UNIVERSIDADE ESTADUAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
CURSO DE ENGENHARIA DA COMPUTAÇÃO

JOSÉ CHRYSTIAN LIMA PACHECO

REDUZINDO ALUCINAÇÕES E APRIMORANDO A QUALIDADE DE RESPOSTAS EM SISTEMAS ALIMENTADOS POR LARGE LANGUAGE MODELS ATRAVÉS DE RE-RANKING DE DOCUMENTOS PARA APLICAÇÕES EM DOMÍNIOS ESPECÍFICOS

SÃO LUÍS
2025

UNIVERSIDADE ESTADUAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
CURSO DE ENGENHARIA DA COMPUTAÇÃO

JOSÉ CHRYSTIAN LIMA PACHECO

REDUZINDO ALUCINAÇÕES E APRIMORANDO A QUALIDADE DE RESPOSTAS EM SISTEMAS ALIMENTADOS POR LARGE LANGUAGE MODELS ATRAVÉS DE RE-RANKING DE DOCUMENTOS PARA APLICAÇÕES EM DOMÍNIOS ESPECÍFICOS

Trabalho de Conclusão de Curso apresentado para obtenção do grau de Bacharel em Engenharia da Computação, pela Universidade Estadual do Maranhão.

Orientador: Prof. Dr. Luis Carlos Costa Fonseca

SÃO LUÍS
2025

Pacheco, Jose Chrystian Lima.

Reduzindo alucinações e aprimorando a qualidade de respostas em sistemas alimentados por large language models através de re-ranking de documentos para aplicações em domínios específicos./ José Chrystian Lima Pacheco . São Luís- MA, 2025.

46p.

Trabalho de Conclusão de Curso (Curso de Engenharia de Computação) Universidade Estadual do Maranhão - UEMA, São Luís - MA, 2025.

Orientador: Prof. Dr. Luis Carlos Costa Fonseca.

1. Alucinações. 2. Modelos de Linguagem de Grande Escala. 3. Retrieval Augmented Generation. 4. Re-ranqueamento de Documentos . 5. Processo de Linguagem Natural. I.Título.

CDU:004.41

Elaborado por Luciana de Araújo - CRB 13/445

JOSÉ CHRYSTIAN LIMA PACHECO

Reduzindo Alucinações e Aprimorando a Qualidade de Respostas em Sistemas Alimentados Por Large Language Models Através de Re-Ranking de Documentos Para Aplicações Em Domínios Específicos

Trabalho de Conclusão de Curso apresentado para obtenção do grau de Bacharel em Engenharia da Computação, pela Universidade Estadual do Maranhão.

SÃO LUÍS, DIA de janeiro de 2025:

Documento assinado digitalmente
gov.br
LUIS CARLOS COSTA FONSECA
Data: 01/10/2025 18:37:39-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Luis Carlos Costa Fonseca
Orientador - UEMA

Prof. Pedro Brandão Neto
Examinador Interno - UEMA

Documento assinado digitalmente
gov.br
REINALDO DE JESUS DA SILVA
Data: 03/10/2025 13:41:48-0300
Verifique em https://validar.iti.gov.br

Prof. Reinaldo de Jesus da Silva
Examinador Interno - UEMA

SÃO LUÍS
2025

AGRADECIMENTOS

A concretização deste trabalho de conclusão de curso é resultado de um esforço coletivo, e expresso aqui minha profunda gratidão a todos que contribuíram para esta conquista. Aos colegas de curso, que compartilharam desafios e aprendizados ao longo desta jornada, meu sincero reconhecimento. Em especial, dirijo meus agradecimentos à minha mãe, minha madrinha e minha avó, pilares fundamentais em minha vida. O apoio incondicional, o amor e a dedicação destas mulheres extraordinárias foram essenciais para que eu chegasse até aqui. A elas, minha eterna gratidão..

RESUMO

Este trabalho aborda o problema das alucinações em Modelos de Linguagem de Grande Escala (LLMs) por meio da melhoria de sistemas de *Retrieval Augmented Generation* (RAG). Especificamente, propõe-se um modelo de re-ranqueamento de documentos, baseado na arquitetura BERT, para refinar os resultados recuperados em um sistema RAG, priorizando documentos mais relevantes e, consequentemente, mitigando a geração de informações incorretas ou infundadas (alucinações) pelos LLMs. Para treinar e avaliar o modelo, foi criado um *dataset* inovador a partir de dez Trabalhos de Conclusão de Curso (TCCs) da Universidade Estadual do Maranhão (UEMA), utilizando técnicas de Processamento de Linguagem Natural (PLN) para extração de texto e geração automática de perguntas com diferentes níveis de relevância (scores 1, 3 e 5). O modelo de re-ranqueamento BERTimbau foi treinado para classificar pares de pergunta e documento de acordo com sua relevância. Os resultados experimentais demonstram que o modelo alcança uma acurácia de 92% na classificação de relevância e um MRR de 0.7367, indicando uma melhora significativa na ordenação dos documentos recuperados em comparação com uma abordagem sem re-ranqueamento (MRR de 0.4140). A análise qualitativa ilustra a capacidade do modelo de discernir entre diferentes níveis de relação semântica entre perguntas e documentos. Este trabalho contribui para o avanço do estado da arte em RAG, fornecendo um método eficaz para reduzir alucinações em LLMs e melhorar a confiabilidade das informações geradas em aplicações no domínio da língua portuguesa, especificamente no contexto de TCCs da UEMA.

Palavras-chave: Alucinações, Modelos de Linguagem de Grande Escala, *Retrieval Augmented Generation*, Re-ranqueamento de Documentos, BERT, Processamento de Linguagem Natural.

ABSTRACT

This work addresses the problem of hallucinations in Large Language Models (LLMs) by improving Retrieval Augmented Generation (RAG) systems. Specifically, a document re-ranking model, based on the BERT architecture, is proposed to refine the results retrieved in a RAG system, prioritizing more relevant documents and, consequently, mitigating the generation of incorrect or unfounded information (hallucinations) by LLMs. To train and evaluate the model, an innovative dataset was created from 10 Undergraduate Theses (TCCs) from the State University of Maranhão (UEMA), using Natural Language Processing (NLP) techniques for text extraction and automatic generation of questions with different levels of relevance (scores 1, 3, and 5). The re-ranking model, based on *neuralmind/bert-base-portuguese-cased*, was trained to classify question-document pairs according to their relevance. The experimental results demonstrate that the model achieves an accuracy of 92% in relevance classification and an MRR of 0.7367, indicating a significant improvement in the ranking of retrieved documents compared to an approach without re-ranking (MRR of 0.4140). The qualitative analysis illustrates the model's ability to discern between different levels of semantic relationship between questions and documents. This work contributes to advancing the state of the art in RAG, providing an effective method to reduce hallucinations in LLMs and improve the reliability of the information generated in applications in the Portuguese language domain, specifically in the context of UEMA's TCCs.

Keywords: Hallucinations, Large Language Models, Retrieval Augmented Generation, Document Re-ranking, BERT, Natural Language Processing, UEMA TCCs.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

LLM	*Large Language Model*
RAG	*Retrieval Augmented Generation*
IR	*Information Retrieval*
ISAR	*Information Storage and Retrieval*
BERT	*Bidirectional Encoder Representations from Transformers*
TF-IDF	*Term Frequency-Inverse Document Frequency*
BM25	*Best Match 25*
SBERT	*Sentence-BERT*
NLI	*Natural Language Inference*
KDE	*Kernel Density Estimation*
DPR	*Dense Passage Retrieval*
CON	*chain-of-note*
IA	Inteligência Artificial
PLN	Processamento de Linguagem Natural
MRR	*Mean Reciprocal Rank*
MAP	*Mean Average Precision*
NDCG	*Normalized Discounted Cumulative Gain*

SUMÁRIO

1 INTRODUÇÃO

1.1 Contextualização do Problema

Os Modelos de Linguagem de Grande Escala (LLMs), como GPT-3, Gemini e Llama 3, têm demonstrado avanços impressionantes em tarefas de compreensão e geração de linguagem natural. No entanto, esses modelos apresentam limitações significativas, como a incapacidade de revisar ou expandir seu conhecimento após o treinamento e a alarmante tendência a gerar respostas imprecisas ou inventadas, conhecidas como "alucinações"(JI et al., 2023). Em aplicações críticas, como na área da saúde, finanças ou jornalismo, a geração de informações incorretas pode ter consequências graves, comprometendo a confiabilidade e a segurança dos sistemas baseados em LLMs. (JI et al., 2023) classifica as alucinações em LLMs em duas categorias principais: alucinações de factualidade, onde as informações geradas não correspondem a fatos verificáveis, e alucinações de fidelidade, que envolvem inconsistências com as instruções ou o contexto fornecido pelo usuário.

Estudos recentes, como o de Lewis et al. (2020), propõem o uso de memórias paramétricas e não-paramétricas para endereçar essas questões, permitindo que os modelos acessem informações externas, o que possibilita a atualização e a verificação de seu conhecimento em tempo real. Adicionalmente, (JI et al., 2023) destaca que as alucinações podem ser causadas por fatores relacionados aos dados de treinamento, arquitetura do modelo e estratégias de inferência, indicando a necessidade de abordagens mais robustas para mitigar esses problemas. Uma dessas abordagens promissoras é a *Retrieval Augmented Generation* (RAG) (LEWIS et al., 2020), que combina a capacidade generativa dos LLMs com a recuperação de informações relevantes de uma base de conhecimento externa.

1.1.1 Justificativa

Para mitigar as limitações dos LLMs, técnicas de *Retrieval Augmented Generation* (RAG) têm sido exploradas (LEWIS et al., 2020). O RAG combina um modelo de geração de linguagem com um sistema de recuperação de informações, aprimorando a confiabilidade e a relevância das respostas geradas pelos LLMs. Este método tem

se mostrado crucial em aplicativos que utilizam LLMs, particularmente em domínios específicos que requerem informações atualizadas ou proprietárias (GAO et al., 2023). Embora avanços significativos tenham sido feitos no desenvolvimento de LLMs mais poderosos, como o Llama 3, que demonstra desempenho comparável a modelos líderes como o GPT-4 em diversas tarefas (DUBEY et al., 2024), desafios persistem na capacidade desses modelos de acessar e utilizar informações atualizadas ou específicas de domínio.

Mesmo com melhorias na escala e na qualidade dos dados de treinamento, a necessidade de sistemas eficazes de recuperação de informação permanece crítica para garantir respostas precisas e relevantes. Conforme discutido em (JI et al., 2023), as alucinações em LLMs não são apenas um problema técnico, mas também levantam preocupações sobre a confiabilidade desses modelos em aplicações do mundo real. O artigo enfatiza que, apesar dos avanços, os LLMs continuam propensos a gerar conteúdo que parece plausível, mas que não é suportado por fatos ou pelo contexto fornecido. Embora técnicas de recuperação de informações, como BM25 e busca híbrida, sejam fundamentais e ajam como bons *baselines*, elas têm suas limitações, muitas vezes falhando em avaliar a relevância dos documentos recuperados em relação às consultas específicas do usuário (ROBERTSON, 2009).

Neste contexto, este trabalho se destaca por propor uma abordagem para mitigar o problema das alucinações em LLMs: a introdução de um sistema de re-ranqueamento de documentos como uma camada de pós-processamento em sistemas RAG. Esta camada adicional irá avaliar e selecionar os documentos mais pertinentes à consulta do usuário, visando melhorar a precisão da recuperação de informações e, consequentemente, a qualidade das respostas dos LLMs. Além disso, este trabalho contribui com a criação de um dataset original e específico para a língua portuguesa, baseado em Trabalhos de Conclusão de Curso (TCCs) da Universidade Estadual do Maranhão (UEMA). A utilização deste dataset, em conjunto com a técnica de re-ranqueamento, representa uma contribuição significativa e original para a área de pesquisa em RAG, especialmente no contexto da língua portuguesa e para o domínio específico de TCCs.

A proposta contribui significativamente para a pesquisa em inteligência artificial e processamento de linguagem natural, abordando um desafio crítico na interação entre LLMs e sistemas de recuperação de informações. Este estudo tem o potencial de estabe-

lecer novos padrões de precisão e confiabilidade para aplicações práticas de LLMs em diversos setores.

1.1.2 Objetivo Geral

Desenvolver um modelo de re-ranqueamento de documentos, baseado na arquitetura BERT, para ser integrado a sistemas RAG, a fim de melhorar a qualidade e reduzir a alucinação em LLMs aplicados ao domínio de TCCs da UEMA.

1.1.3 Objetivos Específicos

- Desenvolver um algoritmo de re-ranqueamento, baseado em BERT, que atua como uma camada de pós-processamento entre o sistema de recuperação de informação e o modelo generativo.
- Criar um dataset estruturado, a partir de TCCs da UEMA, para treinar e avaliar a eficácia do modelo de re-ranqueamento proposto.
- Realizar e avaliar experimentos comparando a abordagem proposta (com re-ranqueamento) com uma abordagem sem re-ranqueamento (usando BM25 como *baseline*).

1.2 Estrutura do Trabalho

No Capítulo 2, apresentamos a fundamentação teórica, incluindo uma revisão dos LLMs, recuperação de informação, RAG, re-ranqueamento de documentos e métricas de avaliação.

O Capítulo 3 detalha a metodologia empregada, incluindo o desenvolvimento do algoritmo, criação dos datasets, treinamento do modelo e procedimentos de avaliação.

No Capítulo 4, apresentamos os experimentos realizados e discutimos os resultados obtidos. O Capítulo 5 conclui o trabalho, destacando as principais contribuições e sugerindo trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Large Language Models (LLMs) e suas Limitações

A evolução dos modelos de grande porte, do inglês *Large Language Models* (LLMs), representa um salto significativo nas aplicações de Inteligência Artificial (IA), evidenciando-se em tarefas como tradução, sumarização e respostas a perguntas (BROWN et al., 2020). Contudo, os LLMs enfrentam limitações ao lidar com questões que exigem dados atualizados ou externos ao seu conjunto de treinamento. Além disso, possuem dificuldade em expandir ou revisar sua memória, não fornecem informações diretas sobre suas previsões e geram erros conhecidos como “alucinações” (JI et al., 2023).

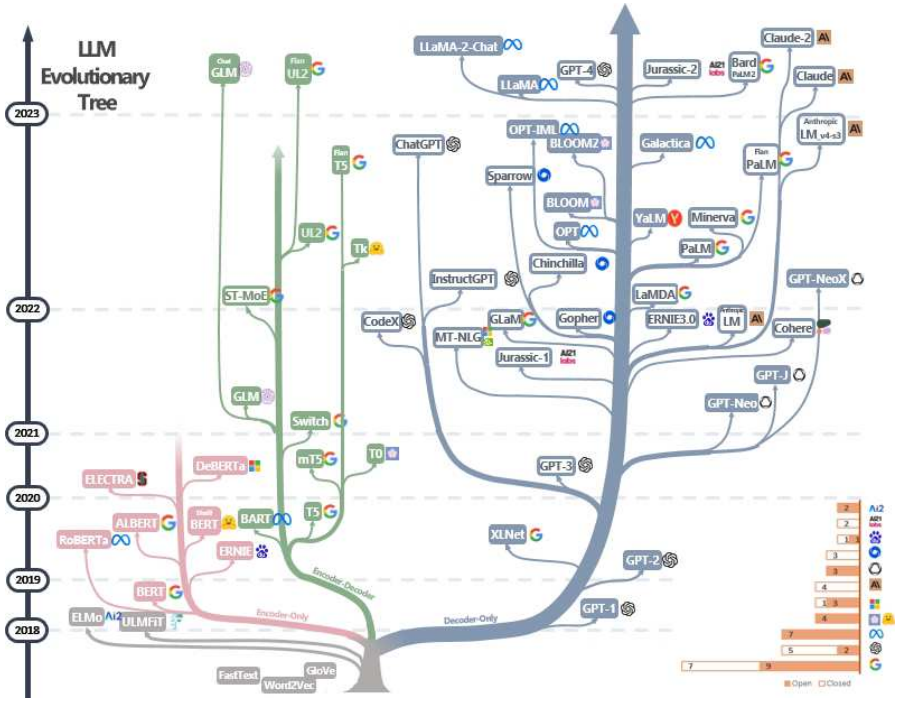


Figura 1 – LLMs evolutionary tree. (Adapted from Yang et al., 2023)

Modelos poderosos como o GPT-3 (175 bilhões de parâmetros) ainda enfrentam dificuldades em tarefas de compreensão de linguagem e inferência, especialmente quando os dados de treinamento não abrangem adequadamente o domínio em questão (BROWN et al., 2020). Já o Llama 3, treinado em um corpus de aproximadamente 15 trilhões de

tokens multilíngues, busca mitigar essas limitações com melhorias nos dados, escala e gerenciamento da complexidade (DUBEY et al., 2024).

2.1.1 Alucinação

Uma das principais preocupações com os LLMs é sua tendência à alucinação (JI et al., 2023). Alucinação, nesse contexto, refere-se à geração de informações falsas ou não suportadas pelo contexto fornecido (JI et al., 2023). Esse problema é particularmente crítico em domínios específicos onde a precisão e a confiabilidade das informações são primordiais (JI et al., 2023). Por exemplo, em áreas como saúde, direito ou finanças, informações falsas geradas por um LLM podem ter consequências graves (JI et al., 2023).

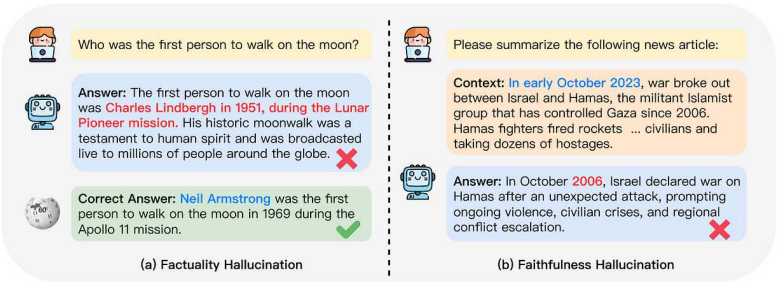


Figure 1: An intuitive example of LLM hallucination.

Figura 2 – Exemplo de alucinação em um LLM[1]

A alucinação em LLMs surge de vários fatores, incluindo:

Dados de Treinamento Ruidosos: Datasets de treinamento massivos frequentemente contêm informações incorretas ou inconsistentes (JI et al., 2023). Essa 'infidelidade' de dados encoraja alucinações, pois o modelo aprende e reproduz as imprecisões presentes nos dados (JI et al., 2023). A coleta heurística de dados, onde frases ou tabelas reais são selecionadas e emparelhadas como fonte e alvo, pode resultar em um alvo de referência contendo informações não suportadas pela fonte (JI et al., 2023). Por exemplo, na construção do WIKIBIO, um conjunto de dados para gerar notas biográficas com base em infoboxes da Wikipedia, 62% das primeiras frases contêm informações adicionais não declaradas na infobox correspondente, levando à alucinação (JI et al., 2023).

[1] <https://profile.caotouchan.tech/the-mirage-of-knowledge-hallucination-in-llms-2b5b79bda467>

Compreensão Semântica Limitada: Os LLMs podem ter dificuldade em entender completamente o significado subjacente do texto, especialmente quando se trata de nuances complexas ou informações implícitas (JI et al., 2023). Essa limitação pode levar à geração de respostas plausíveis do ponto de vista linguístico, mas factualmente incorretas (JI et al., 2023).

Viés Inerente aos Dados: LLMs são suscetíveis a replicar vieses presentes nos dados de treinamento (JI et al., 2023). Isso pode levar a resultados discriminatórios ou imprecisos, perpetuando e amplificando os vieses existentes na sociedade (JI et al., 2023).

A tolerância à alucinação varia entre as tarefas de NLG (JI et al., 2023). Em tarefas como sumarização e *data-to-text*, a tolerância à alucinação é muito baixa, pois a fidelidade à entrada é essencial. Por outro lado, em sistemas de diálogo, a tolerância é relativamente maior, pois características como engajamento do usuário também são desejáveis, especialmente em sistemas de diálogo abertos (JI et al., 2023).

2.1.2 Outras Limitações

Além da alucinação, os LLMs podem apresentar outras limitações que afetam sua confiabilidade e desempenho (JI et al., 2023):

Dificuldade em Lidar com Informações de Cauda Longa: LLMs podem ter dificuldade em acessar e processar informações raras ou especializadas que não estão bem representadas nos dados de treinamento (JI et al., 2023). Essa limitação é particularmente problemática em domínios específicos onde o conhecimento especializado é essencial (JI et al., 2023).

Sensibilidade a *Prompts* e Formulação: A forma como uma pergunta ou instrução é formulada pode influenciar significativamente a resposta de um LLM (JI et al., 2023). Pequenas variações na formulação podem levar a resultados inconsistentes, destacando a sensibilidade dos LLMs à entrada específica (JI et al., 2023).

Falta de Transparência e Interpretabilidade: A tomada de decisão de LLMs é frequentemente opaca, tornando difícil entender por que um modelo produz uma resposta específica (JI et al., 2023). Essa falta de transparência pode ser um obstáculo em áreas onde a explicabilidade e a auditabilidade são cruciais (JI et al., 2023).

2.1.3 Mitigação de Limitações

Pesquisas em andamento estão explorando várias estratégias para mitigar as limitações dos LLMs, incluindo:

Métodos Relacionados a Dados: A construção de conjuntos de dados de treinamento mais confiáveis, por meio da curadoria manual (JI et al., 2023), revisão de dados existentes e aumento de dados (JI et al., 2023), visa reduzir a alucinação resultante de dados ruidosos.

Métodos de Modelagem e Inferência: Modificações na arquitetura (JI et al., 2023), métodos de treinamento aprimorados (JI et al., 2023) e técnicas de pós-processamento (JI et al., 2023) visam melhorar a fidelidade, o raciocínio e o controle sobre a saída do modelo.

Técnicas de Grounding de Conhecimento: Integrar conhecimento externo de fontes confiáveis, como bancos de dados de conhecimento (JI et al., 2023), pode ajudar os LLMs a gerar respostas mais precisas e factualmente corretas.

Aprendizado com Feedback Humano: Incorporar feedback humano no processo de treinamento pode ajudar os LLMs a aprender com seus erros, melhorando sua capacidade de gerar respostas factualmente corretas.

Abordagens como essas são cruciais para impulsionar o avanço dos LLMs e torná-los mais confiáveis e eficazes em aplicações de domínio específico.

2.1.4 Recuperação de Informação

Sistemas de Recuperação de Informação (*Information Retrieval*; IR) são onipresentes em nosso dia a dia, desde mecanismos de busca na web até catálogos de bibliotecas e índices de livros de receitas (SOERGEL, 2004). A recuperação de informação, também conhecida como Armazenamento e Recuperação de Informação (*Information Storage and Retrieval*; ISAR) ou organização e recuperação de informação, trata da arte e da ciência de recuperar, de uma coleção de itens, um subconjunto que satisfaça a necessidade de informação do usuário (SOERGEL, 2004).

O objetivo primordial da recuperação de informação é identificar e recuperar informações relevantes à consulta do usuário (HAMBARDE; PROENçA, 2023). Dado

que múltiplos registros podem ser relevantes, os resultados são frequentemente ordenados de acordo com sua pontuação de relevância em relação à consulta (HAMBARDE; PROENçA, 2023). Segundo (SOERGEL, 2004), o processo de recuperação de informação envolve várias etapas interdependentes:

1. **Indexação de documentos:** A menos que o sistema opere diretamente sobre o texto do documento, ele se prepara para a recuperação indexando os documentos, o que resulta em representações de documentos. A indexação automática inicia com a extração bruta de características, como extrair todas as palavras de um texto. Esta etapa é seguida por refinamentos, como a remoção de palavras irrelevantes como "e", "isto" e "de", aplicação de stemização ("canalizações" para "canaliza", por exemplo), contagem (utilizando apenas as palavras mais frequentes) e mapeamento de conceitos usando um tesauro (tubo e cano mapeiam para o mesmo conceito, por exemplo) (SOERGEL, 2004).

2. **Formulação de consultas:** O sistema formula consultas, resultando em representações de consultas. Um sistema de recuperação de informação pode exibir uma hierarquia de assuntos para navegação e identificação de bons descritores ou pode formular uma série de perguntas ao usuário e, a partir das respostas, construir uma consulta. É crucial que o sistema sugira sinônimos e termos mais restritos e mais amplos de seu tesauro para auxiliar o usuário, visto que, sem ajuda, os usuários podem não considerar todos os recursos relevantes (SOERGEL, 2004).

3. **Correspondência de representações:** O sistema compara as representações de documentos e consultas. Essa correspondência utiliza as características especificadas na consulta para prever a relevância do documento (SOERGEL, 2004).

4. **Exibição e seleção:** Finalmente, o sistema exibe os documentos encontrados, classificados por uma pontuação de relevância esperada, e o usuário seleciona os itens relevantes. O processo de busca geralmente passa por múltiplas iterações, nas quais o conhecimento das características que distinguem documentos relevantes de irrelevantes é usado para aprimorar a consulta ou a indexação (*feedback* de relevância) (SOERGEL, 2004).

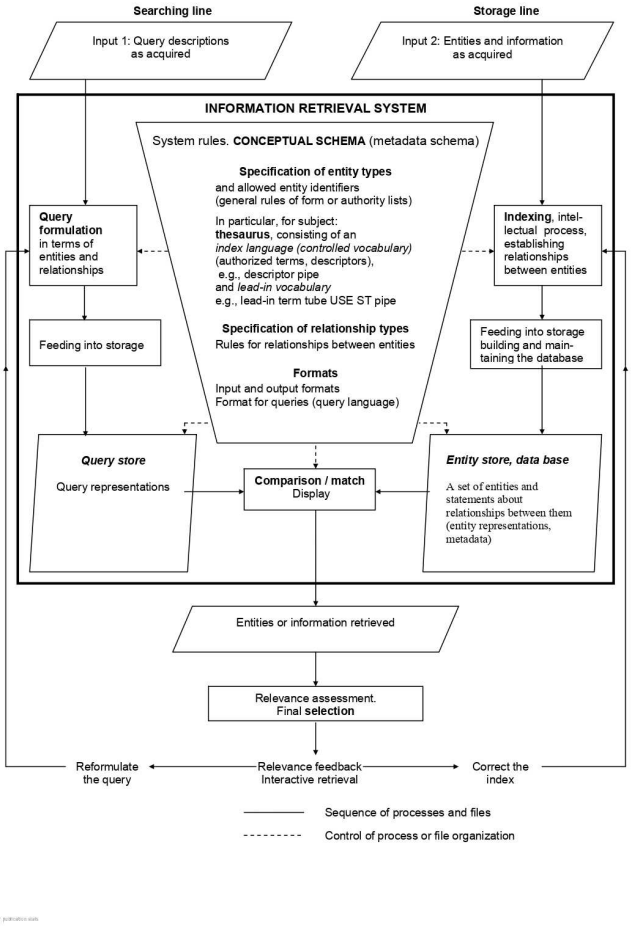


Figura 3 – The general structure of the IRS (SOERGEL, 2004).

A representação de documentos é crucial para o processo de recuperação de informação. Um documento pode ser representado como um conjunto de declarações, incluindo informações como autor, título, resumo, texto livre, descritores, função e URL. Se uma entidade (como um documento ou um arquivo de dados) é buscada como fonte de dados/informação, os dados sobre a entidade são usados como metadados (dados que descrevem dados).

A classificação de documentos em classes (mutuamente exclusivas) de uma classificação também é conhecida como categorização de texto (SOERGEL, 2004). Na ausência de uma classificação adequada, o sistema pode gerar uma por meio de agrupa-

mento, agrupando documentos que estão próximos uns dos outros (ou seja, documentos que compartilham muitas características) (SOERGEL, 2004).

É fundamental considerar a importância dos termos para o usuário, a frequência dos termos nos documentos e a raridade dos termos na coleção ao calcular as pontuações de relevância (SOERGEL, 2004). Além disso, o sistema deve ser capaz de lidar com a expansão de sinônimos, como em "se a consulta pedir cano, também encontra tubos", e expansão hierárquica ou pesquisa inclusiva ("também encontra capilar", por exemplo) para aumentar a revocação.

Uma técnica importante para melhorar o desempenho dos sistemas de IR é a expansão de documentos, que consiste em expandir a representação de cada documento incluindo termos relacionados adicionais (HAMBARDE; PROENçA, 2023). Ao fazer isso, o sistema de IR poderá corresponder melhor a consulta com os documentos relevantes. Diversos estudos têm se concentrado na relação entre a estrutura do *corpus*, os modelos de linguagem e a recuperação de informação *ad-hoc*, propondo novas abordagens usando técnicas de agrupamento. Outras pesquisas se concentraram em expandir a representação de cada documento usando termos relacionados, *WordNet* (um grande banco de dados lexical do inglês) e até mesmo coleções externas para melhorar o desempenho da recuperação, especialmente para textos curtos (HAMBARDE; PROENçA, 2023).

A escolha de um modelo de recuperação de informação adequado depende das necessidades específicas da aplicação, do domínio e dos recursos disponíveis.

2.1.4.1 Metodos Modernos

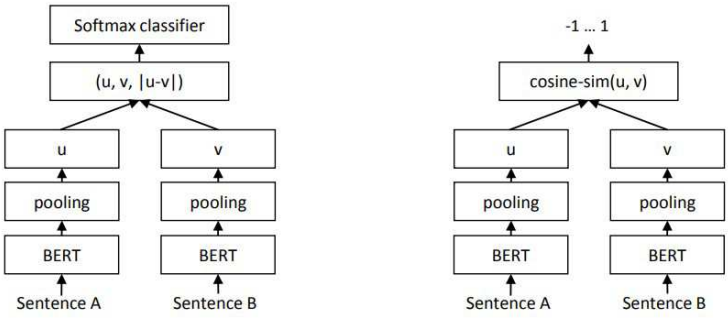
O advento do *Bidirectional Encoder Representations from Transformers* (BERT), um modelo de linguagem pré-treinado apresentado por Devlin et al. (2018), marcou um ponto de inflexão significativo na área de recuperação de informação. Sua capacidade de capturar relações bidirecionais entre palavras em um texto, proporcionou uma compreensão contextualizada da linguagem natural sem precedentes, impactando diretamente a forma como os sistemas de recuperação de informação compreendem e representam documentos e consultas (DEVLIN et al., 2018).

A principal inovação do BERT reside na sua arquitetura bidirecional, que permite

que o modelo leve em consideração o contexto tanto à esquerda quanto à direita de uma palavra ao gerar sua representação (DEVLIN et al., 2018). Essa característica contrasta com os modelos de linguagem anteriores, que processavam o texto de forma unidirecional, seja da esquerda para a direita ou da direita para a esquerda (DEVLIN et al., 2018). A capacidade do BERT de capturar dependências bidirecionais de longo alcance entre palavras se mostrou crucial para uma série de tarefas de processamento de linguagem natural, incluindo a recuperação de informação (DEVLIN et al., 2018).

Nesse contexto, o BERT pode ser utilizado para gerar representações vetoriais de documentos e consultas que capturam nuances semânticas e contextuais da linguagem natural (DEVLIN et al., 2018). Essas representações, por sua vez, podem ser utilizadas para calcular a similaridade entre documentos e consultas de forma mais precisa, levando a resultados de recuperação mais relevantes (DEVLIN et al., 2018). Diversos estudos têm demonstrado a efetividade do BERT na melhoria da performance de sistemas de recuperação de informação, especialmente em cenários de recuperação ad-hoc (DEVLIN et al., 2018).

Figura 4 – Sentence Bert Architecture



Fonte: Reimers e Gurevych (2019)

2.2 Retrieval Augmented Generation (RAG)

Retrieval Augmented Generation (RAG) surge como uma solução promissora para superar as limitações dos LLMs em tarefas que demandam conhecimento abrangente. Apesar da capacidade de armazenar informações factuais, os LLMs ainda enfrentam dificuldades em acessar e manipular dados de forma precisa, especialmente em domínios

específicos (LEWIS et al., 2020). A arquitetura RAG visa contornar esses desafios combinando a memória paramétrica de um modelo pré-treinado com a memória não paramétrica de uma fonte de conhecimento externa, como a Wikipédia (LEWIS et al., 2020). Essa integração permite o acesso a informações relevantes de forma mais eficaz, complementando o conhecimento intrínseco do LLM e reduzindo a ocorrência de alucinações.

2.2.1 Arquitetura RAG

O framework RAG se baseia na interação de três componentes principais:

1. **Retriever:** Responsável por buscar os documentos mais relevantes em um corpus de conhecimento externo, como um índice denso da Wikipédia, para responder a uma pergunta ou input (LEWIS et al., 2020).

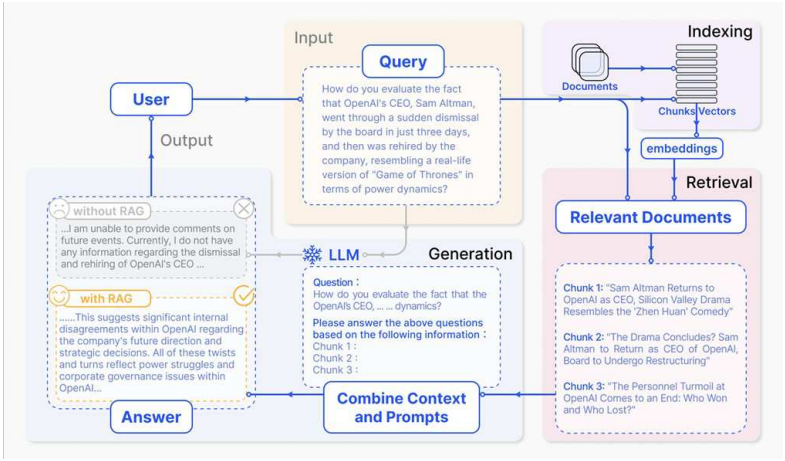
    - ***Dense Passage Retrieval* (DPR):** Utiliza uma arquitetura bi-encoder para calcular a similaridade entre documentos e perguntas, codificando ambos em vetores densos por meio de modelos de linguagem pré-treinados, como o BERT. O produto escalar desses vetores determina a similaridade, e os documentos com maior pontuação são recuperados (KARPUKHIN et al., 2020).
    - ***Chunking*:** A otimização do índice de documentos é essencial para a eficácia da recuperação. O chunking é uma técnica que divide os documentos em unidades menores, equilibrando a integridade semântica com o comprimento do contexto para indexação e pesquisa (LEWIS et al., 2020; LIU et al., 2024; YU et al., 2023).

2. **Gerador:** Geralmente um modelo de linguagem pré-treinado do tipo *sequence-to-sequence*, utiliza os documentos recuperados pelo retriever para gerar a resposta final. A geração é condicionada aos documentos fornecidos, garantindo que a resposta seja fundamentada em informações relevantes e atualizadas (LEWIS et al., 2020; RAFFEL et al., 2019).

3. **Mecanismo de Aumento:** Define como a informação recuperada é incorporada ao processo de geração, influenciando a capacidade do RAG de lidar com informações ruidosas e irrelevantes (LEWIS et al., 2020).

    - **Condicionamento nos Documentos:** Existem duas formulações principais de RAG: uma condiciona a geração nos mesmos documentos recuperados para toda a sequência, enquanto a outra permite o uso de diferentes documentos por token (LEWIS et al., 2020).
    - **Geração de Notas:** O *chain-of-note* (CON) gera notas sequenciais para cada documento recuperado, avaliando sua relevância e identificando as informações mais confiáveis, filtrando conteúdo irrelevante e aumentando a precisão das respostas (YU et al., 2023).
    - **Raciocínio:** O framework de autorraciocínio proposto por Xia et al. (2024) utiliza raciocínios gerados pelo próprio LLM para aprimorar a confiabilidade e rastreabilidade do RAG. O framework analisa a relevância dos documentos, seleciona e cita evidências e gera uma análise concisa com base nas trajetórias de raciocínio, resultando em respostas mais precisas e transparentes (XIA et al., 2024).

Figura 5 – Arquitetura do RAG



Fonte: Gao et al. (2023)

2.2.2 Benefícios do RAG

Redução de Alucinações: O acesso a informações relevantes de fontes externas auxilia na mitigação das alucinações, garantindo que as respostas sejam baseadas em evidências (LEWIS et al., 2020).

Atualização Contínua de Conhecimento: A memória não paramétrica do RAG permite fácil atualização com novas informações, mantendo o modelo alinhado com os dados mais recentes (LEWIS et al., 2020).

Integração de Conhecimento Específico de Domínio: A incorporação de fontes de conhecimento personalizadas para domínios específicos torna o RAG ideal para aplicações que demandam conhecimento aprofundado em áreas como medicina ou direito (LEWIS et al., 2020).

2.2.3 Desafios do RAG

Robustez ao Ruído: Documentos irrelevantes podem comprometer o desempenho do RAG, tornando crucial o desenvolvimento de mecanismos para filtrar o ruído e assegurar o uso de informações relevantes (LEWIS et al., 2020).

Compreensão da Relevância: A capacidade do modelo de entender a relevância dos documentos recuperados é fundamental para gerar respostas precisas. Técnicas como CHAIN-OF-NOTE e self-reasoning framework visam aprimorar essa capacidade (YU et al., 2023; XIA et al., 2024).

Gerenciamento do Contexto: A gestão do volume de informação recuperada é crucial para evitar sobrecarga. O problema "Lost in the Middle"evidencia a dificuldade dos modelos em utilizar informações localizadas no meio de longos contextos (LIU et al., 2024). Estratégias de reranking, seleção e compressão de contexto podem ser implementadas para otimizar o processamento (GAO et al., 2023).

2.2.4 Considerações Adicionais sobre o RAG

A implementação do RAG envolve decisões importantes que impactam seu desempenho:

Fontes de Informação: A escolha da fonte de dados, como a Wikipédia, corpus de domínio específico ou até mesmo conteúdo gerado por LLMs, influencia a qualidade e a cobertura do conhecimento disponível (LEWIS et al., 2020).

Granularidade da Recuperação: A decisão sobre a granularidade dos elementos recuperados, como frases, parágrafos ou documentos inteiros, impacta a precisão e a eficiência do processo (GAO et al., 2023).

Técnicas de Aumento: A seleção de métodos de aumento adequados, como o condicionamento em documentos, a geração de notas ou o self-reasoning, é fundamental para integrar efetivamente a informação recuperada ao processo de geração (YU et al., 2023; XIA et al., 2024).

Avaliação do RAG: Métricas como relevância do contexto, fidelidade, relevância da resposta, robustez ao ruído e capacidade de rejeição negativa são essenciais para avaliar o desempenho do RAG e direcionar futuras pesquisas (GAO et al., 2023).

O desenvolvimento e aprimoramento do RAG dependem de pesquisas contínuas para superar os desafios e explorar novas abordagens, como a integração de dados estruturados e semi-estruturados, o desenvolvimento de mecanismos de raciocínio mais sofisticados e a otimização do gerenciamento de contexto em cenários complexos (GAO et al., 2023).

Compreender o funcionamento do RAG, seus benefícios, desafios e as decisões envolvidas em sua implementação é essencial para o desenvolvimento de sistemas robustos e eficazes, capazes de reduzir alucinações, aprimorar a qualidade das respostas e impulsionar o avanço dos LLMs em diversas aplicações.

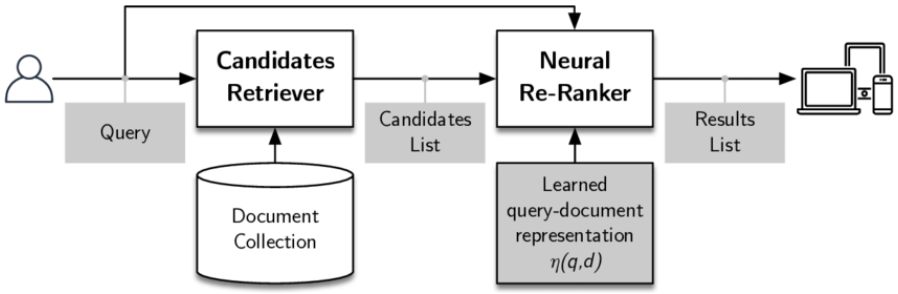
2.3 Re-Ranqueamento de Documentos

O processo de IR busca identificar documentos relevantes em resposta a uma consulta específica. Tradicionalmente, métodos como *Term Frequency-Inverse Document Frequency* (TF-IDF) ou *Best Match 25* (BM25) são utilizados para uma primeira seleção de documentos. No entanto, esses métodos, apesar de eficientes, podem apresentar limitações, especialmente em grandes conjuntos de dados e em cenários que exigem uma compreensão mais profunda da semântica da consulta(KARPUKHIN et al., 2020). O

re-ranqueamento de documentos surge como uma etapa crucial para refinar os resultados da recuperação, aumentando a precisão e a relevância dos documentos apresentados ao usuário.

2.3.1 Definição

O re-ranqueamento é uma técnica que, após a recuperação inicial de documentos, aplica um modelo mais sofisticado para ordenar novamente os documentos, priorizando aqueles que são mais relevantes para a consulta. Este processo é particularmente importante em domínios abertos, onde a complexidade das consultas e a diversidade dos documentos exigem uma análise mais refinada do que a oferecida pelos métodos de recuperação inicial(KARPUKHIN et al., 2020).



Fonte: Gao et al. (2023)

Figura 6 – Re-ranking pipeline architecture for interaction-focused neural IR systems.(Tonellotto, 2022)

Em conjuntos de dados de grande escala, como o MS MARCO, a recuperação inicial pode retornar documentos que, apesar de conterem termos da consulta, não são realmente relevantes (KARPUKHIN et al., 2020). O re-ranqueamento, neste contexto, visa aumentar a precisão dos resultados, garantindo que os documentos mais relevantes sejam apresentados no topo da lista. O MS MARCO, por exemplo, é um conjunto de dados de grande escala que contém mais de um milhão de consultas e milhões de passagens, sendo um ambiente ideal para testar e aprimorar técnicas de re-ranqueamento (KARPUKHIN et al., 2020).

A recuperação de passagens relevantes para responder a perguntas em domínios abertos é outro desafio que demanda o uso de técnicas de re-ranqueamento. Em vez de analisar documentos inteiros, o re-ranqueamento pode identificar as passagens específicas

que contêm a resposta para uma determinada pergunta. Isso é crucial para sistemas de perguntas e respostas, onde a precisão e a velocidade são essenciais (KARPUKHIN et al., 2020).

2.3.2 Métodos de Re-ranqueamento

As abordagens baseadas em aprendizado profundo têm demonstrado grande eficácia no re-ranqueamento de documentos. Estes modelos, ao contrário dos métodos tradicionais, são capazes de capturar nuances semânticas e relações complexas entre a consulta e os documentos, resultando em resultados mais precisos(KARPUKHIN et al., 2020).

Ao tratar a consulta e a passagem como um par de sentenças, o BERT pode calcular a relevância de uma passagem para uma consulta específica (KARPUKHIN et al., 2020). O modelo BERT, quando ajustado para a tarefa de re-ranqueamento, atinge resultados de ponta em conjuntos de dados como o MS MARCO e o TREC-CAR (KARPUKHIN et al., 2020). Ele consegue isso ao analisar a consulta como "sentença A" e a passagem como "sentença B", calculando a probabilidade de relevância da passagem com base na sua representação contextualizada.

O Sentence-BERT (SBERT) é uma modificação do BERT que produz *embeddings* de frases semanticamente significativos. Esses embeddings podem ser comparados usando a similaridade de cossenos, permitindo que o sistema identifique passagens que são semanticamente semelhantes à consulta (REIMERS; GUREVYCH, 2019). O SBERT supera outros métodos de *embeddings* de sentenças em diversas tarefas de similaridade textual, e ainda consegue reduzir significativamente o tempo de processamento para encontrar o par de sentenças mais similar. O SBERT foi ajustado com dados de *Natural Language Inference* (NLI), criando representações de sentenças que superam outros métodos de *embeddings* de sentenças (REIMERS; GUREVYCH, 2019).

O desempenho do SBERT em tarefas de pontuação de pares de sentenças pode ser melhorado através do aumento de dados (THAKUR et al., 2021). Técnicas de aumento de dados, como o uso de sinônimos ou amostras geradas por modelos como o BM25 e Kernel Density Estimation (KDE), podem criar conjuntos de dados sintéticos que complementam os dados de treinamento existentes. Estes métodos adicionam pares

de frases com similaridade variável aos dados de treinamento, resultando em melhor generalização do modelo (THAKUR et al., 2021). O aumento de dados com BM25 cria conjuntos de dados sintéticos com distribuição de similaridade mais próxima do conjunto de dados original, resultando em melhor desempenho do SBERT (THAKUR et al., 2021).

Embora *embeddings* densas de baixa dimensionalidade sejam úteis para reduzir o tempo de processamento, elas podem apresentar desafios em índices de grande escala. Estudos teóricos e empíricos mostram que o desempenho de *embeddings* densas diminui mais rapidamente do que representações esparsas à medida que o tamanho do índice aumenta. Isso se deve ao aumento de falsos positivos, ou seja, documentos irrelevantes são classificados como relevantes (REIMERS; GUREVYCH, 2021). Este fenômeno é mais pronunciado quanto menor a dimensionalidade dos *embeddings*, pois há maior sobreposição entre os vetores, levando à recuperação de documentos incorretos (REIMERS; GUREVYCH, 2021).

3 METODOLOGIA

3.1 Tipo de Pesquisa

Este trabalho se caracteriza como uma pesquisa experimental e exploratória. É experimental pois envolve a manipulação de variáveis, criação e treinamento de um modelo de re-ranqueamento e a medição dos seus efeitos. É exploratória pois visa investigar um problema relativamente novo na área de recuperação de informação em língua portuguesa, utilizando uma base de dados específica (TCCs da UEMA), abrindo caminho para investigações futuras.

A pesquisa adota uma abordagem quantitativa, pois utiliza métricas objetivas (acurácia, MRR, MAP, NDCG, *Recall*, Precision) para avaliar o desempenho do modelo de re-ranqueamento.

3.2 Desenvolvimento do Algoritmo de Re-Ranqueamento

O algoritmo de re-ranqueamento é implementado como um modelo de classificação de sentenças pré-treinado baseado no modelo BERTimbau[1], modelo baseado em BERT e treinado em português brasileiro.

O algoritmo foi desenvolvido em Python, utilizando as bibliotecas `transformers` para o modelo BERT e tokenização, `torch` para o treinamento do modelo, `sklearn` para divisão do dataset e métricas de avaliação, `json` para carregar o dataset, `numpy` para operações numéricas, `unicodedata` e `re` para pré-processamento.

Inicialmente, o modelo recebe como entrada um par pergunta-documento e retorna uma avaliação de relevância. Internamente, o tokenizador do BERT converte o par em uma sequência de tokens, adicionando tokens especiais `[CLS]` (início da sequência) e `[SEP]` (separador entre pergunta e documento). O modelo BERT processa essa sequência e gera um vetor de embedding para o token `[CLS]`, que representa a relação semântica entre a pergunta e o documento. Esse vetor é então passado para uma camada de classificação linear que produz um score para cada uma das classes de relevância (1, 3 e 5).

[1] <https://huggingface.co/neuralmind/bert-base-portuguese-cased>

O re-ranqueamento é feito com base no score de probabilidade retornado pelo modelo para cada classe. Para cada pergunta, os documentos são ranqueados em ordem decrescente de acordo com a probabilidade da classe de maior score (5).

A escolha do BERT se justifica por seu desempenho estado-da-arte em diversas tarefas de Processamento de Linguagem Natural (PLN), incluindo classificação de sentenças. O modelo BERTimbau foi escolhido por ser pré-treinado em português, o que é crucial para o bom desempenho na base de TCCs da UEMA.

3.3 Criação dos Datasets

Descrição dos dados originais:

Fonte dos dados: Os dados originais são provenientes de uma base de 10 Trabalhos de Conclusão de Curso (TCCs) da Universidade Estadual do Maranhão (UEMA), baixados em formato PDF. Embora o código permita a expansão, o experimento foi limitado a 10 documentos devido a restrições de processamento e tempo.

Formato dos dados: Os dados originais estão em formato PDF.

Quantidade de dados: Foram utilizados 10 arquivos PDF.

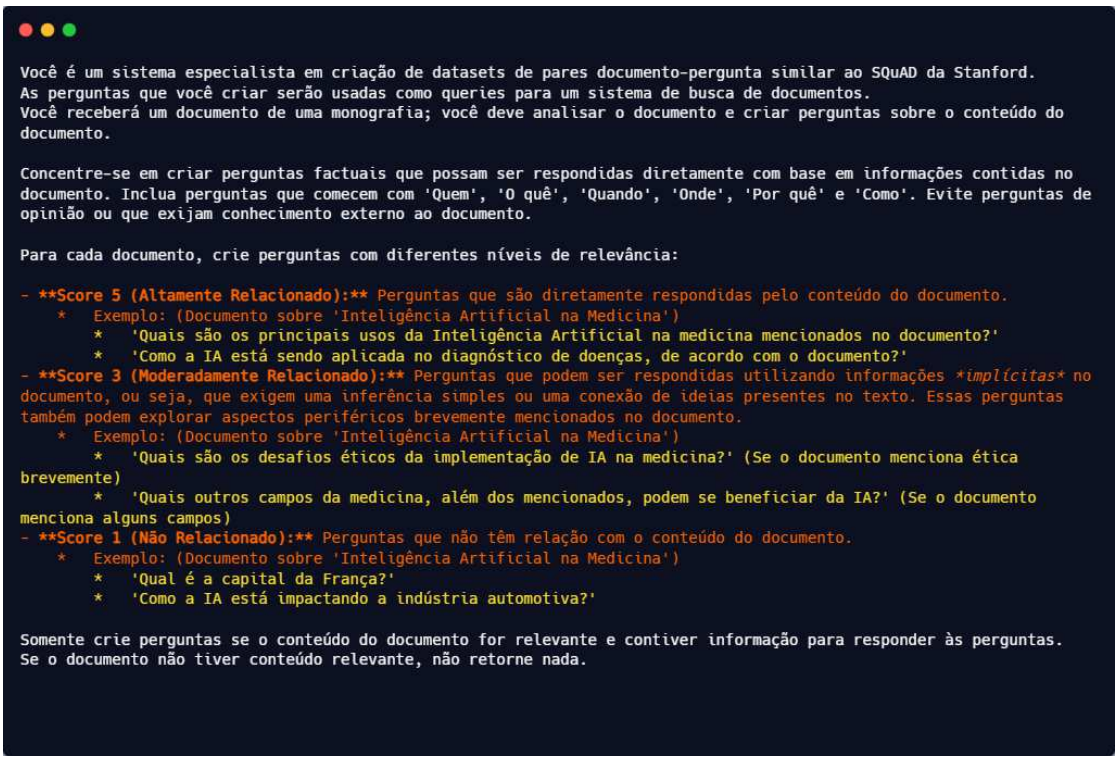
Características relevantes dos dados: O conteúdo textual dos TCCs, incluindo título, resumo, introdução, desenvolvimento, conclusão e referências bibliográficas, é a informação relevante para a criação das perguntas.

Código de criação do dataset:

Explicação do código: O código utiliza a biblioteca `pymupdf4llm` para extrair o texto dos PDFs e convertê-los para o formato Markdown. Em seguida, a biblioteca `langchain` é utilizada para dividir o texto em chunks (pedaços) com tamanho de 1024 tokens e sobreposição de 0 tokens, utilizando o `MarkdownTextSplitter`. Para cada chunk, uma cadeia de prompts (`qa_chain`) é executada, utilizando modelos de linguagem como o LLaMA 3, GPT-4o e Gemini 1.5 Pro para gerar perguntas com diferentes níveis de relevância (scores 1, 3 e 5). As perguntas geradas, juntamente com o chunk correspondente e o score atribuído, são armazenadas em um arquivo JSONL.

Detalhes sobre a transformação dos dados: A conversão para Markdown e a

Figura 7 – Prompts utilizado para geração das perguntas.



```
Você é um sistema especialista em criação de datasets de pares documento-pergunta similar ao SQuAD da Stanford.
As perguntas que você criar serão usadas como queries para um sistema de busca de documentos.
Você receberá um documento de uma monografia; você deve analisar o documento e criar perguntas sobre o conteúdo do
documento.

Concentre-se em criar perguntas factuais que possam ser respondidas diretamente com base em informações contidas no
documento. Inclua perguntas que comecem com 'Quem', 'O quê', 'Quando', 'Onde', 'Por quê' e 'Como'. Evite perguntas de
opinião ou que exijam conhecimento externo ao documento.

Para cada documento, crie perguntas com diferentes níveis de relevância:

- **Score 5 (Altamente Relacionado):** Perguntas que são diretamente respondidas pelo conteúdo do documento.
    *   Exemplo: (Documento sobre 'Inteligência Artificial na Medicina')
        *   'Quais são os principais usos da Inteligência Artificial na medicina mencionados no documento?'
        *   'Como a IA está sendo aplicada no diagnóstico de doenças, de acordo com o documento?'
- **Score 3 (Moderadamente Relacionado):** Perguntas que podem ser respondidas utilizando informações *implícitas* no
documento, ou seja, que exigem uma inferência simples ou uma conexão de ideias presentes no texto. Essas perguntas
também podem explorar aspectos periféricos brevemente mencionados no documento.
    *   Exemplo: (Documento sobre 'Inteligência Artificial na Medicina')
        *   'Quais são os desafios éticos da implementação de IA na medicina?' (Se o documento menciona ética
brevemente)
        *   'Quais outros campos da medicina, além dos mencionados, podem se beneficiar da IA?' (Se o documento
menciona alguns campos)
- **Score 1 (Não Relacionado):** Perguntas que não têm relação com o conteúdo do documento.
    *   Exemplo: (Documento sobre 'Inteligência Artificial na Medicina')
        *   'Qual é a capital da França?'
        *   'Como a IA está impactando a indústria automotiva?'

Somente crie perguntas se o conteúdo do documento for relevante e contiver informação para responder às perguntas.
Se o documento não tiver conteúdo relevante, não retorne nada.
```

Fonte: Elaborado pelo autor

divisão em chunks são essenciais para o processamento pelos modelos de linguagem. O prompt instrui os modelos a gerarem perguntas factuais, baseadas no conteúdo do documento, com diferentes níveis de relevância, simulando um cenário de busca de informações.

Justificativa das transformações: A divisão em chunks é necessária devido à limitação de contexto dos modelos de linguagem. A geração de perguntas com diferentes níveis de relevância permite treinar um modelo de re-ranqueamento mais robusto e capaz de distinguir entre documentos altamente relevantes, moderadamente relevantes e irrelevantes.

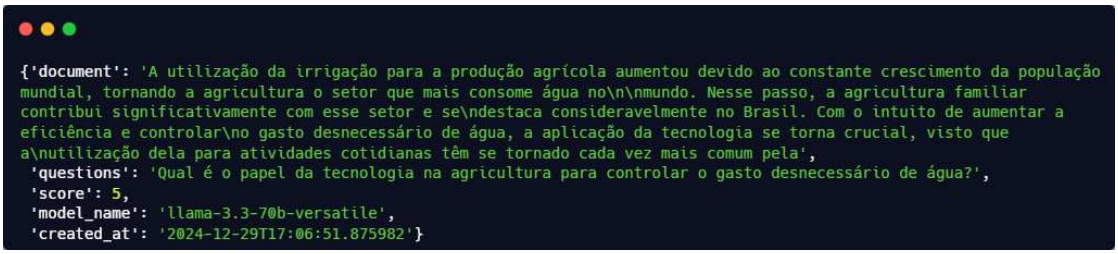
Formato do dataset final: O dataset final está no formato JSONL, onde cada linha representa um objeto JSON contendo o chunk do documento (`document`), a pergunta gerada (`questions`), o score de relevância (`score`), o nome do modelo que gerou a pergunta (`model_name`) e a data de criação (`created_at`).

Quantidade de dados no dataset final: O dataset final possui 4323 exemplos, após a divisão em treino e teste, o conjunto de treinamento possui 3458 exemplos e o de teste 865.

Divisão do dataset: O dataset foi dividido em conjuntos de treinamento (80%) e teste (20%), de forma aleatória, utilizando a função `train_test_split` da biblioteca `sklearn`, com `random_state=42` para garantir a reprodutibilidade.

Exemplos de dados no dataset final:

Figura 8 – Amostra do dataset



Fonte: Elaborado pelo autor

3.4 Treinamento do Modelo

Explicação do código: O código define a classe `ReRankingDataset` para carregar e formatar os dados para o modelo BERT. Ele utiliza o tokenizador do BERT para converter os pares (pergunta, documento) em sequências de tokens, adicionando os tokens especiais e aplicando padding. O modelo `BertForSequenceClassification` é carregado com o modelo pré-treinado BERTimbau e configurado para classificação com 3 classes (scores 1, 3 e 5). O otimizador `AdamW` e o scheduler `get_linear_schedule_with_warmup` são utilizados para o treinamento. O treinamento é realizado por 10 épocas, com lotes (batches) de tamanho 16.

Detalhes sobre o processo de treinamento: O código implementa as funções `train_epoch` e `eval_model` para treinar e avaliar o modelo, respectivamente. Em cada época, o modelo processa os lotes de dados, calcula a função de perda (loss), atualiza os pesos do modelo através do backpropagation e calcula a acurácia.

Arquitetura do modelo: O modelo é baseado na arquitetura BERT, que consiste em múltiplas camadas de atenção (transformers). O modelo utilizado é o BERTimbau, que

possui 12 camadas, 768 dimensões ocultas e 12 cabeças de atenção. Uma camada linear de classificação é adicionada ao final do modelo para produzir os scores de relevância.

Hiperparâmetros: Os principais hiperparâmetros utilizados foram:

- Taxa de aprendizado (`lr`): 2e-5
- Número de épocas (`epochs`): 10
- Tamanho do lote (`batch_size`): 16
- Tamanho máximo da sequência (`max_len`): 256

Função de perda (loss function): Foi utilizada a função de perda de entropia cruzada categórica (`CrossEntropyLoss`), que é padrão para problemas de classificação multiclasse.

Otimizador: Foi utilizado o otimizador `AdamW`, que é uma variante do Adam com correção de *weight decay*.

Critérios de parada: O treinamento foi realizado por um número fixo de 10 épocas. O modelo com melhor acurácia no conjunto de validação foi salvo.

[2]

3.4.1 Procedimentos de Avaliação

A avaliação do desempenho do modelo proposto será realizada através de um conjunto abrangente de métricas, amplamente utilizadas na literatura de recuperação de informação. A seguir, descreve-se cada uma dessas métricas, juntamente com a justificativa para sua escolha.

Métricas de Avaliação

Acurácia: A acurácia mede a porcentagem de classificações corretas, isto é, a proporção de pares (pergunta, documento) onde o score predito pelo modelo coincide com o score real. Formalmente, a acurácia é calculada como a razão entre o número de predições corretas e o número total de predições. Embora forneça uma visão geral do

[2] Códigos disponíveis em:<https://github.com/ChrystianGreen2/reranking>

desempenho, a acurácia pode ser enganosa em cenários com classes desbalanceadas, o que justifica a inclusão de métricas mais específicas.

Relatório de Classificação (Precision, Recall, F1-score): O relatório de classificação fornece uma análise mais granular do desempenho do modelo, desmembrando-o por classe. Ele é composto pelas seguintes métricas:

Precision: Avalia a proporção de predições positivas que foram de fato corretas. É calculada pela fórmula:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

onde TP (True Positives) representa o número de instâncias positivas corretamente classificadas e FP (False Positives) representa o número de instâncias negativas incorretamente classificadas como positivas.

Recall: Avalia a proporção de instâncias positivas que foram recuperadas corretamente. É calculada pela fórmula:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

onde FN (False Negatives) representa o número de instâncias positivas incorretamente classificadas como negativas.

F1-score: Representa a média harmônica entre Precision e Recall, fornecendo uma medida balanceada entre as duas. É calculada pela fórmula:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

MRR (Mean Reciprocal Rank): O MRR calcula a média dos inversos dos ranks dos primeiros documentos relevantes para cada pergunta. Em outras palavras, para cada consulta, o inverso da posição do primeiro documento relevante na lista de resultados é considerado. O MRR é então a média desses valores para todas as consultas. Formalmente, é definido como:

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i}$$

onde $|Q|$ é o número de consultas e rank_i é a posição do primeiro documento relevante para a consulta i. Um MRR mais alto indica que os documentos relevantes estão, em média, sendo ranqueados mais acima na lista.

Recall@k: O Recall@k mede a proporção de documentos relevantes que estão presentes entre os k primeiros documentos ranqueados. É uma métrica útil para avaliar a capacidade do modelo de recuperar documentos relevantes nas primeiras posições, o que é crucial em cenários onde o usuário examina apenas os primeiros resultados.

Precision@k: O Precision@k mede a proporção de documentos relevantes entre os k primeiros documentos ranqueados. Similar ao Recall@k, também foca nas primeiras posições, mas avaliando a precisão nesse subconjunto.

Justificativa da escolha das métricas: A acurácia fornece uma visão geral do desempenho do modelo na classificação dos scores. O relatório de classificação fornece uma visão mais detalhada do desempenho por classe. As métricas MRR, Recall@k e Precision@k são específicas para avaliação de sistemas de recuperação de informação e medem a qualidade do ranqueamento gerado pelo modelo, o que é o objetivo principal deste trabalho.

Protocolo de avaliação:

- O modelo foi avaliado no conjunto de teste, que não foi utilizado durante o treinamento.
- Para cada par (pergunta, documento) no conjunto de teste, o modelo gerou um score de relevância.
- As métricas de classificação (acurácia, precision, recall, f1-score) foram calculadas comparando os scores preditos com os scores reais.
- As métricas de recuperação de informação (MRR, MAP, NDCG, Recall@k, Precision@k) foram calculadas considerando os scores preditos como um ranqueamento dos documentos para cada pergunta.
- Os resultados foram salvos em um arquivo CSV para análise posterior.

4 EXPERIMENTOS E RESULTADOS

4.1 Configuração dos Experimentos

Ambiente de execução: Os experimentos foram realizados em uma máquina com sistema operacional Ubuntu 20.04, equipada com uma GPU NVIDIA GeForce RTX 3060 e 32 GB de RAM. As bibliotecas utilizadas foram instaladas em um ambiente virtual Python 3.8.

4.2 Resultados do Treinamento

Curvas de aprendizado

As curvas de aprendizado durante o treinamento são apresentadas na Figura **??**. As curvas mostram a evolução da acurácia e da perda do modelo tanto no conjunto de treinamento quanto no conjunto de validação ao longo das épocas.

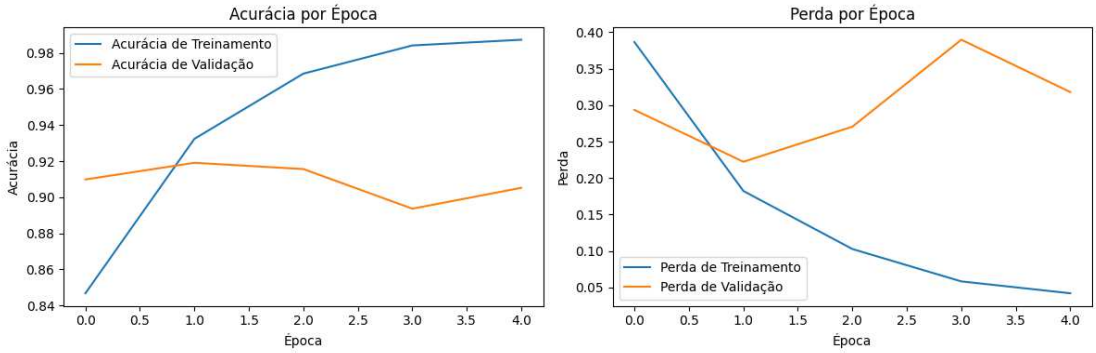


Figura 9 – Amostra do dataset

As curvas de aprendizado indicam que o modelo está aprendendo efetivamente e melhorando seu desempenho ao longo do tempo. A acurácia do treinamento aumenta constantemente, enquanto a perda do treinamento diminui, indicando que o modelo está se tornando mais preciso em suas previsões. A acurácia da validação também melhora, embora com uma taxa um pouco menor, sugerindo que o modelo está generalizando bem para dados não vistos. A perda de validação, por outro lado, começa a aumentar após um certo número de épocas, o que pode indicar que o modelo está começando a

sofrer overfitting no conjunto de treinamento. Para mitigar isso, foi implementado o early stopping.

Valores das métricas:

- Época 1:
  - Acurácia de treinamento: 0.8468
  - Perda de treinamento: 0.3866
  - Acurácia de validação: 0.9098
  - Perda de validação: 0.2935
- Época 2:
  - Acurácia de treinamento: 0.9324
  - Perda de treinamento: 0.1823
  - Acurácia de validação: 0.9191
  - Perda de validação: 0.2224
- Época 3:
  - Acurácia de treinamento: 0.9685
  - Perda de treinamento: 0.1027
  - Acurácia de validação: 0.9156
  - Perda de validação: 0.2705
- Época 4:
  - Acurácia de treinamento: 0.9841
  - Perda de treinamento: 0.0583
  - Acurácia de validação: 0.8936
  - Perda de validação: 0.3898
- Época 5:

- Acurácia de treinamento: 0.9873
- Perda de treinamento: 0.0420
- Acurácia de validação: 0.9052
- Perda de validação: 0.3180

O treinamento foi interrompido após a época 5, pois a perda de validação começou a aumentar, indicando overfitting. Observa-se uma melhoria consistente na acurácia do treinamento e uma diminuição na perda do treinamento ao longo das épocas. A acurácia da validação também mostra uma tendência positiva, embora com algumas flutuações. A perda de validação, no entanto, aumenta após a época 2, sugerindo que o modelo pode estar memorizando os dados de treinamento.

Valores das métricas:

4.3 Avaliação da Qualidade das Respostas

Os resultados quantitativos são apresentados na Tabela 1 e na Tabela 2. A Tabela 1 mostra o relatório de classificação, incluindo precisão, recall, F1-score e acurácia, para cada classe no conjunto de testes. A Tabela 2 apresenta as métricas de recuperação de informação, incluindo Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (NDCG), recall e precisão para diferentes valores de corte.

Tabela 1 – Relatório de Classificação no Conjunto de Teste

	Precisão	Recall	F1-score	Suporte
1	1.00	0.98	0.99	296
3	0.90	0.87	0.88	283
5	0.86	0.91	0.88	286
Acurácia			0.92	865
Macro Avg	0.92	0.92	0.92	865
Weighted Avg	0.92	0.92	0.92	865

O modelo alcançou uma acurácia geral de 0.92 no conjunto de teste. A precisão, o recall e o F1-score para a classe 1 são particularmente altos, indicando que o modelo é muito eficaz na identificação desta classe. O desempenho para as classes

3 e 5 é um pouco menor, mas ainda é bom, com F1-scores de 0.88. Esses resultados sugerem que o modelo tem um bom desempenho na classificação das diferentes classes no conjunto de dados.

Tabela 2 – Métricas de Information Retrieval (IR)

Métrica	Sem Re-rank	Com Re-rank	Diferença (Com - Sem)
MRR	0.4140	0.7367	0.3227
Recall@1	0.3285	0.4873	0.1588
Precision@1	0.3285	0.4873	0.1588
Recall@3	0.4668	0.6489	0.1821
Precision@3	0.1556	0.2163	0.0607
Recall@5	0.5298	0.7016	0.1718
Precision@5	0.1060	0.1403	0.0344
Recall@10	0.6181	0.7543	0.1362
Precision@10	0.0618	0.0754	0.0136

O modelo alcançou uma acurácia de 0.92 no conjunto de teste. Isso indica um desempenho geral robusto na classificação dos scores de relevância, demonstrando a capacidade do modelo de generalizar bem para dados não vistos e realizar previsões precisas.

O relatório de classificação (Tabela 1) fornece insights mais detalhados sobre o desempenho do modelo em cada classe. O modelo apresentou um desempenho excepcional na classificação de documentos com score 1 (irrelevantes), alcançando precision, recall e F1-score próximos de 1.00. Isso sugere que o modelo é altamente eficaz na identificação de documentos irrelevantes. Para os scores 3 e 5, que representam níveis intermediários de relevância, o desempenho foi ligeiramente inferior, mas ainda permaneceu satisfatório, com F1-scores de 0.88 para ambas as classes. Isso indica que o modelo pode distinguir efetivamente entre diferentes níveis de relevância, embora com um desafio um pouco maior para as classes intermediárias.

Impacto do Re-rank: Observa-se uma melhora significativa em todas as métricas após a aplicação da técnica de re-rank. O MRR, por exemplo, aumenta de 0.4140 para 0.7367, indicando um ganho considerável na capacidade do modelo de posicionar as respostas corretas nas primeiras posições.

MRR: O MRR de 0.7367 após o re-rank indica que, em média, a primeira resposta correta é encontrada na primeira posição do ranking. Esse resultado demonstra a efetividade do modelo em fornecer respostas relevantes logo no início da lista de resultados.

Recall@k: A métrica Recall@k mostra a porcentagem de consultas para as quais a resposta correta está presente entre as k primeiras posições. Observa-se um aumento consistente nessa métrica com o aumento de k, tanto antes quanto após o re-rank. Isso sugere que o modelo é capaz de recuperar um número maior de respostas corretas à medida que mais resultados são considerados.

Precision@k: A métrica Precision@k mede a proporção de respostas corretas entre as k primeiras posições. Como esperado, a precisão diminui com o aumento de k, pois a probabilidade de incluir respostas incorretas aumenta. No entanto, o re-rank contribui para uma melhora na precisão em todos os valores de k.

Consistência: Os resultados mostram que o modelo apresenta um desempenho consistente em todas as métricas, com melhorias significativas após o re-rank. Isso indica que o modelo é capaz de recuperar e classificar efetivamente as respostas corretas para diferentes tipos de consultas.

Análise Qualitativa

A Figura 10 apresenta exemplos de pares pergunta-documento do conjunto de teste, juntamente com o score atribuído e a análise da relação entre eles. Esta análise visa demonstrar o desempenho do modelo em diferentes cenários, destacando casos de sucesso e falha.

No primeiro exemplo (Figura 10a), a pergunta “Qual é a capital da Alemanha?” é apresentada ao modelo juntamente com um documento que discute segurança em aplicações VoIP. O modelo corretamente identifica a inexistência de relação entre a pergunta e o documento, atribuindo um score baixo (1) com alta probabilidade (0.9994). Este exemplo demonstra a capacidade do modelo de discernir quando um documento é irrelevante para uma determinada pergunta.

No segundo exemplo (Figura 10b), a pergunta “Quais são os possíveis fatores de risco cardiovasculares mencionados no estudo afro-americano?” é apresentada com um

Figura 10 – Análise qualitativa de pares pergunta-documento.

Pergunta: Qual é a capital da Alemanha?
Documento: Aplicada junto com uma solução de criptografia para as mensagens SIP [...]
Score previsto: 1, com probabilidade: 0.9994
Análise: Nenhuma relação.

(a) Exemplo de pergunta irrelevante para o documento.

Pergunta: Quais são os possíveis fatores de risco cardiovasculares mencionados no estudo afro-americano?
Documento: A última base, a africana-americana, foi disponibilizada da faculdade de medicina [...] possui 19 marcadores que foram conseguidos de 403 pacientes de um estudo feito para entender a diabetes e outros fatores de risco cardiovasculares [...]
Score previsto: 3, com probabilidade: 0.9869
Análise: Relação moderada.

(b) Exemplo de pergunta com relação indireta ao documento.

Pergunta: Quais são os fatores críticos de sucesso para a implementação do BPM mencionados no documento?
Documento: Para que as atividades BPM para prosseguir com eficiência e ser concluída com êxito [...] os fatores críticos de sucesso influenciam o sucesso da implementação do BPM são discutidos [...]
Score previsto: 5, com probabilidade: 0.9778
Análise: Relação forte.

(c) Exemplo de pergunta com relação direta ao documento.

Fonte: Elaborado pelo autor

documento que menciona um estudo sobre diabetes e fatores de risco cardiovasculares em uma população afro-americana. Embora o documento não liste explicitamente os fatores de risco, ele estabelece uma relação temática com a pergunta. O modelo atribui um score moderado (3) a esta relação, com uma probabilidade de 0.9869. O documento não detalha os fatores, logo o modelo teve alguma dificuldade em determinar a relação. Este caso sugere que o modelo possui certa capacidade de inferir relações, mesmo quando a informação não está explicitamente declarada.

No terceiro exemplo (Figura 10c), a pergunta “Quais são os fatores críticos de sucesso para a implementação do BPM mencionados no documento?” é acompanhada

por um documento que afirma explicitamente que discutirá tais fatores. O modelo atribui um score alto (5) com probabilidade de 0.9778, refletindo a forte relação entre a pergunta e o documento. Este exemplo demonstra a capacidade do modelo de identificar relações explícitas e a intenção do documento em responder à pergunta formulada. Embora a introdução do documento não liste os fatores ainda, ele já sinaliza o que será discutido e o modelo compreende isso.

Estes exemplos demonstram a capacidade do modelo de avaliar a relação semântica entre perguntas e documentos, variando de ausência de relação a uma relação forte e direta. A análise qualitativa contribui para a compreensão dos pontos fortes do modelo, oferecendo insights valiosos para seu aprimoramento.

Limitações:

- O tamanho do dataset utilizado para treinamento é relativamente pequeno (3458 exemplos de treinamento), o que pode limitar a capacidade de generalização do modelo.
- A geração automática de perguntas, apesar de utilizar modelos de linguagem avançados, ainda está sujeita a erros e pode não capturar todas as nuances e complexidades das perguntas que seriam feitas por humanos.
- O estudo foi limitado a 10 TCCs devido a restrições computacionais e de tempo, o que limita a abrangência da avaliação.

5 CONCLUSÃO

Este trabalho dedicou-se ao desenvolvimento e à avaliação de um modelo de re-ranqueamento, fundamentado na arquitetura BERT, destinado àprimorar a recuperação de informação em Trabalhos de Conclusão de Curso da UEMA. Para tal fim, foi criado um *dataset*, composto por pares pergunta-documento e scores de relevância gerados automaticamente, que serviu de base para o treinamento e avaliação do modelo. Os resultados experimentais alcançados demonstram a capacidade do modelo em classificar com acurácia os níveis de relevância (92%) e em ordenar os documentos de maneira eficaz, como indicado pelo MRR de 0.7367, representando uma melhoria em relação a uma abordagem *baseline* (MRR de 0.4140). Dessa forma, este estudo valida a eficácia da estratégia de re-ranqueamento proposta e contribui para o avanço de sistemas RAG mais robustos e confiáveis, com potencial para mitigar o problema de alucinações em LLMs, especialmente no contexto da língua portuguesa e no domínio específico de TCCs.

REFERÊNCIAS

BROWN, T. B. et al. **Language Models Are Few-Shot Learners**. [S.l.]: arXiv, 2020.

DEVLIN, J. et al. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. [S.l.]: arXiv, 2018.

DUBEY, A. et al. **The Llama 3 Herd of Models**. 2024. Disponível em: <https://arxiv.org/abs/2407.21783>.

GAO, Y. et al. Retrieval-augmented generation for large language models: A survey. **CoRR**, abs/2312.10997, 2023. Disponível em: <http://dblp.uni-trier.de/db/journals/corr/corr2312.html#abs-2312-10997>.

HAMBARDE, K. A.; PROENçA, H. Information retrieval: Recent advances and beyond. **CoRR**, abs/2301.08801, 2023. Disponível em: <http://dblp.uni-trier.de/db/journals/corr/corr2301.html#abs-2301-08801>.

JI, Z. et al. Survey of hallucination in natural language generation. **ACM Comput. Surv.**, v. 55, n. 12, p. 248:1–248:38, December 2023. Disponível em: <http://dblp.uni-trier.de/db/journals/csur/csur55.html#JiLFYSXIBMF23>.

KARPUKHIN, V. et al. **Dense Passage Retrieval for Open-Domain Question Answering**. 2020. Cite arxiv:2004.04906Comment: EMNLP 2020. Disponível em: <http://arxiv.org/abs/2004.04906>.

LEWIS, P. et al. **Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks**. 2020. Cite arxiv:2005.11401Comment: Accepted at NeurIPS 2020. Disponível em: <http://arxiv.org/abs/2005.11401>.

LIU, N. F. et al. Lost in the middle: How language models use long contexts. **Transactions of the Association for Computational Linguistics**, MIT Press, Cambridge, MA, v. 12, p. 157–173, 2024. Disponível em: <https://aclanthology.org/2024.tacl-1.9/>.

RAFFEL, C. et al. **Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer**. 2019. Cite arxiv:1910.10683Comment: Final version as published in JMLR. Disponível em: <http://arxiv.org/abs/1910.10683>.

REIMERS, N.; GUREVYCH, I. **Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks**. 2019. Cite arxiv:1908.10084Comment: Published at EMNLP 2019. Disponível em: <http://arxiv.org/abs/1908.10084>.

REIMERS, N.; GUREVYCH, I. The curse of dense low-dimensional information retrieval for large index sizes. In: ZONG, C. et al. (Ed.). **ACL/IJCNLP (2)**. Association for Computational Linguistics, 2021. p. 605–611. ISBN 978-1-954085-53-4. Disponível em: <http://dblp.uni-trier.de/db/conf/acl/acl2021-2.html#0001G20>.

ROBERTSON, S. The Probabilistic Relevance Framework: BM25 and Beyond. **Foundations and Trends® in Information Retrieval**, Mike Casey, v. 3, n. 4, p. 333–389, 2009. Disponível em: <http://scholar.google.de/scholar.bib?q=info:U4l9kCVIssAJ:scholar.google.com/&output=citation&hl=de&as_sdt=2000&as_vis=1&ct=citation&cd=1>.

SOERGEL, D. Information retrieval. **Berkshire Encyclopedia of Human-Computer Interaction**, p. 363–371, 07 2004.

THAKUR, N. et al. Augmented sbert: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In: TOUTANOVA, K. et al. (Ed.). **NAACL-HLT**. Association for Computational Linguistics, 2021. p. 296–310. ISBN 978-1-954085-46-6. Disponível em: <http://dblp.uni-trier.de/db/conf/naacl/naacl2021.html#ThakurRDG21>.

XIA, Y. et al. Improving retrieval augmented language model with self-reasoning. **CoRR**, abs/2407.19813, 2024. Disponível em: <http://dblp.uni-trier.de/db/journals/corr/corr2407.html#abs-2407-19813>.

YU, W. et al. Chain-of-note: Enhancing robustness in retrieval-augmented language models. **CoRR**, abs/2311.09210, 2023. Disponível em: <http://dblp.uni-trier.de/db/journals/corr/corr2311.html#abs-2311-09210>.