de livros. Em posse dessas informações, consulte os seguintes links:

1. `<https://www.bibtex.com/c/doi-to-bibtex-converter/>`
2. `<https://www.bibtex.com/c/isbn-to-bibtex-converter/>`
3. `<http://doi-to-bibtex-converter.herokuapp.com/>`
4. `<https://www.doi2bib.org/>`

Para saber mais sobre os tipos de referências `BIBTEX` e seus significados, consultar a documentação oficial em `<https://www.bibtex.com/e/entry-types/>`.

Este template está integrado com o pacote `abnt2cite` que produz as referências no padrão ABNT NBR 6023.

Capítulo 4

Título do Quarto Capítulo

Vivamus ultricies tincidunt lacus ut pharetra. Sed fringilla hendrerit tempus. Suspendisse potenti. Cras hendrerit tortor ac est condimentum pellentesque. Morbi pretium lectus nec sapien laoreet eu malesuada diam adipiscing. Aliquam nisl ipsum, fermentum ut aliquam nec, varius sit amet nisi. Pellentesque interdum cursus malesuada. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nullam malesuada bibendum tortor, ut bibendum lorem varius eu. In eros orci, volutpat ut facilisis sit amet, commodo quis nulla.

Sed lectus metus, mollis nec vulputate id, imperdiet eget urna. Nam ut dolor at metus venenatis suscipit et in ligula. In hac habitasse platea dictumst. Mauris scelerisque dolor sed nisl mattis accumsan. Aliquam vulputate placerat feugiat. Pellentesque faucibus neque mi. Etiam porttitor varius tempus. Mauris varius porttitor posuere. Pellentesque iaculis imperdiet lobortis. Sed vulputate purus nec felis rutrum molestie.

4.1 Algoritmos

Nunc at fringilla dui. Pellentesque id tortor eu libero auctor rhoncus id vel velit. Duis auctor laoreet turpis, sed commodo tellus sollicitudin sit amet. Phasellus quis purus consectetur turpis hendrerit pretium eget in velit. Cras dignissim est vel mi malesuada a imperdiet velit condimentum. Vivamus ultrices diam non urna aliquet hendrerit. Sed lobortis, mauris quis egestas ullamcorper, nunc nulla auctor nulla, eu rutrum velit velit in nulla. Etiam lectus augue,

pellentesque et porta at, pharetra id lectus. Duis eleifend eleifend mauris, nec mollis mauris vehicula nec. Nam sed ipsum ut massa lacinia vestibulum. Duis vitae sapien a lectus aliquam luctus eget sit amet nunc. Etiam a ipsum auctor tortor condimentum consectetur. Aliquam vestibulum libero sit amet nulla auctor aliquet. Sed laoreet imperdiet tellus non vulputate. Vivamus tristique ipsum vel metus venenatis in laoreet tortor hendrerit. Suspendisse potenti. Aenean tincidunt molestie libero sit amet porttitor. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos.

Cras nec quam mi, ut mattis ante. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed fringilla auctor dictum. Nam hendrerit sapien sed massa consequat rutrum. Nullam congue, augue sed commodo malesuada, lectus nulla mollis magna, eget semper risus nisl eget elit. Duis vitae hendrerit massa. In a odio nunc, sit amet mollis dolor. In accumsan suscipit dui, a vestibulum diam condimentum ullamcorper. Etiam ut quam arcu, ac tristique ante. Vestibulum imperdiet elit non ante tristique accumsan. Donec vulputate fringilla tempor. Proin porttitor nisi nisi. Fusce vel ullamcorper orci. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Vivamus ultricies tincidunt lacus ut pharetra. Sed fringilla hendrerit tempus. Suspendisse potenti. Cras hendrerit tortor ac est condimentum pellentesque. Morbi pretium lectus nec sapien laoreet eu malesuada diam adipiscing. Aliquam nisl ipsum, fermentum ut aliquam nec, varius sit amet nisi. Pellentesque interdum cursus malesuada. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nullam malesuada bibendum tortor, ut bibendum lorem varius eu. In eros orci, volutpat ut facilisis sit amet, commodo quis nulla.

Referências Bibliográficas

BENNETT, C. H.; SHOR, P. W. Quantum information theory. *IEEE Transactions on Information Theory*, v. 44, n. 6, p. 2724–2742, 1998.

BOMMASANI, R. et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

BROWN, T. et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, v. 33, p. 1877–1901, 2020.

DEVLIN, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

FIX, E.; JR, J. L. H. Discriminatory analysis: Nonparametric discrimination: Consistency properties. *Project 21-49-004, Report Number 4, USAF School of Aviation Medicine*, 1951.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016.

JIANG, M. et al. Efficient similarity search for embeddings of high-dimensional data. *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, 2021.

JOHNSON, J.; DOUZE, M.; JÉGOU, H. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019.

LEWIS, P. e. a. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*, 2020.

MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. *Introduction to Information Retrieval*. [S.l.]: Cambridge University Press, 2008.

MIALON, G.; SCIALOM, T. et al. Augmented language models: a survey. *arXiv preprint arXiv:2302.07842*, 2023.

MIKOLOV, T. et al. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, v. 26, 2013.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, p. 1532–1543, 2014.

ROBERTSON, S.; ZARAGOZA, H. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, v. 3, n. 4, p. 333–389, 2009.

RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. [S.l.]: Pearson, 2016.

SHEN, D.; WU, G.; SUK, H.-I. Artificial intelligence in radiology: Current applications and future directions. *Nature Biomedical Engineering*, v. 5, p. 509–525, 2021.

VASWANI, A. et al. Attention is all you need. *Advances in Neural Information Processing Systems*, v. 30, 2017.