UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

VINÍCIUS LETTIÉRI PROENÇA

AUTOMATIZAÇÃO DA REVISÃO DE LITERATURA CIENTÍFICA COM GERAÇÃO AUMENTADA POR RECUPERAÇÃO

RIO DE JANEIRO
2024

VINÍCIUS LETTIÉRI PROENÇA

AUTOMATIZAÇÃO DA REVISÃO DE LITERATURA CIENTÍFICA COM GERAÇÃO AUMENTADA POR RECUPERAÇÃO

Trabalho de conclusão de curso de graduação apresentado ao Instituto de Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. João Carlos Pereira da Silva, D.Sc.

RIO DE JANEIRO

2024

CIP - Catalogação na Publicação

P964a

Proença, Vinícius Lettiéri
Automatização da Revisão de Literatura Científica com Geração Aumentada por Recuperação / Vinícius Lettiéri Proença. -- Rio de Janeiro, 2024.
59 f.

Orientador: João Carlos Pereira da Silva.
Trabalho de conclusão de curso (graduação) - Universidade Federal do Rio de Janeiro, Instituto de Computação, Bacharel em Ciência da Computação, 2024.

1. Geração Aumentada por Recuperação. 2. Grandes Modelos de Linguagem. 3. Revisão de Literatura Automatizada. 4. Inteligência Artificial. 5. Retrieval-Augmented Generation. I. Silva, João Carlos Pereira da, orient. II. Título.

Elaborado pelo Sistema de Geração Automática da UFRJ com os dados fornecidos pelo(a) autor(a), sob a responsabilidade de Miguel Romeu Amorim Neto - CRB-7/6283.

VINÍCIUS LETTIÉRI PROENÇA

AUTOMATIZAÇÃO DA REVISÃO DE LITERATURA CIENTÍFICA COM GERAÇÃO AUMENTADA POR RECUPERAÇÃO

Trabalho de conclusão de curso de graduação apresentado ao Instituto de Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 19 de Julho de 2024

BANCA EXAMINADORA:

gov.br Documento assinado digitalmente
JOAO CARLOS PEREIRA DA SILVA
Data: 30/07/2024 14:56:01-0300
Verifique em https://validar.iti.gov.br

João Carlos Pereira da Silva
D.Sc. (Instituto de Computação - UFRJ)

gov.br Documento assinado digitalmente
GISELI RABELLO LOPES
Data: 30/07/2024 15:01:38-0300
Verifique em https://validar.iti.gov.br

Giseli Rabello Lopes
D.Sc. (Instituto de Computação - UFRJ)

gov.br Documento assinado digitalmente
VIVIAN DOS SANTOS SILVA
Data: 30/07/2024 15:58:31-0300
Verifique em https://validar.iti.gov.br

Vivian dos Santos Silva
Ph.D (Instituto de Computação - UFRJ)

À minha família por todo o apoio que sempre me deram em todas as áreas da minha vida. Em especial à minha mãe Patrícia e meu pai José, por toda a dedicação e preocupação com minha educação e formação tanto acadêmica como de vida. À minha namorada Vívian, por todo o afeto, parceria, paciência e incentivo que me deu nesse período. Aos meus grandes amigos desde o ensino médio, João Gabriel; do ensino médio e faculdade, Poppolino, Mota e Gorchinsky; e meu grande parceiro de faculdade, Thierry.

AGRADECIMENTOS

Gostaria de agradecer profundamente ao meu professor orientador João Carlos, por ter acreditado no projeto e me acompanhado tão de perto em cada etapa dele. Sua presença foi essencial para o desenrolar do trabalho. Estendo o agradecimento à todos os professores que tive durante a graduação, pois esse é um projeto de anos, pensado e instigado desde o início do curso, e todos contribuíram de alguma maneira para a compreensão da problemática e na reflexão de possíveis soluções.

Importante também expressarmos nossa gratidão à *Cohere* pelo apoio a esta pesquisa através da provisão de créditos gratuitos para sua *API*. Esta colaboração foi fundamental para testar e validar a eficácia do sistema.

"Everything around you that you call life
was made up by people that were no smarter than you.
And you can change it, you can influence it, you can build
your own things that other people can use"

Steve Jobs

RESUMO

A revisão de literatura, embora essencial para a pesquisa científica, é muitas vezes percebida como uma etapa exploratória demorada, que poderia ser melhor aproveitado para pensamento crítico e na experimentação. Os Grandes Modelos de Linguagem (LLMs) revolucionaram a busca de informações, fazendo uma transição da exploração ativa para uma experiência mais passiva de recuperação de informação. No entanto, quando utilizados como fonte primária de conhecimento, os modelos muitas vezes produzem conteúdo impreciso ou "alucinado" e carecem de atualizações de informações em tempo real, devido à dependência da memória paramétrica do modelo. O paradigma da Geração Aumentada por Recuperação (RAG) surgiu como uma solução para esses desafios de lidar com tarefas de intenso uso de conhecimento, combinando a robusta capacidade da recuperação de informação tradicional com a sofisticada geração de texto de LLMs. Este estudo propõe utilizar a arquitetura RAG para auxiliar na automatização da fase de levantamento bibliográfico para revisões de literatura, contando com uma recuperação de documentos relevantes aprimorada, com maior compreensão do conteúdo dos documentos, e uma experiência de usuário enriquecida por meio de sumarização do material e síntese da relevância dos resultados. O objetivo é simplificar o processo de busca, com resultados mais relevantes e permitindo uma pré-seleção mais eficiente, através das sumarizações. O estudo foca em documentos científicos em português, para expandir o campo de estudo para além do inglês. Os resultados finais são promissores para recuperação e geração, indicando potencial para ampliar a pesquisa para um banco de dados mais completo.

Palavras-chave: Revisão de Literatura Automatizada; RAG; Geração Aumentada por Recuperação; LLM; Grandes Modelos de Linguagem;

ABSTRACT

Literature Review, although essential for scientific research, is often perceived as a time-consuming exploratory stage that could be better used for critical thinking and experimentation. Large Language Models (LLMs) have revolutionized information searching, making a transition from active exploration to a more passive information retrieval experience. However, when used as a primary source of knowledge, models often produce inaccurate or "hallucinated" content and lack real-time information updates, due to the model's reliance on parametric memory. The Retrieval Augmented Generation (RAG) paradigm has emerged as a solution to these challenges of dealing with knowledge-intensive tasks, combining the robust capacity of traditional information retrieval with the sophisticated text generation of LLMs. This study proposes using the RAG architecture to help automate the bibliographic survey phase for literature reviews, with improved retrieval of relevant documents with greater understanding of the content of the documents, and an enriched user experience through summarization of the material and synthesis of the relevance of the results. The goal is to simplify the search process, with more relevant results and enabling more efficient pre-selection through summarization. The study focuses on scientific documents in Portuguese, in order to expand the field of study beyond English. The final results are promising for retrieval and generation, indicating potential for extending the search to a larger database.

Keywords: Automated Literature Review; RAG; Retrieval-Augmented Generation; LLM; Large Language Model;

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

RI	Recuperação da Informação
LLM	Large Language Model
RAG	Retrieval-Augmented Generation
PLN	Processamento de Linguagem Natural
TF-IDF	Term Frequency — Inverse Document Frequency
BoW	Bag of Words
CBoW	Continuous Bag of Words
RNN	Recurrent Neural Network
LSTM	Long-Short Term Memory
OCR	Optical Character Recognition
GPT	Generative Pre-trained Transformer
BERT	Bidirectional Encoder Representations from Transformers
UFRJ	Universidade Federal do Rio de Janeiro
RIS	Research Information Systems
PDF	Portable Document Format
GPU	Graphics Processing Unit
CPU	Central Processing Unit
MR	Mean Rank (Métrica de Validação)
MRR	Mean Reciprocal Rank (Métrica de Validação)
P@K	Precision at K (Métrica de Validação)
CR	Context Relevance (Métrica de Validação)
AR	Answer Relevance (Métrica de Validação)
F	Faithfulness (Métrica de Validação)
AS	Answer Semantic Similarity (Métrica de Validação)

SUMÁRIO

1 INTRODUÇÃO

O processo de revisão da literatura, embora seja uma parte essencial da pesquisa acadêmica, é muitas vezes considerado demorado e cansativo, exigindo extensa exploração e análise de numerosos artigos para identificar aqueles mais relevantes para um novo estudo (AGARWAL et al., 2024; LI et al., 2024). Os métodos tradicionais de pesquisa empregados pelos pesquisadores podem ser ineficientes, baseando-se principalmente na correspondência exata de palavras, o que desconsidera a compreensão do contexto ou das nuances do tópico em questão. Os resultados desse tipo podem ser menos relevantes (HAMBARDE; PROENCA, 2023), visto que pesquisas com objetivos muito distintos podem conter as mesmas palavras contidas na consulta realizada. Isso gera uma extensa lista de documentos a serem analisados, cuja maioria não se enquadra nas especificidades.

Este estudo visa aumentar a produtividade da pesquisa científica ao sugerir ferramental para automatização e aceleração da fase de levantamento bibliográfico, com artigos relevantes para uma nova pesquisa a ser realizada. O objetivo é entregar resultados personalizados que atendam às necessidades específicas do pesquisador e seu objeto de estudo, melhorando o sistema de busca e oferecendo resumos concisos com informações relevantes que facilitem uma primeira análise e filtragem mais rápida e eficaz. Importante notar que o sistema *não almeja substituir* o processo de revisão da literatura e leitura dos documentos, mas otimizar a descoberta e seleção de artigos relevantes.

Os Grandes Modelos de Linguagem, ou em inglês *Large Language Models* (LLMs), revolucionaram o cenário da recuperação de informações, passando da exploração ativa para formas mais passivas de recuperação (GAO et al., 2023). Com seu grande conhecimento embarcado, os modelos podem executar tarefas complexas com base em instruções textuais simples e sem a necessidade de treinar ou refinar para tarefas específicas, democratizando o acesso à tecnologia de linguagem de última geração (BROWN et al., 2020). Porém, apesar de sua assertividade, quando utilizados como fonte de conhecimento, esses modelos muitas vezes produzem conteúdos "alucinados" (JI et al., 2023) e não possuem a capacidade de incorporar informações em tempo real, pois utiliza os conhecimentos implícitos no modelo (LEWIS et al., 2020).

O paradigma da Geração Aumentada por Recuperação (RAG - *Retrieval-Augmented Generation*) (LEWIS et al., 2020), surgiu como uma solução para os desafios das tarefas com intenso uso de conhecimento, ao combinar capacidades robustas de recuperação de informação por bases de dados, como fonte não paramétrica de informação, com a sofisticada geração de texto de LLMs (LI et al., 2022). Esta integração permite que os modelos enriqueçam as suas respostas com dados factuais, resultando em resultados mais precisos e contextualmente relevantes.

Avanços recentes no fluxo de trabalho de pesquisa por meio de aprendizado de máquina

e modelos de linguagem introduziram várias ferramentas:

- *Elicit*[1] se especializa na assistência de revisão de literatura com ferramentas de levantamento bibliográfico, agrupamento de conceitos e resposta a perguntas.
- *Semantic Scholar*[2] possui um complexo grafo que relaciona publicações, autores e classificações, e ferramentas de recomendação e busca que se aproveitam do grafo. Além disso, em sua busca de documentos, cria um pequeno resumo para cada artigo usando um LLM, para facilitar uma rápida pré-seleção.
- *System Pro*[3] faz revisão de literatura com foco na extração e consolidação de estatísticas de publicações.
- *Consensus App*[4] é uma ferramenta que analisa a literatura de um tema pesquisado e resume o consenso do material encontrado. Sua ferramenta é melhor utilizada em cenários de pergunta e resposta, em que a resposta final é consolidada.
- *Scite AI*[5] é um sistema de pergunta e resposta geral, em que o resultado final é suportado por artigos encontrados na literatura com as respectivas referências.

Uma limitação comum dessas ferramentas é que ainda dependem de bases de dados de documentos predominantemente em inglês. É possível encontrar documentos em português mas existe uma grande defasagem. Outro ponto é que essas ferramentas não se baseiam no texto completo para todos os documentos. Em sua maioria utilizam apenas o *abstract*, pois conseguir acesso ao texto processado de todos os documentos é complexo.

Este estudo propõe a utilização da arquitetura RAG para automatizar o processo de levantamento bibliográfico no contexto de revisões de literatura. Ao receber uma consulta do usuário com a descrição detalhada do seu tema de pesquisa, o sistema utiliza modelos de linguagem para interpretar o texto e identificar os documentos e seus parágrafos mais relevantes à consulta realizada, por meio da busca semântica vetorial e a reclassificação por relevância. Em seguida, cria resumos personalizados para cada documento recuperado, juntamente com explicações sobre sua relevância para a consulta original. Para atingir isso, um extenso processamento de dados é necessário para consolidar a base de publicações, visto que a maior parte das pesquisas são armazenadas em arquivo PDF. Nesta pesquisa serão utilizados os textos completos dos artigos, para garantir mais informações a serem recuperadas. Além disso, os documentos utilizados serão um subconjunto de publicações da UFRJ em português.

[1] https://elicit.com/
[2] https://semanticscholar.org/
[3] https://pro.system.com/
[4] https://consensus.app/search/
[5] https://scite.ai/assistant

Para validar o sistema, o *Framework* RAGAS (ES et al., 2023) foi adaptado para atender às necessidades específicas do estudo, juntamente com métricas comuns da literatura para recuperação e validação de geração, com testes por componente e de ponta a ponta.

Em resumo, este estudo visa contribuir e avançar a produtividade dos pesquisadores, ao automatizar tarefas demoradas durante a fase de exploração e potencialmente melhorar a qualidade dos resultados finais. Ao simplificar os fluxos de trabalho dos pesquisadores e reduzir o tempo gasto em tarefas preliminares, pode-se liberar tempo para o pensamento crítico e a experimentação. Espera-se que esta melhoria na eficiência beneficie vários campos do conhecimento, acelerando, em última análise, o ritmo da descoberta científica e da inovação.

Esta monografia foi organizada da seguinte forma: o Capítulo 2 cobre os conceitos fundamentais da recuperação de informação, modelos de linguagem e o RAG, essenciais como insumo para o entendimento da arquitetura proposta; no Capítulo 3 abordam-se os trabalhos relacionados à automação de revisão de literatura e sistemas fundamentados pelo RAG; o Capítulo 4 aprofunda-se em cada uma das etapas da arquitetura proposta; o Capítulo 5 aborda como os experimentos foram realizados, apresenta e analisa os resultados das métricas extraídas; e o Capítulo 6 consolida toda a pesquisa e suas descobertas, além de possíveis melhorias em próximos trabalhos e uma discussão da ética envolvida no uso desse tipo de sistema para a academia.

2 CONCEITOS

Neste capítulo, apresentaremos conceitos chave que fundamentam esta pesquisa, estabelecendo um entendimento comum sobre este tópico. A explanação seguirá uma ordem cronológica, desde os princípios básicos de informação até abordagens avançadas de Processamento de Linguagem Natural (PLN). O foco não será aprofundar em detalhes técnicos, mas compreender e aplicar cada conceito na metodologia do sistema proposto.

2.1 RECUPERAÇÃO DA INFORMAÇÃO

A essência desta pesquisa encontra suas raízes em um campo interdisciplinar que se situa na interseção da Ciência da Computação e Ciência da Informação, conhecido como Recuperação da Informação (do inglês, *Information Retrieval*), ou pela sigla RI. Este campo, apesar de em alta, não é novo, e em sua definição abrange um espectro vasto e diversificado de estudos.

Dentre as figuras emblemáticas da área, destaca-se *Gerard Salton*, amplamente reverenciado como o pai da Recuperação da Informação. Em seu livro *Automatic Information Organization and Retrieval* (SALTON, 1968), define o campo como:

> Recuperação da Informação é uma área preocupada com a estrutura, análise, organização, armazenamento, pesquisa e recuperação de informações *(Tradução Livre)*

Sua definição transcende a simples recuperação, incluindo tarefas de interação, sumarização e extração de conhecimento a partir da informação. Com a evolução em técnicas de PLN, modelos de linguagem nos ofereceram a capacidade de interpretação e geração de texto automatizada, permitindo que tarefas de consolidação, filtragem e síntese, antes realizadas manualmente, pudessem ser realizadas de maneira automatizada. Este desenvolvimento representa uma economia significativa de tempo, simplificando o acesso à informação relevante e melhorando a eficiência do processo de busca (ZHU et al., 2023).

Áreas como automatização de perguntas e respostas (*Question-Answering*), detecção e rastreamento de tópicos, sumarização, recuperação multimídia (incluindo imagens, vídeos e músicas), estruturação de texto, mineração de texto e genômica, começaram a fazer parte efetiva do escopo de RI (ALLAN et al., 2003). Para os propósitos desta monografia, nos concentraremos nos seguintes aspectos da Recuperação da Informação:

2.1.1 Técnicas para Recuperação de Documentos

No nascer do campo de Recuperação da Informação, os sistemas de busca textual se baseavam essencialmente na correspondência direta de termos (em inglês, *Exact Match*)

entre as consultas dos usuários e os documentos disponíveis, pouco eficaz com termos similares. Melhorias, com o uso de palavras-chave e a técnica de aumento de texto (*text augmentation*), em que palavras similares são acrescentadas ao texto para facilitar a busca exata, foram aplicadas, mas ainda não chegaram ao nível cognitivo humano. (HAMBARDE; PROENCA, 2023)

Em uma virada recente, o PLN experimentou avanços notáveis em como informações podem ser recuperadas, com técnicas novas como a busca semântica e modelos de linguagem (HAMBARDE; PROENCA, 2023). Podemos dividir as técnicas de recuperação nos seguintes tipos:

Busca Booleana. Um dos primeiros e mais simples métodos, opera através de uma correspondência exata dos termos da consulta e documentos. Um problema é a polissemia (uma palavra com múltiplos significados), a sinonímia (múltiplas palavras para um mesmo significado) e as falhas lexicais (palavra em uma língua que não existe em outra), visto que sua busca é muito exata e não se molda a variações linguísticas, prejudicando a eficácia do modelo. (BAEZA-YATES; RIBEIRO-NETO et al., 1999; HAMBARDE; PROENCA, 2023)

Busca em Espaços Vetoriais. Com o objetivo de solucionar o problema de variações linguísticas e representar melhor a semântica de palavras, *Gerard Salton* sugeriu um modelo baseado em espaço vetorial (SALTON; WONG; YANG, 1975), tal que cada termo tivesse seu próprio vetor no espaço e a proximidade entre eles indicasse a proximidade semântica. Desde a concepção, com a evolução dos modelos de linguagem esse tipo de busca permitiu espaços vetoriais não só de palavras como de parágrafos e até documentos inteiros. Uma explicação mais aprofundada pode ser encontrada na Seção 2.3.

Busca com Modelos Probabilísticos. Calculam a probabilidade de um documento ser relevante para o usuário, tornando-os adequados para recomendação personalizada. Esses modelos, aprimorados por técnicas de *Inferência Bayesiana*, têm complexidade elevada na construção e dependem de suposições simplificadas, o que pode limitar sua aplicação prática (BAEZA-YATES; RIBEIRO-NETO et al., 1999).

Busca Híbrida. Apesar da grande diferença entre os métodos, cada um possui seu próprio ponto forte e a melhor opção para a eficiência da busca é uma combinação. Um caso muito utilizado é o *Exact Match* com *Busca Semântica* (YAN, 2023). A primeira, ideal para encontrar termos específicos em textos, e a última, possui um desempenho melhor em situações de similaridades semânticas entre documentos.

Uma outra nomenclatura muito utilizada, mais recentemente, é a *busca semântica*. Esta se refere a utilizar significados semânticos para encontrar documentos ou informações em uma base de dados. Existem várias formas de realizar esse tipo de busca, por exemplo

com um grafo semântico, em que a consulta navega os relacionamentos para encontrar alguma informação, ou então através dos espaços vetoriais, que condensam informação em um vetor com semântica.

Nesta pesquisa, será explorada a técnica de *Busca em Espaços Vetoriais*, uma das técnicas para realizar *Busca Semântica*. Essa escolha se deve à natureza da busca que o sistema propõe, com o objetivo de encontrar os artigos científicos que estudam o mesmo tópico de pesquisa e sejam o mais similar possível a ele.

2.1.2 Filtragem de Documentos por Relevância

Uma arquitetura comum na busca por documentos é adicionar uma camada mais especializada e refinada após a recuperação de informação em massa da base, que é o caso da filtragem por relevância. A primeira etapa lida com um volume muito grande de dados e precisa de modelos mais simples e velozes para operar, enquanto a filtragem possui modelos de linguagem mais refinados e complexos, que comparam o texto dos documentos e da consulta, para identificar a relevância. (HAMBARDE; PROENCA, 2023)

A filtragem por relevância, apesar do nome, também é capaz de ordenar os documentos por relevância, em uma técnica chamada de *Document Reranking*, essencial para garantir a qualidade dos resultados (ZHU et al., 2023). Essa será uma funcionalidade essencial no funcionamento do sistema proposto nesta pesquisa, melhorando a relevância dos documentos selecionados.

2.1.3 Sumarização Automática de Texto

A Sumarização Automática de Texto, um ramo em recente ascensão dentro da RI, permite que sistemas aprimorem os resultados de busca oferecidos ao usuário final, através da síntese das informações encontradas, facilitando o acesso a vastas quantidades de dados com maior escalabilidade. (ALLAN et al., 2003)

Dentro do universo da RI, o componente dedicado à sumarização é denominado *Componente de Leitura*. Os sistemas de RI tipicamente consistem em um *Componente de Recuperação*, responsável por encontrar e filtrar as informações relevantes ao contexto, e um *Componente de Leitura*, que sintetiza e sumariza para otimizar o tempo do usuário final. (ZHU et al., 2023)

No contexto de revisão de literatura para pesquisas científicas, a sumarização automática oferece uma ferramenta valiosa para os pesquisadores. Permite uma triagem inicial mais eficaz, evitando que precise se aprofundar e gastar tempo com artigos de pouca relevância para sua pesquisa. Graças à automatização, as sumarizações adaptam-se às necessidades específicas de cada pesquisa, otimizando o processo de busca por documentos relevantes (AGARWAL et al., 2024). O presente trabalho utilizará a sumarização como ponta final da geração do texto mostrado ao usuário.

2.2 PROCESSAMENTO DE LINGUAGEM NATURAL

A área de Processamento de Linguagem Natural (PLN), explora técnicas para automatizar a interpretação e geração de texto através de algoritmos, sem a necessidade de inteligência humana envolvida. Nesta seção, serão pontuados momentos chave na linha do tempo da PLN, que podem ajudar a entender a construção das tecnologias mais recentes, que serão utilizadas no estudo.

Em seus dias iniciais, o PLN seguiu a *abordagem simbólica*, a partir de regras escritas à mão para manipulação de símbolos. Algumas técnicas como *stemização e lematização* (redução de palavras para sua forma base, removendo inflexões) e *tokenização* (identificação das palavras por números/tokens) foram importantes para o início da representação textual em máquinas, apresentadas na Figura 1. Um dos trabalhos relevantes foi o de *Chomsky*, em *Syntatic Structures* (CHOMSKY, 1957), que buscou uma gramática universal, ao introduzir uma metodologia que traduz a linguagem natural em um formato utilizável por computadores. Neste momento, palavras são identificadas por números, sem nenhum relacionamento ou significado atrelado.

Figura 1 – Stemização, Lematização e Tokenização no PLN

Nas décadas seguintes, até os anos 80, a abordagem se transformou para estatística. Um método que se tornou comum foi o *Bag of Words*, cujo princípio fundamental é que textos sobre assuntos semelhantes tenderão a possuir o mesmo conjunto de palavras. Dessa forma, seria possível agrupar e contar as palavras de um texto e criar um vetor que identificaria quais palavras cada texto possui, e assim analisar a similaridade entre textos, como indica a Figura 2. Esse foi um começo para análises de texto, mesmo com as palavras tendo identificadores rígidos, sem semântica.

Figura 2 – Bag of Words no PLN

Um problema do *Bag of Words* é a contagem excessiva de termos comuns, podendo tornar todos os textos muito parecidos. Um avanço criado foi o TF-IDF (JONES, 1972), que significa *Frequência do Termo – Frequência Inversa dos Documentos*, cuja fórmula está representada na Equação 2.1. Cada termo do dicionário possui um peso (IDF), calculado

pelo logaritmo do inverso da fração de vezes que o termo aparece nos documentos, ou seja, quanto menos comum, maior o peso. Os valores finais utilizados no vetor TF-IDF são dados pela frequência dos termos (TF) multiplicado pelo seu peso de relevância (IDF), resultando nos valores da Figura 3.

$$\text{TF-IDF} = TF(term, doc) \times IDF(term) = TF(term, doc) \times log\left(\frac{N}{DF(term)}\right) \quad (2.1)$$

Figura 3 – Bag of Words com TF-IDF no PLN

Esses modelos formam o primórdio da análise de similaridade entre textos. A partir dos vetores, seria possível encontrar similaridades entre os textos a partir de funções de distância entre vetores, como a função cosseno (Equação 2.2). Porém, ainda não há uma real interpretação do texto, a ordem das palavras não importa e a polissemia (mesma palavra para significados diferentes) não é tratada. Esse tipo de busca é classificada como uma *busca vetorial esparsa*, pois como as dimensões equivalem aos termos, em sua maioria os vetores possuem muitos zeros, ou seja, esparsos.

$$\text{Similaridade Cosseno} = cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} \quad (2.2)$$

Durante a década de 80 e 90, uma nova abordagem de aprendizado se expandiu, as Redes Neurais. No caso do PLN, uma evolução importante foram as redes recorrentes, em que a sequência das palavras no texto ganhou relevância, como normalmente seria em um texto. Neste contexto, Redes Neurais Recorrentes (RNNs) (RUMELHART; HINTON; WILLIAMS, 1986) e Memória Longa de Curto Prazo (ou LSTM) (HOCHREITER; SCHMIDHUBER, 1997) surgem para melhorar o processamento e compreensão de textos e se tornam a metodologia padrão para qualquer tarefa de compreensão de texto.

Na Figura 4 é mostrado o funcionamento básico de uma RNN, tanto na visão de um neurônio da rede quanto no fluxo de dados desse mesmo neurônio ao longo do tempo. O grande diferencial deste tipo de modelo é que seu estado oculto h^t é calculado com base na entrada x^t e no estado h^{t-1} anterior. Dessa forma, uma frase introduzida palavra a palavra (x^t) no modelo altera iterativamente o estado oculto h^t, absorvendo toda a informação sequencial. Existem diversas formações pare redes recorrentes, com entradas e saídas N para N, 1 para N ou N para 1. Na figura, cada entrada possui uma saída, calculada com cada estado oculto.

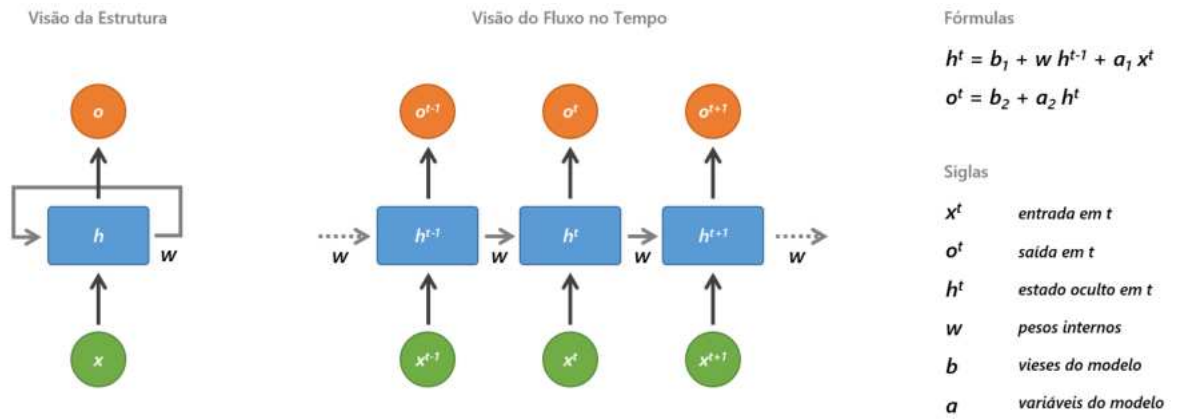

Figura 4 – Funcionamento de uma Rede Neural Recorrente

Em 2013, o modelo *Word2Vec* (MIKOLOV et al., 2013) foi lançado, mudando o paradigma de representação semântica no PLN. As palavras deixam de ser marcadas pelos *tokens*, considerados vetores esparsos, e passam a ser representadas por vetores densos em um mesmo espaço vetorial, trazendo semântica a partir da área posicionada e a proximidade entre as mesmas. Esse posicionamento e relacionamento entre os vetores pode ser observado na Figura 5, em que vetores de significado masculino estariam mais próximos entre si que os pares femininos. Além disso, mesmo que separados, os vetores carregam informação direcional tal a relação entre homem e rei é a mesma de mulher e rainha. Ou seja, as operações entre os vetores são possíveis: $\vec{rei} - \vec{homem} + \vec{mulher} = \vec{rainha}$

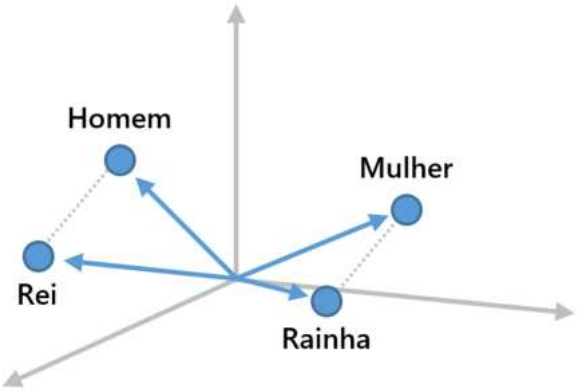

Figura 5 – Representação Visual da Relação de Embeddings no Word2Vec

Mais tarde, estes vetores são denominados *embeddings*, que terá o funcionamento aprofundado na Seção 2.3. Esse tipo é classificado como *busca vetorial densa*, pois as dimensões absorvem muita informação e, diferente do *bag-of-words*, não representa cada palavra em uma dimensão. Neste momento da cronologia, os *embeddings* de palavras ainda desconsideram o contexto em que estão inseridos. Cada palavra tem seu *embedding* fixo, o que pode levar a compreensões erradas em caso de homonímia e polissemia.

Em 2017, a introdução da arquitetura *Transformer* e seu mecanismo de auto-atenção (VASWANI et al., 2017) foram um marco para a pesquisa em modelos de rede neural para linguagem. O grande destaque dessa nova arquitetura é a capacidade do mecanismo de auto-atenção em olhar individualmente para cada palavra, analisar suas vizinhas, determinar como cada palavra vizinha afeta o significado semântico desta, e gerar um *embedding* corrigido com base no seu contexto, para cada termo do texto. Essa mudança não só melhorou a funcionalidade dos *embeddings*, resolvendo os problemas de homonímia e polissemia, como criou um algoritmo paralelizável e altamente escalável. A arquitetura do *Transformer* pode ser encontrada na Figura 9 da Seção 2.4, composta pela Camada de Atenção e o Perceptron Multicamada.

No fim da década de 2010 e início de 2020, vimos uma aceleração nos avanços de modelos pré-treinados com arquitetura *Transformer*, para geração de *embeddings* e geração de texto. Com cada vez mais parâmetros, esses novos modelos de linguagem criaram um ramo especializado, os *Grandes Modelos de Linguagem*. O maior salto foi com o lançamento do GPT-3 (BROWN et al., 2020), em que a capacidade de modelos de linguagem escaparam do contexto acadêmico para uma disseminação popular. Desde então, as aplicabilidades relacionadas aos modelos de linguagem cresceram e geraram muitas ferramentas novas.

Graças a essa evolução das arquiteturas para modelos de PLN foi possível chegar à técnicas mais avançadas de recuperação de informação e sumarização de texto, que serão utilizadas nesta pesquisa. O sistema de recuperação se fundamenta em *embeddings* para documentos, obtidos por modelos de linguagem baseados na arquitetura *Transformer*, e o componente de geração de texto utiliza o GPT-3, também baseado em *Transformer*. Nas próximas seções, duas partes específicas dessa evolução serão aprofundadas, os *embeddings* e os grandes modelos de linguagem.

2.3 EMBEDDINGS E A BUSCA SEMÂNTICA VETORIAL

Em um contexto geral, os *embeddings* são representações vetoriais de objetos em um espaço multidimensional, tal que a posição e direção desses vetores capturam a relação semântica desses objetos. Esse conceito pode ser aplicável a textos, imagens, modelos tridimensionais para reconhecimento facial, documentos ou qualquer outro tipo de objeto em que possa haver comparação por semelhança.

Esse comportamento de organização semântica é natural e pode ser encontrada em vários contextos da ciência da informação. Na biblioteconomia, por exemplo, nosso objeto são os livros e o espaço é a biblioteca. A formação e organização do espaço em seções, fileiras, estantes e prateleiras permite que qualquer livro seja facilmente encontrável com base em seu conteúdo. Além disso, livros de conteúdo similares estarão próximos entre si dentro desse espaço. Essa mesma formação poderia ser aplicada com *embeddings*, em

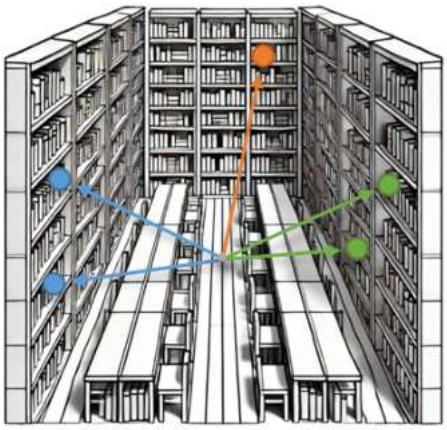

Figura 6 – Representação de Embeddings em uma Biblioteca

que os livros seriam condensados em um vetor e posicionados dentro do espaço vetorial, permitindo que buscas de similaridade pudessem ser realizadas, como mostra a Figura 6.

No contexto de representação das palavras, podemos definir formalmente os *Word Embeddings* por (ALMEIDA; XEXÉO, 2019):

> São vetores de palavras densos, distribuídos e de comprimento fixo, construídos usando estatísticas de coocorrência de palavras conforme a hipótese distribucional. *(Tradução Livre)*

Uma parte importante dessa definição, que nos dá arcabouço para gerar um espaço semântico, é a *Hipótese Distribucional* (HARRIS, 1954), em que se afirma que a distribuição estatística dos elementos linguísticos no contexto determina seu comportamento semântico. Isso implica que distribuir palavras no espaço, como exemplo da Figura 5 é possível e permite relacionamento entre as palavras.

O processo de treinamento de modelos de *embedding* para palavras é iterativo, através de um grande volume de textos recuperados na língua que o *embedding* se propõe. O modelo, inicialmente vazio, começa a ser treinado palavra por palavra através do relacionamento com as demais no texto. O *Word2Vec*, por exemplo, como mostra a Figura 7, funciona com dois componentes: o *Continuous Bag of Words* (CBOW) que se propõe a predizer a palavra alvo com base nas vizinhas, e o *Skip-Gram* que utiliza a palavra predita do anterior para predizer as palavras próximas que haviam sido inseridas como entrada do modelo (MIKOLOV et al., 2013).

No CBOW, as palavras são codificadas com *one-hot encoding*, um tipo de representação vetorial esparsa, e através da matriz de peso interna W, o vetor esparso é mapeado para seu *embedding*, um tipo de vetor denso. Por fim, no Skip-Gram, a palavra predita o_t é passada pela matriz de peso interna W para obter os vetores de probabilidade para cada palavra, cuja função *softmax* seleciona as de maior probabilidade. O modelo é treinado

para minimizar o erro entre as palavras de contexto iniciais e as preditas através de *backpropagation*, iterativamente modificando a matriz W de pesos internos, responsável pela geração dos *embeddings*.

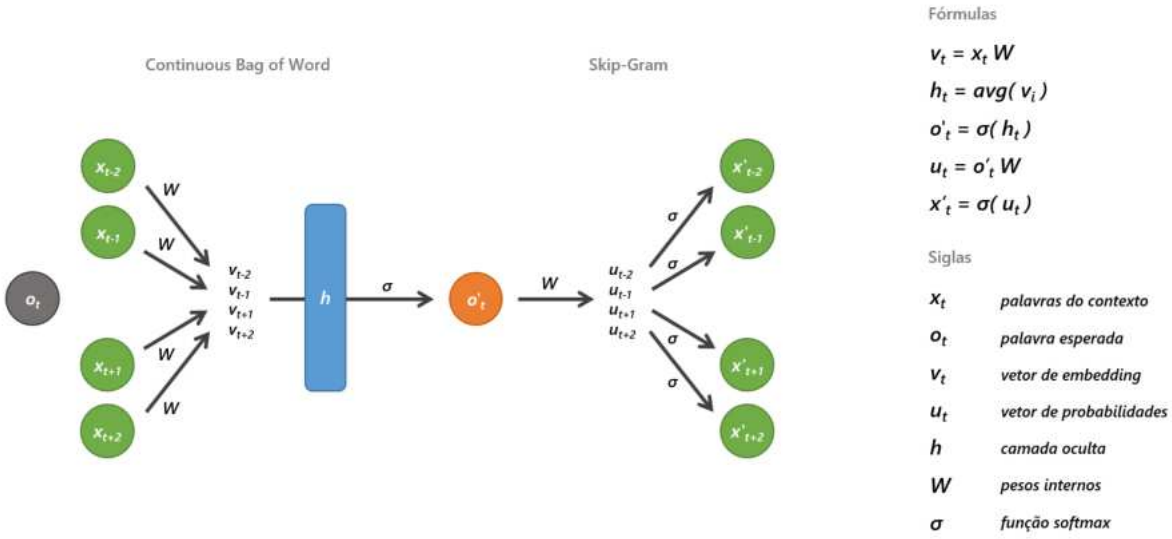

Figura 7 – Arquitetura do Word2Vec (CBOW + Skip-Gram)

Inicialmente utilizado para representação de palavras, o termo *embedding* passou a ser utilizado para representação de textos e documentos, com o surgimento de novos modelos de linguagem mais robustos, capazes de interpretar textos completos. Além disso, os novos modelos permitiram que os *embeddings* das palavras fossem específicos para o contexto que estão no texto, resolvendo o problema da polissemia e homonímia.

Outro detalhe importante é que *embeddings* são universais e podem representar qualquer língua dentro de um mesmo espaço vetorial, com preservação semântica entre as línguas. Mas, para que isso ocorra, o treinamento dos modelos deve ser feito com textos nas línguas esperadas de serem utilizadas. Por isso existem *modelos multilíngues*, treinados em dezenas de línguas, e que permitem realizar buscas em documentos sem a necessidade de identificar ou traduzir a língua para indexar.

Os *embeddings* criaram o campo da *busca vetorial densa*, devido a como a informação é condensada no vetor. Essa é uma das técnicas dentro da busca semântica, como falado anteriormente.

Como muitas pesquisas utilizam referências internacionais, parafraseiam artigos em outras línguas e utilizam muitos termos estrangeiros, a escolha por um modelo de linguagem *multilíngue* para geração de *embeddings* foi importante para este trabalho. No Capítulo 4, da metodologia de construção do sistema, será abordado mais a fundo o modelo utilizado e como foi implementado.

2.4 GRANDES MODELOS DE LINGUAGEM

O diferencial dos Grandes Modelos de Linguagem, ou *Large Language Models* (LLMs), para os demais é seu tamanho significantemente maior em termos de parâmetros. Em redes neurais, chamamos de parâmetro cada unidade de variável treinada em vetores ou matrizes internas, como é o caso da matriz W no *Word2Vec*, em que cada célula da matriz é um parâmetro. Enquanto o modelo BERT, lançado em 2019, possuía na casa de centenas de milhões de parâmetros treinados (DEVLIN et al., 2018), modelos mais recentes, como o GPT-3 foram treinados com centenas de bilhões de parâmetros (BROWN et al., 2020), e o GPT-4 treinado com centenas de trilhões de parâmetros (KOUBAA, 2023). Com esse aumento exponencial no número de parâmetros é compreensível que estes modelos consigam interpretar tão bem a linguagem e se articular naturalmente.

Figura 8 – LLM: Geração de Tokens por Probabilidade

O funcionamento técnico por trás dos Modelos de Linguagem é através da geração das probabilidades de uma palavra ser utilizada após uma sequência (Figura 8). Ou seja, textos gerados por esses modelos computam uma palavra por vez, sempre encontrando a palavra mais provável de continuar o texto. (ZHU et al., 2023)

A Figura 9 representa a arquitetura geral de modelos baseados na arquitetura *Transformer*. Primeiramente, toda palavra do texto é *tokenizada*, utilizando diversas técnicas, como a stemização e lematização, comentadas anteriormente. Para simplificação, será definido que cada *token* é uma palavra. Em seguida, são mapeados para *embeddings*, através de um modelo de *embeddings*, genérico para o *token*, sem analisar o contexto inserido. Essa lista de *embeddings* é então passada de forma paralelizada à camada de atenção, que identifica, para cada palavra vizinha, qual a influência realiza para a palavra alvo. Esse cálculo é acumulado em um vetor que, somado ao *embedding* inicial, ajusta-o para uma posição mais relacionada com o real significado daquele *token* no contexto. Porém, todo esse ajuste é feito com os *embeddings* anteriores que também foram recém-ajustados, por isso diversas camadas de atenção são adicionadas na rede, para iterativamente se ajustarem ao ponto ideal. Entre cada uma dessas camadas de atenção existe uma multicamada de *perceptrons*, que realizam transformações individuais a cada *embedding*, sem analisar os demais. Ao fim, espera-se que toda a informação do texto foi atribuída ao último vetor, que, quando passado por uma camada *softmax* indica o *token* seguinte de maior probabilidade.

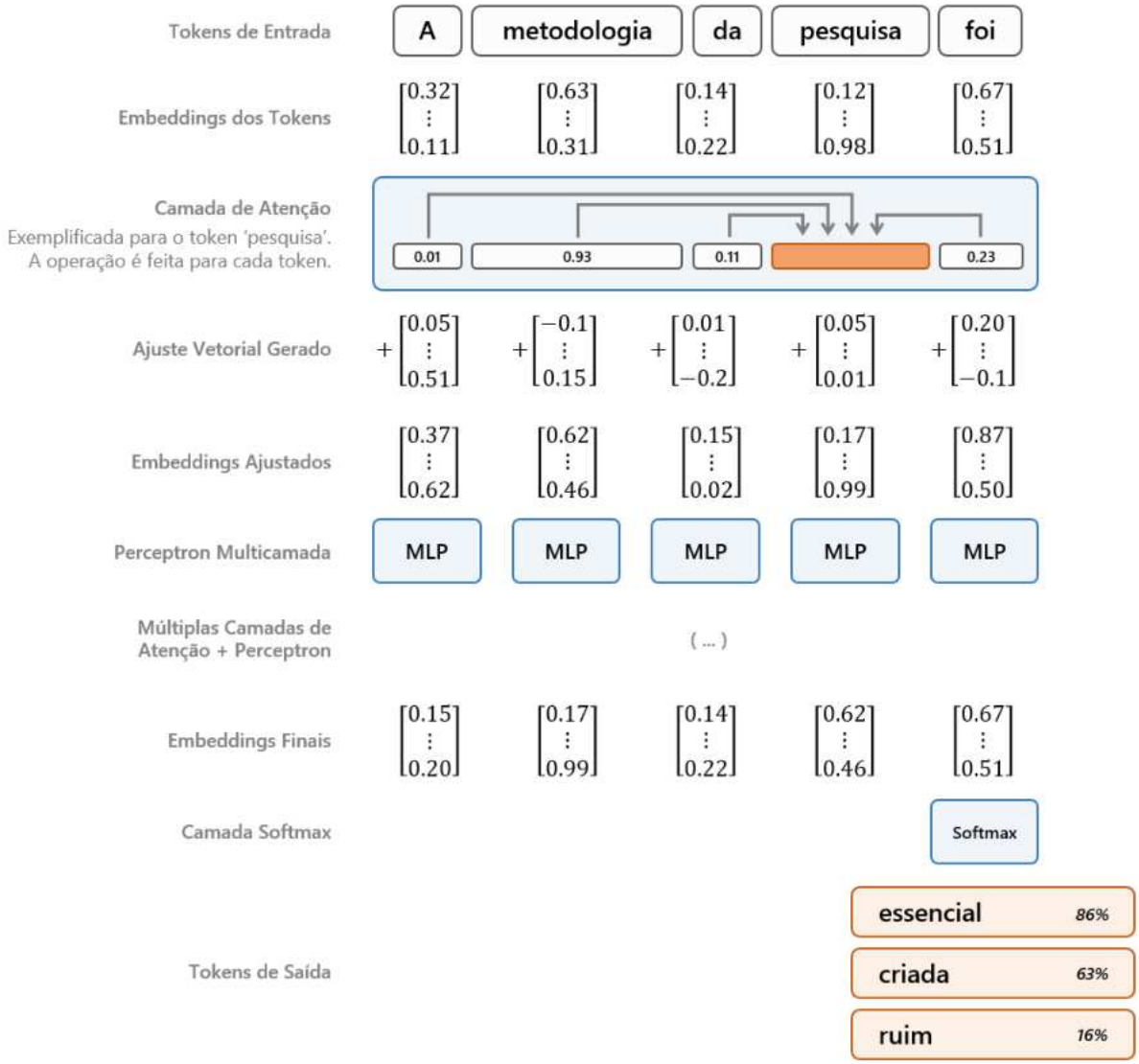

Figura 9 – Arquitetura do GPT

2.4.1 Contexto Brasileiro

Dentro do contexto brasileiro, não possuímos tantas referências de modelos treinados para o português como para outras línguas. Isso traz diversas dificuldades no cenário de uso desses modelos para a construção de sistemas de Recuperação da Informação.

Em uma tentativa de democratizar o acesso a tais modelos, projetos como o BLOOM (SCAO et al., 2023) realizam o treinamento e construção de modelos multilíngues, de países com menos recursos, e em código aberto, para incentivar o progresso nesse campo.

Existem também trabalhos brasileiros, como o Sábia (ALMEIDA et al., 2024) e o Bode (GARCIA et al., 2024), cujo objetivo é utilizar os grandes modelos pré-treinados em inglês, como o GPT e Llama (TOUVRON et al., 2023), e realizar um treinamento de refinamento para o português brasileiro, através de vastas quantidades de texto em português, gastando um valor muito inferior ao gasto na etapa de pré-treinamento.

2.4.2 Prompting

Em uma significativa mudança de paradigma, introduzida em 2020 pela OpenAI, o lançamento do GPT-3 representou uma nova forma de interação para os modelos de linguagem (BROWN et al., 2020). Diferente da abordagem tradicional de treinamento, em que o modelo aprende através de muitos exemplos de entrada e saída, o GPT-3 foi projetado para realizar uma variedade de tarefas com pouco ou nenhum exemplo prévio, apenas com base nas instruções textuais fornecidas.

Essa abordagem de descrever detalhadamente uma tarefa em linguagem natural, que o modelo usa como ponto de partida para gerar texto de maneira coerente e contextualizada, é conhecida como *prompting*. (BROWN et al., 2020)

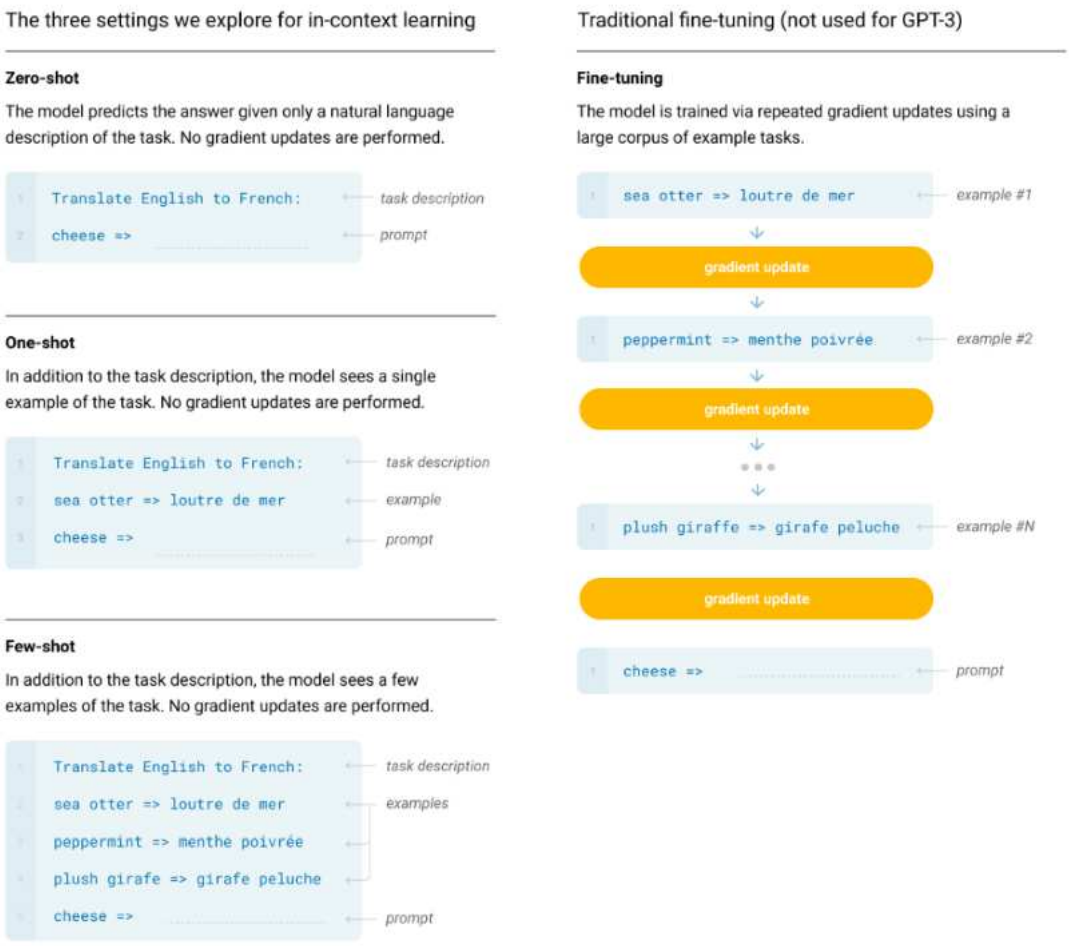

Figura 10 – Comparação de Treinamento Tradicional e One/Few/Zero Shots de LLM (BROWN et al., 2020)

2.4.3 One Shot, Few Shot e Zero Shot

A utilização dos conceitos de *One-Shot, Few-Shot e Zero-Shot Learning* é fundamental no treinamento e aplicação dos LLMs. Esses termos referem-se à quantidade de exemplos

necessários para que um modelo execute uma determinada tarefa utilizando *prompting*, em que *"shot"* vem de "tiro".

A Figura 10 representa cada uma das técnicas de treinamento. No *one-shot learning*, um único exemplo é fornecido para guiar o modelo em uma tarefa específica, como a classificação, orientando-o sobre o formato desejado de entrada e saída. No *few-shot learning*, o modelo recebe múltiplos exemplos para auxiliar sua compreensão e execução da tarefa. Já no *zero-shot learning*, o modelo é capaz de realizar a tarefa sem nenhum exemplo prévio, apenas utilizando seu treinamento inicial e sua capacidade de inferência (BROWN et al., 2020).

No presente estudo usaremos apenas *zero-shot learning* para interagir com os modelos. Isso porque será usado com o objetivo de sumarização, então não será necessário especificar como a saída é desejada.

2.4.4 Limitações dos LLMs

Apesar de serem um enorme avanço para a área de PLN, com uma profunda capacidade de compreensão e geração automatizada de texto, os Grandes Modelos de Linguagem possuem muitas limitações técnicas e de usabilidade.

Falta de Raciocínio. Apesar de muito utilizados para chegar a conclusões, realizar contas ou resolver problemas lógicos, modelos de linguagem não foram criados para esse propósito e podem gerar respostas incorretas. Algumas técnicas mais recentes, como o *chain-of-thought* (WEI et al., 2022) tentam provocar a criação de um raciocínio lógico a partir de iterações com LLMs, mas por padrão os modelos não possuem essa capacidade.

Criatividade Restrita. Outro problema de uso dos LLMs é a falsa impressão de criatividade que possuem. Toda a "criatividade" dos modelos está restrita ao vasto conhecimento que interpretou em seu treinamento.

Falta de Dados Atualizados. Os modelos de linguagem geram textos a partir do conhecimento implícito adquirido no treinamento, o que define um prazo de validade para a atualização das informações que possui conhecimento. Arquiteturas como a do *Retrieval-Augmented Generation* tentam resolver esse problema alimentando o modelo com informações recuperadas em bases externas, mais atualizadas (LI et al., 2022).

Fenômeno das "Alucinações". A forma probabilística e assertiva de resposta atrelada ao fator de falta de atualização dos dados pode gerar respostas com fatos inventados (JI et al., 2023). Arquiteturas, como a do tópico anterior, que complementam a entrada dos LLMs com dados reais também ajudam a resolver as "alucinações".

Limite do Contexto. Todo modelo possui uma entrada com tamanho limitado, que chama-se contexto (LIU et al., 2024). Da mesma forma, a saída dos modelos também

possui tamanho limitado. Isso restringe a quantidade de informação que um modelo pode receber e responder.

Fenômeno de "Perda no Meio". Conhecido como *Lost in the Middle*, em inglês, pesquisas recentes a respeito das consequências de modelos com longo contexto mostram que o conteúdo no meio de longos *prompts* pode ser esquecido (LIU et al., 2024).

2.5 GERAÇÃO AUMENTADA POR RECUPERAÇÃO

A popularidade dos LLMs, como o *ChatGPT*, desencadeou uma ampla exploração de quais áreas poderiam se beneficiar da tecnologia para desafios complexos de automatizar (BROWN et al., 2020). Muito disso, se deu pela facilidade e naturalidade da interação com os modelos, através do *prompt*, dando a falsa impressão ao usuário de conhecimento e raciocínio. Um dos usos mais comuns para esses modelos são sistemas de pergunta e resposta, aproveitando-se da capacidade de armazenamento de vasta quantidade de conhecimento em memória paramétrica. No entanto, limitações como a desatualização de informações e a geração de respostas imprecisas ou fictícias evidenciaram as fragilidades dessa abordagem para consultas baseadas em fatos (JI et al., 2023).

A Geração Aumentada por Recuperação, ou no inglês *Retrieval-Augmented Generation* (RAG), surgiu do desafio de usar os LLMs para tarefas altamente dependentes de conhecimentos gerais (LEWIS et al., 2020). Sua arquitetura combina as forças da recuperação de informação tradicional (através da busca de fatos relevantes) e dos modelos de linguagem generativos (capazes de gerar texto em linguagem natural a partir de um contexto), permitindo que os sistemas possam recuperar informações relevantes antes de responder, passando aos modelos de linguagem essas informações. Essa união permitiu respostas com maior acurácia, redução nas "alucinações", melhor relevância contextual e performance, quando comparado com sistemas apenas de LLMs (GAO et al., 2023).

Em uma visão geral, a literatura separa as etapas dos sistemas RAG da seguinte maneira (ZHU et al., 2023; LI et al., 2022; GAO et al., 2023):

1. *Fase de Pré-processamento*:

    - *Processamento de Documentos* converte os documentos brutos em dados a nível individual da informação, geralmente por meio de análise de texto e divisão em *chunks*, que são seções coesas de texto que encapsulam uma informação específica. Esse processo é altamente específico para cada domínio de conhecimento. É importante garantir que as partes não sejam nem muito pequenas (para preservar a completude da informação) e nem muito grandes (para manter cada informação individualizada e encapsulada), a fim de melhorar a recuperação.
    - *Consolidação do Banco de Dados* integra as partes processadas de informação em um banco de dados otimizado para recuperação. Pode ser utilizado um

banco de dados de documentos tradicional, que depende de correspondências exatas de termos, ou mais comumente, um banco de dados vetorial.

2. *Fase em Tempo Real*:

   - *Recuperação de Informação* busca os documentos mais relevantes da base de dados em relação a consulta do usuário. Isso pode ser alcançado por métodos de busca tradicionais ou vetoriais, a depender da base de dados. Um aspecto crucial dessa etapa é o *reranking* de documentos para garantir a relevância dos documentos recuperados (ZHU et al., 2023).
   - *Geração de Respostas* emprega LLMs para sintetizar a informação recuperada de forma sucinta, abordando a consulta do usuário e oferecendo *insights* sobre sua possível relevância para a pesquisa.

2.6 CONCLUSÃO

Neste capítulo foram apresentados conceitos importantes para culminar no entendimento do paradigma da Geração Aumentada por Recuperação e na arquitetura que será proposta em próximos capítulos dessa monografia. A contextualização contemplou desde a parte teórica da área de Recuperação da Informação até avançadas tecnologias dos Grandes Modelos de Linguagem. Em seguida, serão abordados os trabalhos relacionados nesse escopo da RAG e focalizando em seu uso para literatura científica.

3 TRABALHOS RELACIONADOS

A Revisão Automática de Literatura é um tópico de longa data, mas ganhou força recentemente com o surgimento dos LLMs (BROWN et al., 2020). Para tarefas com intenso uso de conhecimentos gerais, os LLMs demonstraram problemas de "alucinação" e conhecimento desatualizado, o que motivou o desenvolvimento da Geração Aumentada por Recuperação (LEWIS et al., 2020). Essa solução permitiu a geração de resumos automatizados suportado por fatos, aumentando a veracidade e confiabilidade dos resultados.

Esse paradigma evoluiu rapidamente tanto na otimização dos resultados da componente de recuperação quanto na geração automática de texto. Os principais avanços foram: roteamento de consultas (GAO et al., 2023) em que diferentes arquiteturas podem ser utilizadas a depender do tipo de consulta feita, por exemplo, a escolha de qual tipo de busca utilizar; reescrita de consultas (MA et al., 2023) para melhorar a qualidade do texto escrito pelo usuário, corrigir erros e deixar mais claro o direcionamento para realizar a busca; expansão de consultas (WANG; YANG; WEI, 2023), para gerar mais consultas ou destilar em subconsultas a partir da original, permitindo recuperar temas adjacentes que possam ser relevantes; *RAG-fusion* (RACKAUCKAS, 2024), para gerar consultas semelhantes usando LLM; o *reranking* de documentos (GAO et al., 2023) que ordena a relevância dos documentos selecionados através de modelos de linguagens e demonstram uma melhora significativa nos resultados finais; a seleção/compressão de contexto (GAO et al., 2023) que filtra os resultados menos relevantes para reduzir o ruído nos documentos selecionados para geração; e o *chain-of-thought* (WEI et al., 2022) para aprimorar a capacidade de raciocínio e processo de decisão dos LLMs.

Alguns outros estudos tentaram reinventar o fluxo proposto no RAG, como o *Forward-Looking Active Retrieval-Augmented Generation* (FLARE) (JIANG et al., 2023), que decide quando e o que recuperar durante a geração, ao invés de recuperar tudo antes dela como no RAG; e a Recuperação-Geração Iterativa (SHAO et al., 2023), que gera e recupera texto iterativamente até que a resposta final seja refinada.

Apesar de ser uma arquitetura recente, já existem trabalhos de uso da Geração Aumentada por Recuperação para automação da revisão de literatura. Uma primeira aplicação apresentada é responder perguntas científicas de maneira automatizada, se baseando em documentos da literatura científica. Essa interação é chamada *Question-Answering*. Em (LÁLA et al., 2023), é criada uma arquitetura inspirada no paradigma RAG de três etapas: buscar resultados, reunir evidências e responder a pergunta. Sua busca se baseia em *embeddings* e na busca semântica vetorial, assim como o presente trabalho, mas não utiliza uma etapa de reranking, o que pode reduzir drasticamente a relevância do resultado final. Após coletar os trechos de documentos mais relevantes para a pergunta, utiliza um modelo

de linguagem para consolidar a resposta. O presente trabalho realiza um fluxo similar, mas sem o objetivo de responder perguntas, mas realizar um levantamento bibliográfico e consolidar as informações de cada documento relevante ao pesquisador.

Outro foco comum é na sumarização dos documentos científicos. Em (LI et al., 2024), apesar de se inspirar na arquitetura RAG, seu propósito é contrário ao desse trabalho, parte do princípio que o pesquisador realiza o levantamento bibliográfico e a seleção dos documentos relevantes, e foca em gerar o texto da revisão de literatura que será utilizado no artigo final. Entendemos que a parte mais árdua de uma revisão de literatura é encontrar os documentos relevantes através do levantamento bibliográfico e por isso esse é o foco do presente estudo. Já em (AGARWAL et al., 2024), o objetivo foi utilizar o RAG para capturar os fatos mais relevantes das pesquisas e sumarizar em um único texto o resumo da literatura atual sobre o tema. Sua arquitetura é semelhante, com etapas de recuperação de documentos, de reranking, e geração de texto. Mas não utiliza *embeddings* na busca, trocando por uma API do *Semantic Scholar*[1]. Além disso, seu foco é realizar a sumarização geral da literatura, diferente da presente pesquisa, que busca auxiliar no processo de levantamento bibliográfico, oferecendo sumarizações individualizadas para cada um dos artigos relevantes e dando ao pesquisador mais informações para realizar sua própria seleção.

Indo além dos usos mais gerais, nos chamou atenção como a área da medicina e biológicas se sobressaíram em relação a quantidade de pesquisas publicadas aplicando o RAG e LLMs para absorver a literatura científica em seus processos. Alguns casos são o uso dessas tecnologias para melhorar o raciocínio e explicabilidade médica através de resposta a perguntas (JEONG et al., 2024), orientar na recomendação de tratamentos (ZAKKA et al., 2024), gerar textos de biomedicina suportados pela literatura científica (FRISONI et al., 2022) e melhorar as predições clínicas através da literatura (NAIK et al., 2022). Em todos os casos, uma etapa é responsável pela recuperação de documentos ou fatos relevantes e outra pela sumarização do conteúdo. Mas em todos os casos a sumarização é geral, e não individualizada por documento original.

Outro tópico em ascensão são as metodologias para validar sistemas RAG, visto que o paradigma é novo e nenhum padrão-ouro foi definido, bem dito por (XIONG et al., 2024). Alguns *frameworks*, como o RAGAS (ES et al., 2023) e o ARES (SAAD-FALCON et al., 2023), são propostos para atender a essa necessidade e serão usados como inspiração para a validação deste trabalho. O principal foco desses trabalhos é determinar métricas fortes para validação e a criação de uma metodologia de geração de bases de testes rotuladas de forma automatizada.

[1] https://www.semanticscholar.org/product/api

Este trabalho propõe utilizar a arquitetura RAG com otimização de *reranking* e seleção/compressão de contexto, em que a partir da consulta com o tópico de pesquisa do usuário, recupere as pesquisas mais relevantes da literatura em português através da busca semântica vetorial (utilizando *embeddings*), e resuma pontos chave de cada documento, fornecendo uma visão completa dos objetivos, conclusões, principais descobertas e uma explicação da relevância de cada documento em relação a consulta de interesse. O objetivo é que essas informações ajudem o pesquisador a selecionar e filtrar em uma fração do tempo os mais adequados para sua pesquisa.

Uma declaração importante é que o sistema deste trabalho não foi utilizado para fazer este levantamento bibliográfico, pois o sistema é uma prova de conceito com uma base de documentos reduzida que poderá ser expandida futuramente. Porém, as ferramentas apresentadas no Capítulo 1 foram utilizadas para encontrar os documentos a fim de testar e compreender seu uso.

4 METODOLOGIA

O presente trabalho propõe utilizar a arquitetura padrão de Geração Aumentada por Recuperação para ajudar na problemática de automatização do levantamento bibliográfico para revisões de literatura, garantindo factualidade e escalabilidade nos resultados. A arquitetura apresentada na Figura 11 buscou simplificar o processo de encontrar, analisar e extrair ideias sobre quais artigos são relevantes e como podem ser utilizado para um novo tópico de pesquisa. Nas próximas seções cada etapa da arquitetura será apresentada.

O sistema foi desenvolvido em Python e pode ser encontrado no Github[1].

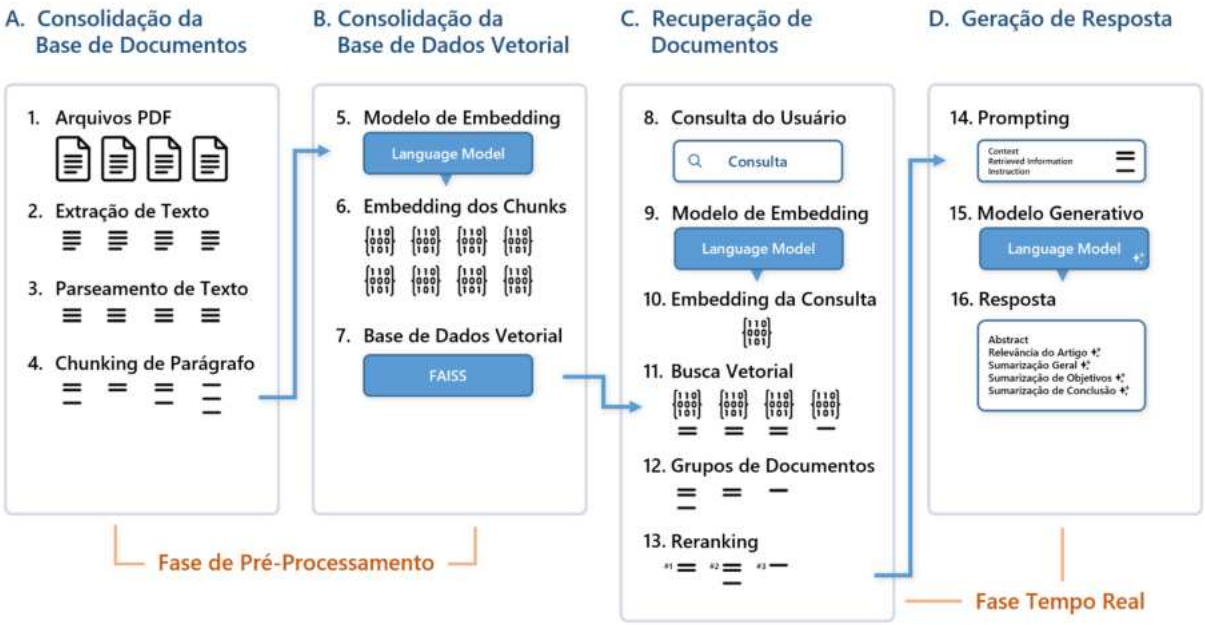

Figura 11 – Arquitetura RAG Proposta

4.1 CONSOLIDAÇÃO DA BASE DE DOCUMENTOS

Nesta primeira fase, apenas pré-processamento dos documentos é feito. Inicia-se com a aquisição dos documentos em formato PDF, que precisam ter seu texto extraído e processado para organizá-lo e corrigir imperfeições. Por fim, quebra-se o texto em blocos chamados de *chunks*, que buscam encapsular as informações individualizadas para a próxima etapa de base de dados.

4.1.1 Aquisição de Dados

O desafio inicial foi localizar uma base de dados com pesquisas relevantes em português para servir como o fundamento do sistema. Diversos repositórios de pesquisa brasileiros

[1] https://github.com/viniciuslettieri/RAG-Automated-Literature-Review/

foram avaliados, como OasisBR[2], BrCris[3], Scielo[4], UFRJ Pantheon[5], Base de Patentes da UFRJ[6] e, a escolhida, *Base Minerva*[7]. A escolha se deu pois esse repositório possui em seus metadados a *url* direta dos PDFs para todas as publicações, diferente dos demais que referenciam a página original da base, precisando de um *web scraper* especializado para cada página para obter o texto completo. Este repositório oferece uma coleção abrangente de pesquisas da Universidade Federal do Rio de Janeiro (UFRJ), como monografias, dissertações e teses de doutorado. A interface web, apresentada na Figura 12, oferece suporte a recursos avançados de recuperação de documentos no formato RIS, fornecendo metadados essenciais, como títulos, resumos, URLs para documentos PDF e detalhes adicionais. Para o estudo foram recuperados 1.539 documentos sob *Teses de Doutorado* e *Dissertações de Mestrado*, com datas de publicação entre 2020 e 2023. Uma ampla gama de temas e áreas de pesquisa foi abrangida, como engenharia, química, linguagens e outras, pois não foi aplicado nenhum filtro para áreas específicas.

Figura 12 – Base Minerva

4.1.2 Extração de Texto

Um desafio particular para o domínio da literatura científica é que grande parte da pesquisa é armazenada em PDF, necessitando de pré-processamento para consolidar a base de documentos. Para extrair o texto de arquivos PDF, foi realizado um teste abrangente de diversas ferramentas, com três técnicas principais:

- *Extração Padrão de Texto* utiliza o texto armazenado no próprio arquivo PDF, sem nenhuma técnica especial. Nossos testes utilizaram o PyPDF2[8].

[2] https://oasisbr.ibict.br
[3] https://brcris.ibict.br
[4] https://www.scielo.br
[5] https://pantheon.ufrj.br
[6] https://patentes.ufrj.br/
[7] https://minerva.ufrj.br/
[8] https://pypdf2.readthedocs.io/

- *Modelos de Aprendizado de Máquina* usam modelos de Reconhecimento Óptico de Caracteres (OCR) e Reconhecimento de Imagem, para reconhecer seções, imagens, cabeçalhos e rodapés. Nossos testes usaram o GROBID (LOPEZ, 2009).

- *Modelos Baseados em Regras* são criados por regras feitas à mão seguindo algumas premissas, como identificação de layout. Nossos testes utilizaram o PDFX (CONSTANTIN; PETTIFER; VORONKOV, 2013) e CERMINE (TKACZYK et al., 2014).

Em um teste inicial, observou-se que tanto os modelos de *aprendizado de máquina* quanto os modelos *baseados em regras* apresentam desempenho inferior na extração de texto completo de documentos em português, provavelmente devido ao treinamento predominantemente em textos em inglês. Notavelmente, apesar de o Grobid ser o padrão da indústria, muitas das vezes ignorou seções e parágrafos inteiros, por não conseguir compreender seu conteúdo. Um ponto positivo para ele foi a efetiva identificação de continuação da linha e exclusão de cabeçalhos do texto que ficam perdidas no meio do texto do PDF. Estes desafios estão alinhados com as conclusões de (MOREIRA; CUNHA, 2019), que destacou problemas semelhantes na recuperação de metadados para artigos científicos em português ao usar Grobid, Cermine e PDFX.

Em contrapartida, ferramentas mais simples como o PyPDF2 conseguem recuperar todo o texto, independentemente do idioma, apenas necessitando de um pós-processamento para ajustes de formatação. Este foi um fator importante na decisão pelo uso do PyPDF2, visto que o sistema não apresenta o texto original ao usuário, mas os utiliza para *embeddings* internos e resumos.

Na Figura 13 podemos ver o processo de extração com as duas primeira imagens, o arquivo PDF original, com o texto quebrado em duas páginas, e o texto capturado através do PyPDF2. Podemos ver que a ferramenta captura todo o texto, mas quebra as linhas sem continuidade e também adiciona o número da página do cabeçalho.

4.1.3 Parseamento do Texto

Para esta tarefa, dois métodos foram explorados: um algoritmo próprio baseado em regras e o uso de um LLM para resolver os problemas de texto da etapa de extração. Embora a abordagem do LLM tenha produzido melhores resultados, o algoritmo manual provou ser muito próximo em termos de precisão, mas significativamente mais rápido e menos custoso. Dado que a formatação perfeita não é crucial para a funcionalidade do sistema, foi escolhida a análise personalizada baseada em regras.

Durante o processo de análise, diversos filtros foram aplicados para remover informações desnecessárias, como figuras, tabelas, índice, páginas de apresentação antes do texto principal e todos os textos de referência e pós-referência. Como o PyPDF2 identifica o

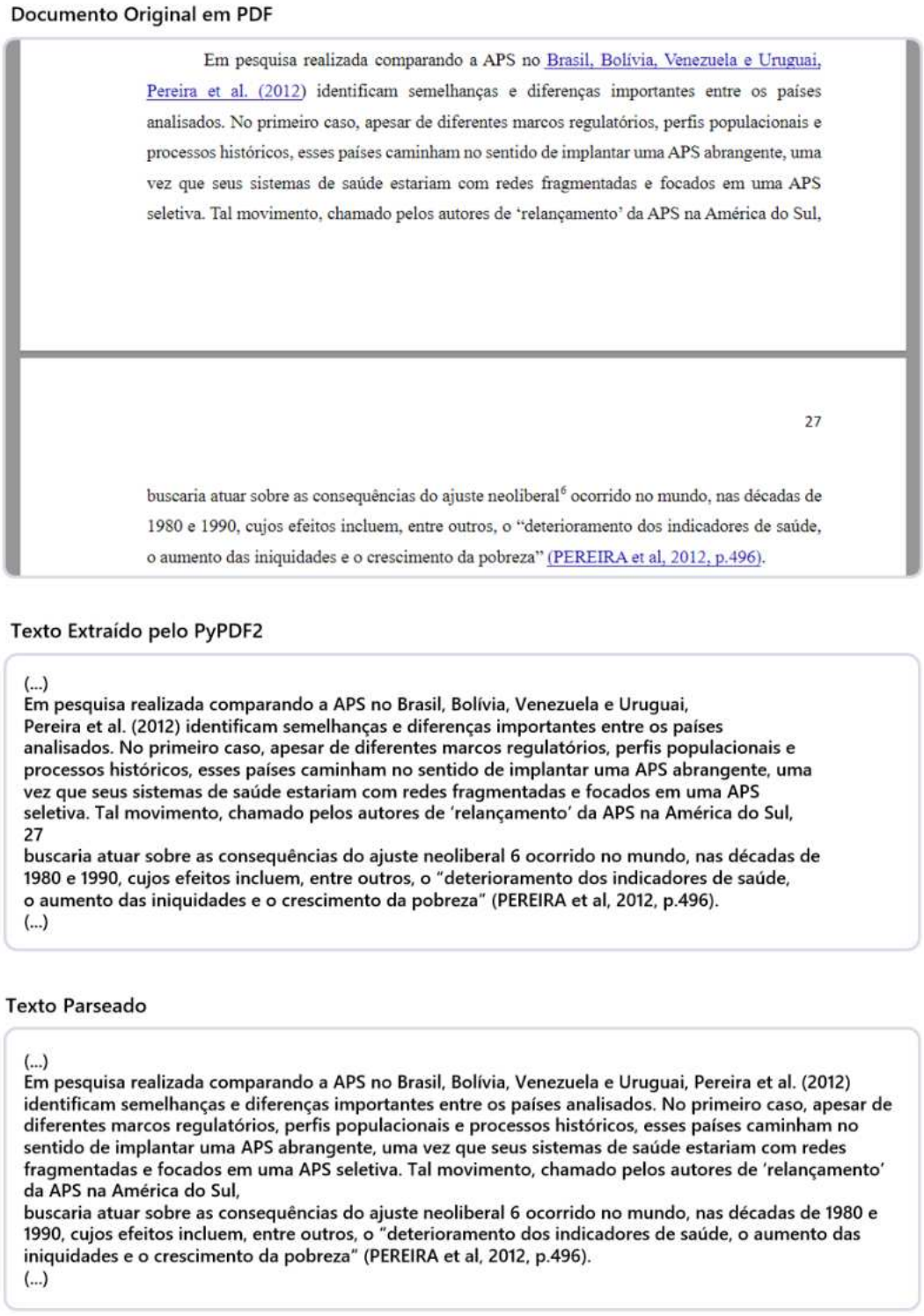

Figura 13 – Extração e Processamento de Texto

texto linha por linha, também foi necessário unir as linhas e identificar o final dos parágrafos, para dar continuidade ao texto e evitar agrupar parágrafos incorretamente. Para conseguir isso, a análise de texto baseada em sinais de pontuação e uma análise estatística do intervalo de confiança do comprimento das linhas foram empregadas para identificar e agrupar os parágrafos corretamente. Dado um bloco de texto, identificado por linhas

vazias antes e depois, obtém-se cada linha desse bloco, cuja quantidade de caracteres é x, extrai-se a média $\bar{x}$ e o desvio padrão s da quantidade de caracteres por linha. A partir da média e desvio padrão das linhas pode-se aplicar a fórmula de intervalo de confiança para classificar quais linhas possuem uma quantidade de caracteres menor que a média (nesse caso é uma linha de final de parágrafo). A Equação 4.1 é utilizada para checar quais linhas estão abaixo do intervalo de confiança com $\alpha = 0.05$ e $z = 1.65$. Em que $\bar{x}$ é a média, s o desvio padrão e N o total de linhas.

$$x \leq \bar{x} - 1.65 \cdot s/\sqrt{N} \tag{4.1}$$

Na Figura 13 podemos observar nas duas últimas imagens o parseamento realizado a partir do texto retornado pelo PyPDF2. As linhas se tornam texto contínuo e textos de cabeçalho/rodapé são identificados e excluídos do texto. O resultado final ainda possui a quebra de página, devido ao número de página que interrompeu a continuidade. Esse detalhe será tratado com os *chunkings*, que consideram a união entre parágrafos seguidos.

4.1.4 Chunking de Parágrafos

Por último, o *chunking* é muito importante para indexar os documentos no banco de dados vetorial. Cada *chunk* é uma seção do texto, dividida por um tamanho fixo, parágrafos ou por uma seção inteira. A escolha do tamanho deve equilibrar a eficiência (pedaços menores são processados mais rápido), número total de pedaços (pedaços menores aumentam o total de documentos) e, o mais importante, manter o significado da informação contida neles (pedaços menores podem dividir uma informação ao meio, enquanto pedaços maiores podem conter mais de uma informação) (LEWIS et al., 2020; GAO et al., 2023). O dimensionamento do bloco afeta o desempenho da recuperação, devido ao volume de informações individuais armazenadas.

Para recuperar informações sobre artigos científicos, presumimos que cada parágrafo representa um tópico distinto, embora os tópicos possam abranger vários parágrafos e o

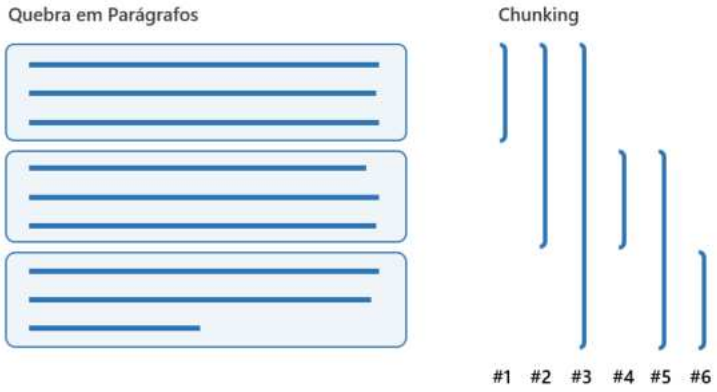

Figura 14 – Chunking de Parágrafos com N-grama

parseamento do texto possa mesclar vários parágrafos. Para garantir a continuidade no caso de informações entre parágrafos, o sistema trata cada *chunk* como uma unidade e gera vários N-gramas para toda a lista de *chunks*, variando de N=1 a N=5, resultando em *chunks* maiores. Para otimizar o desempenho e abordar muitas informações de uma só vez, considera apenas textos com 100 a 500 palavras. Esse processo está representado na Figura 14.

Com os dados consumidos, tratados e quebrados no formato individual das informações, pode-se iniciar a consolidação dessas informações na base vetorial, que será utilizada para realizar a busca semântica vetorial.

4.2 CONSOLIDAÇÃO DA BASE DE DADOS VETORIAL

Na segunda etapa do fluxo, com os *chunks* de texto criados e identificados, inicia-se o processo de indexação dessas informações. A base de dados escolhida é do tipo vetorial, que utiliza *embeddings* como o índice a ser buscado, sendo necessário um bom modelo de linguagem para geração de *embeddings*. Primeiro, cada *chunk* é transformado em um *embedding*, que é então inserido na base de dados.

4.2.1 Embedding dos Chunks

Junto com a escolha do banco de dados, é essencial obter um modelo de linguagem que capture com precisão a semântica dos documentos para mapear para *embeddings*. Também é importante adequar o idioma dos documentos a um modelo com amplo treinamento no mesmo idioma. Dada a necessidade de trabalhar com muitas línguas no contexto da pesquisa, mesmo que focado em documentos em português, os modelos multilíngues são uma ótima opção. Para esta pesquisa, o modelo de linguagem multilíngue utilizado para *embeddings* é o *paraphrase-multilingual-mpnet-base-v2* (REIMERS; GUREVYCH, 2019). Este é um modelo pré-treinado de código aberto baseado no BERT (DEVLIN et al., 2018) e refinado para a tarefa de *embedding* em múltiplas línguas. Esse modelo possui representação vetorial com 768 dimensões.

A referência ao modelo é passada diretamente na construção da base de dados, que é responsável por utilizar o modelo para gerar os índices internamente.

4.2.2 Base de Dados Vetorial

A escolha para o banco de dados foi FAISS (JOHNSON; DOUZE; JÉGOU, 2019), devido à eficiência de memória e busca de similaridade baseada em GPU por seleção de vizinho mais próximo. Os itens no banco de dados são armazenados com um índice, que é o vetor *embedding*, e um metadado relacionado a ele. Isto é útil para armazenar informações que podem ser usadas posteriormente para filtragem, como o tipo de texto

que contém e sobre o texto em si, o documento e a localização. O banco de dados não possui um modelo de linguagem nativo, sendo necessário especificar um a ele para utilizar em sua construção de índices.

A implementação local do banco de dados apresentou desafios significativos devido às suas demandas substanciais de memória. O computador usado para teste estava utilizando todos os 24 GB de memória. Consequentemente, o ajuste fino do tamanho e quantidade de chunks, realizado na fase de *Chunking de Parágrafos*, foi crucial para garantir que o sistema não sobrecarregasse a memória e ao mesmo tempo mantivesse o desempenho de recuperação de informações.

4.3 RECUPERAÇÃO DE DOCUMENTOS

A etapa de recuperação da informação inicia as operações do sistema em tempo real, com foco na seleção de informações relevantes, análise de relevância e reclassificação de documentos. O resultado é uma lista selecionada dos principais artigos científicos mais relevantes para a consulta do usuário.

4.3.1 Busca Vetorial

Na primeira etapa, usando o mesmo modelo de linguagem do banco de dados vetorial, um *embedding* da consulta do usuário é gerado, definindo uma seção no espaço vetorial que possui informações similares, que são os *chunks* de parágrafos processados nas etapas anteriores. A medida de similaridade utilizada foi a distância cosseno, mostrada na Equação 2.2. O tipo de busca utilizado é a busca vetorial densa, um tipo de busca semântica.

Com o *embedding* da consulta, a interface do banco de dados vetorial é usada para selecionar os K *chunks* de documentos mais próximos. Os parâmetros de recuperação foram definidos para 100 documentos mais próximos e posteriormente filtrados até um limite de 5 melhores pedaços por artigo. Os resultados da consulta fornecem metadados para cada *embedding* correspondente, incluindo identificadores do artigo e o texto do bloco. Isso permite agrupar os *chunks* pelos seus artigos originais, que podem ser enriquecidos com informações adicionais dos artigos, como resumos e títulos.

Importante notar que apesar do tamanho da consulta ser menor que o dos *chunks*, o resultado de similaridade não é afetado. Ao construírmos o *embedding*, a informação relevante é o conteúdo do texto.

4.3.2 Reranking dos Resultados

Embora a Busca Vetorial recupere resultados similares, o seu algoritmo é amplo demais para capturar perfeitamente a relevância de cada documento, uma vez que utiliza texto comprimido em uma dimensão baixa, via *embedding*. Essa busca é como uma rede de

pesca, que captura tudo em uma seção do mar, e que necessita de uma avaliação mais minuciosa posteriormente.

Para resolver isso, uma filtragem secundária é implementada para comparar diretamente os textos com um modelo de linguagem mais robusto e reclassificar os documentos de acordo com sua verdadeira relevância. Foi usada a API de reclassificação da Cohere[9], que processa os documentos junto com a consulta e gera uma lista de documentos classificados por relevância através de um modelo de linguagem.

4.4 GERAÇÃO DE RESPOSTA

Para cada um dos documentos retornados, agrupadas com título, resumo (*abstract*) e os *chunks* mais relevantes, quatro resumos automáticos são gerados usando um LLM: resumo do artigo, objetivos do artigo, conclusões do artigo e uma síntese sobre a relevância do artigo recuperado para a busca original do usuário. Para isso, um *prompt* foi gerado, que unifica todas essas informações, explica para o modelo sua tarefa de sumarização e requere um retorno no formato json com os 4 resumos anteriores. A seguir está o modelo de *prompt*, onde as informações entre colchetes são substituídas pelas coletadas:

```
""" Um usuário está trabalhando em uma pesquisa científica e fez a
seguinte consulta no sistema de recuperação de informação:
"{query}"
A partir da consulta foram recuperados trechos e o resumo do seguinte
artigo. A seguir, será passado o titulo, resumo e em seguida os trechos.

[TITULO] {titulo}
[RESUMO] {resumo}
[TRECHO] {trecho1}
[TRECHO] {trecho2}
[TRECHO] {trecho3}
[TRECHO] {trecho4}

Trechos Finalizados.
Com essas informações, responda do artigo recuperado em relação
a consulta realizada. Sua resposta será no formato JSON com 3
propriedades que respondem:
1. "sumarizacao": Sumarize o artigo recuperado, trazendo
informações relevantes, como o tema de pesquisa, as descobertas,
objetivos, etc. (máximo 200 palavras)
2. "objetivos": Sumarize os objetivos do artigo. (máximo 100 palavras)
```

[9] https://cohere.com/

```
3. "conclusoes": Sumarize as conclusoes do artigo. (máximo 100 palavras)
4. "relevancia": Qual o grau de relevância do artigo para a
revisão de literatura do artigo do usuário (altamente relevante,
relevante, pouco relevante ou nada relevante)? Se não for relevante,
responda apenas "Não é relevante para a pesquisa". Se for, explique
a relevância do artigo para a consulta, como seu uso para contribuir
com a pesquisa do usuário. (máximo 200 palavras) """
```

A criação deste *prompt* foi baseada na técnica de *prompt engineering* do manual da OpenAI (OPENAI, 2024), de como criar textos de entrada ótimos para o modelo, e fruto de muitos testes executados.

Para a geração da resposta foi utilizada a API OpenAI com o modelo *gpt-3.5-turbo-16k*, que possui uma janela de contexto de 16000 tokens, ou seja, o total de tokens que podem ser processados pelo modelo em uma execução. A temperatura do modelo, uma medida da criatividade do modelo, foi definida em 0.2. Isso torna as respostas do modelo mais factuais e diretas.

Por fim, para consolidar todo o processo de consolidação dos dados, das bases de dados, da construção do sistema de recuperação e da geração automatizada de resumos, a Figura 15 demonstra um caso obtido pelo sistema através de uma interação real. Nela, fazemos uma solicitação por um tema que sabemos que possui artigos dentro do extrato de documentos que foi recuperado, visto que é uma base limitada. O resultado é o documento mais relevante com cada um dos resumos, que ajudaria um pesquisador a compreender melhor sobre este artigo dentro do contexto buscado e facilitar na decisão de utilizá-lo ou não para a revisão da literatura.

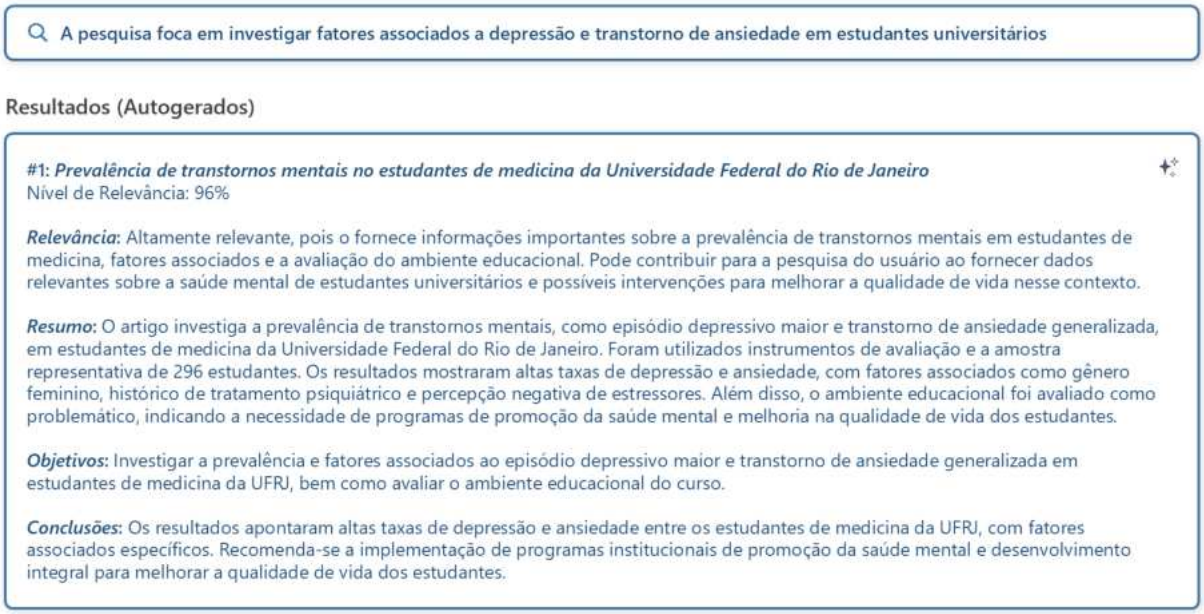

Figura 15 – Exemplo de Resultado Real Retornado pelo Sistema

5 VALIDAÇÃO DO SISTEMA

A validação de sistemas RAG requer testes tanto por componente quanto de ponta a ponta, dada a dependência da arquitetura por vários componentes distintos. Um desafio significativo no teste destes modelos é a criação de anotações feitas à mão para os casos de teste, permitindo extrair métricas. No entanto, alguns *frameworks* como RAGAS (ES et al., 2023) oferecem geração de dados de teste sintéticos, o que permite testes mais rápidos e completos. Inspirado pelo *framework*, nesta pesquisa serão utilizados os próprios documentos das pesquisas, submetidos à sumarização por um LLM para utilizar o resultado como a consulta dos testes, permitindo comparativo. Um detalhe é que essa abordagem permite que comparemos um documento de origem com o resultado esperado dele mesmo. Visto que a base utilizada não foi previamente rotulada com a similaridade entre documentos, não conseguimos realizar a comparação de um para muitos.

Para esclarecer a terminologia utilizada neste capítulo e comum na literatura para validação de sistemas RAG: *contextos* denota os *chunks* de texto recuperados; *ground truth* é o contexto original do qual os pedaços são extraídos; e *resposta* representa o texto final gerado pelo sistema.

5.1 METODOLOGIA DE TESTAGEM

O *pipeline* para teste é mostrado na Figura 16. Primeiramente, 1000 documentos são selecionados de maneira aleatória, dentro da base original, como *ground truth* para serem a base das consultas de teste. Para criar consultas, utiliza-se o LLM *gpt-3.5-turbo-0125* para extrair os objetivos iniciais de cada resumo do documento, com no máximo 80 palavras. A seguir, está o modelo de *prompt* utilizado para a tarefa:

```
""" Você é um pesquisador que está buscando realizar uma revisão de
literatura sobre a nova pesquisa que está desenvolvendo.
Esse é um resumo da nova pesquisa:
"{abstract}"

Instruções:
Identifique qual o objetivo da pesquisa e resuma.
Atenção, não são relevantes conclusões, descobertas ou dizer o que
foi realizado, queremos apenas o objetivo inicial da pesquisa.
Retorne o resumo do objetivo com 40 a 80 palavras: """
```

Posteriormente, cada consulta é alimentada no componente de recuperação, que retorna os *chunks* agrupados por cada documento, em ordem de similaridade. Esses resul-

Figura 16 – Pipeline de Testagem

tados são então passados pelo *reranker*, que organiza os documentos com base em sua relevância para a consulta. Para a análise, apenas os 10 principais documentos da busca vetorial e do *reranker* são salvos separadamente.

A partir dos resultados reordenados por relevância, o componente de geração é acionado com os 5 principais documentos para produzir, para cada artigo, um resumo geral, um resumo dos objetivos e conclusões, e a análise de relevância deste documento em relação à consulta, como visto no exemplo da Figura 15.

O resultado de cada etapa foi registrado para referência e análise futuras. A máquina utilizada para testes possui configuração: CPU AMD Ryzen 5 5600X, GPU NVIDIA GeForce RTX 3050 e 24 GB de RAM.

5.2 VALIDAÇÃO POR COMPONENTE

É essencial testar os componentes de recuperação e geração individualmente, para avaliar o desempenho e ajustar seus parâmetros. No entanto, isto deve ser feito com cautela, pois otimizações prematuras podem degradar a eficácia do sistema.

5.2.1 Recuperação

Para o componente de recuperação, tanto a busca vetorial quanto a reclassificação são realizadas, e as seguintes métricas serão aplicadas para ambos separadamente:

- *Mean Rank (MR)* avalia a posição média na qual o documento original esperado é recuperado. A classificação começa na posição 1. Para cada teste, a posição do documento esperado é obtido e a média de todos os testes é feita.

$$\text{Mean Rank} = \frac{1}{N}\sum_{i}^{N} Rank(query_i, doc_i) \tag{5.1}$$

- *Mean Reciprocal Rank (MRR)* (GAO et al., 2023) é semelhante ao *Mean Rank*, mas medida pelo inverso da classificação. A métrica é dada pela média de todos os inversos dos *ranks* de cada consulta.

$$\text{Mean Reciprocal Rank} = \frac{1}{N}\sum_{i}^{N}\frac{1}{Rank(query_i, doc_i)} \tag{5.2}$$

- *Precision@K (P@K)* (MANNING, 2008) avalia a proporção de documentos relevantes recuperados sobre todos os documentos recuperados, usando os K elementos principais na classificação. Mede a relevância dos documentos selecionados.

$$\text{Precision@K} = \frac{|\text{Documentos Relevantes}|}{|\text{Top-K Documentos}|} \tag{5.3}$$

- *Context Relevance (CR)* (ES et al., 2023) mede a utilidade dos contextos recuperados em relação à consulta. Cada contexto é comparado com a consulta por meio de um LLM para verificar sua relevância. Durante o uso do *reranker*, essa medida de relevância já é retornada, então ela será a utilizada. A pontuação final é a média da pontuação de relevância dos documentos.

$$\text{Context Relevance} = \frac{1}{N}\sum_{i}^{N} RelevanceScore(consulta_i, contextos_i) \tag{5.4}$$

Algumas métricas comuns como *Recall@K e F1* (MANNING, 2008) não serão utilizadas na pesquisa, e o *Precision@K* será limitado a K=1. Esta decisão deve-se ao fato de o conjunto de dados original das consultas não estar rotulado, então não existe uma forma tangível de decidir quais os documentos que seriam relevantes para a pesquisa para além do documento que gerou a própria consulta. Isto não afeta a validação geral do modelo, uma vez que as métricas para o primeiro lugar ainda são as mais relevantes. Uma forma possível de contornar isso seria utilizar a busca vetorial antes da criação da consulta, para descobrir quais seriam as publicações mais próximas da original. Mas isso nos daria métricas enviesadas, visto que o próprio modelo que está sendo metrificado utiliza a mesma busca vetorial que seria utilizada anteriormente.

5.2.2 Geração

Para o componente de geração, são definidas duas métricas:

- *Answer Relevance (AR)* (ES et al., 2023) avalia a pertinência das respostas geradas para a consulta, independentemente da precisão factual. Determinada pela média de todas as pontuações de similaridade cosseno entre ambos os *embeddings*. Pelos testes realizados, boas pontuações são menores que 1, e menos que 0.5 seria excelente, com o alvo ideal 0.

$$\text{AR} = \frac{1}{N}\sum_{i}^{N}\text{Sim. Cosseno} = \frac{1}{N}\sum_{i}^{N}\frac{E(Consulta_i)\cdot E(Resposta_i)}{\left\|E(Consulta_i)\right\|\left\|E(Resposta_i)\right\|} \tag{5.5}$$

- *Faithfulness (F)* (ES et al., 2023) mede o quão factualmente precisos são os resumos gerados em relação aos contextos recuperados. Tanto as respostas quanto os contextos são avaliados por um LLM para verificar a veracidade das informações fornecidas. A medida final é fração de repostas avaliadas como factualmente baseadas nos contextos em relação ao todo.

$$\text{Faithfulness} = \frac{|\text{Respostas Factuais}|}{|\text{Respostas}|} \tag{5.6}$$

5.3 VALIDAÇÃO DE PONTA A PONTA

Como a validação de componentes já verifica métricas específicas, a validação de ponta a ponta apenas verifica a integridade total do sistema.

- *Answer Semantic Similarity (AS)* (ES et al., 2023) mede a similaridade entre os *embeddings* do *ground truth* e as respostas, tomando a média de todos os testes. Difere da Relevância da Resposta, pois esta usa o abstract original para comparação e a anterior usa a consulta. O sistema de pontuação é exatamente como o anterior, menos que 1 é bom e menos que 0.5 é excelente, com alvo 0.

$$\text{AS} = \frac{1}{N}\sum_{i}^{N} \text{Sim. Cosseno} = \frac{1}{N}\sum_{i}^{N} \frac{E(GroundTruth_i) \cdot E(Resposta_i)}{\left\|E(GroundTruth_i)\right\|\left\|E(Resposta_i)\right\|} \tag{5.7}$$

5.4 EXEMPLO DO FLUXO DE VALIDAÇÃO

Para simplificar o entendimento do fluxo, demonstraremos a execução de um caso de exemplo, seguindo o fluxo da Figura 16. Iniciamos o fluxo com um documento original, em que apresentamos seu título e resumo (*abstract*):

Título Original: Gestão de emergências na indústria de óleo e gás: uma proposta de aplicação da metodologia do Incident Command System.

Resumo/Abstract Original: Os incidentes estão presentes ao longo de toda história da indústria de óleo e gás. Por mais que os cuidados na prevenção destes tenham aumentado ao longo dos anos, grandes acidentes continuam ocorrendo. Nesse contexto, se desenvolveu a prática de gestão de emergências e uma metodologia que vem sendo bastante empregada por empresas e órgãos públicos na resposta a emergências: é o Incident Command System (ICS). Essa dissertação traz uma revisão da literatura sobre gestão de emergências e os regulamentos brasileiros abordando planejamento de emergências aplicáveis à indústria de óleo e gás. Também explica a metodologia do ICS, seu histórico

e conceitos básicos. E, por meio de dois estudos de caso de acidentes da indústria, em que foram aplicados o ICS, busca demonstrar a importância do elemento de gestão de emergências e as vantagens trazidas pela sua utilização.

A partir do resumo, gera-se uma consulta extraindo os objetivos iniciais da pesquisa através de um LLM.

Consulta Gerada: O objetivo da pesquisa é realizar uma revisão da literatura sobre gestão de emergências na indústria de óleo e gás, abordando os regulamentos brasileiros relacionados ao planejamento de emergências. Além disso, busca-se explicar a metodologia do Incident Command System (ICS), sua aplicação e vantagens por meio de estudos de caso de acidentes na indústria.

A partir da consulta, iniciamos o fluxo de teste, através do componente de recuperação (*Retriever*). Este componente realiza a busca semântica vetorial através da base de dados vetorial, recuperando diversos *chunks* de parágrafos relevantes. Agrupam-se os parágrafos de um mesmo documento para consolidar as informações e então passar no componente de *reranking*. A seguir, foram trazidos dois documentos similares recuperados, já com os *chunks* agrupados. Os textos foram reduzidos para simplificar.

Resultados do Retriever e Reranker:

1. **Artigo Similar**: Gestão de emergências na indústria de óleo e gás: uma proposta de aplicação da metodologia do Incident Command System.

   Reranker Relevance Score: 0.99

   Chunk Similar 1: Exemplificar através de dois estudos de caso as vantagens e limitações de se utilizar o ICS na gestão de emergências. Identificar potenciais melhorias na regulação voltada para a gestão de emergências. 1.3. Estrutura do Trabalho De modo a atingir os objetivos descritos na seção anterior, este trabalho foi estruturado em cinco capítulos. No Capítulo 2, será feita uma revisão da literatura sobre gestão de emergências, os regulamentos brasileiros abordando planejamento de emergências aplicáveis à indústria de óleo e gás, e, por fim, as normas e procedimentos nacionais que preveem a aplicação do ICS...

   Chunk Similar 2: O Incident Command System (ICS) foi projetado justamente com o objetivo de padronizar procedimentos e comunicações, além de definir uma estrutura organizacional, para o gerenciamento de ações e recursos destinados à resposta a emergências. No Capítulo 2, vimos que a regulamentação brasileira traz a obrigatoriedade de elaboração dos planos de resposta a emergência para diversas atividades da indústria de óleo e gás. No entanto, a aplicação do ICS só é prevista na legislação para incidentes de derramamento de óleo em águas sob jurisdição

nacional. O Plano de Auxílio Mútuo e o projeto APELL, citados no Capítulo 2, exemplificam a atuação conjunta de órgãos públicos e empresas privadas na gestão de emergências...

2. **Artigo Similar**: Aperfeiçoamento da regulação brasileira de segurança do processamento submarino de óleo e gás a partir de análise crítica e comparativa com regulações congêneres.

   Reranker Relevance Score: 0.70

   Chunk Similar 1: A norma era uma prática recomendada pelo BSEE para as operações de óleo e gás nos Estados Unidos, porém, após o acidente de Macondo, tornou-se obrigatória em 2010 (BSEE, 2013). Em 2007, o CCPS desenvolveu uma estrutura de sistema de gerenciamento de segurança baseado em risco, o qual é chamado RBPS. Essa estrutura foi fundamentada na experiência em segurança de processos da indústria química. Além disso, publicou um guia para ajudar na utilização prática desse sistema , o qual é composto por 4 pilares e 20 elementos distribuídos dentro desses pilares. Os pilares são (1) comprom isso com a segurança de processos; (2) entendimento de perigos e riscos; (3) gestão de risco; e (4) aprendizado com a experiência (CCPS, 2007)...

   Chunk Similar 2: A investigação de acidentes é amplamente reconhecida como uma parte essencial de um processo abrangente e eficiente de gestão da segurança. Uma análise detalhada e sistemática de um evento imprevisto permite identificar, não apenas a sua causa imediata (primária), como todo o conjunto de causas raízes, cuja combinação levou à falha do sistema e à ocorrência das consequências prejudiciais correspondentes (acidente grave), ou uma condição perigosa temporária não planejada (quase acidente) (ABDOLHAMIDZADEH et al., 2012). É necessário estudar o acidente além das causas imediatas, identificando pontos como falhas humanas, organizacionais e pontos fracos no sistema de gestão. Além disso, se as causas subjacentes forem encontradas, recomendações adequadas devem ser efetuadas e executadas (KLETZ, 2001)...

Podemos ver que os resultados fazem sentido. O primeiro é do próprio documento original e o segundo também aborda acidentes no cenário de óleo e gás. Dado os documentos recuperados e ordenados por relevância, inicia-se o processo de sumarização:

Resultados de Geração de Texto Automatizada:

1. **Artigo Similar**: Gestão de emergências na indústria de óleo e gás: uma proposta de aplicação da metodologia do Incident Command System.

Relevância (Autogerado): Altamente relevante. O artigo fornece uma revisão da literatura, regulamentos brasileiros e estudos de caso que abordam diretamente os temas de interesse do usuário, contribuindo significativamente para a pesquisa sobre gestão de emergências na indústria de óleo e gás, incluindo a aplicação do Incident Command System (ICS) e suas vantagens.

Sumarização (Autogerado): O artigo aborda a gestão de emergências na indústria de óleo e gás, com foco na aplicação da metodologia do Incident Command System (ICS). Realiza uma revisão da literatura sobre o tema, analisa os regulamentos brasileiros relacionados ao planejamento de emergências e apresenta dois estudos de caso que demonstram a eficácia do ICS na gestão de incidentes. Destaca a importância da integração de esforços entre órgãos públicos e empresas privadas na resposta a emergências.

Objetivos (Autogerado): Realizar uma revisão da literatura sobre gestão de emergências na indústria de óleo e gás, abordar os regulamentos brasileiros relacionados ao planejamento de emergências, explicar a metodologia do Incident Command System (ICS) e demonstrar sua aplicação e vantagens por meio de estudos de caso de acidentes na indústria.

Conclusões (Autogerado): A utilização do Incident Command System (ICS) contribuiu para uma gestão de emergências mais eficaz nos estudos de caso analisados. A integração de esforços entre órgãos públicos e empresas privadas é fundamental para uma resposta coordenada a incidentes. A escassez de referências sobre a aplicação do ICS na indústria de óleo e gás indica que sua implantação ainda está em fase de aprendizagem e aprimoramento.

2. **Artigo Similar**: Aperfeiçoamento da regulação brasileira de segurança do processamento submarino de óleo e gás a partir de análise crítica e comparativa com regulações congêneres.

   Relevância (Autogerado): Relevante. O artigo fornece uma análise crítica da regulação brasileira de segurança do processamento submarino de óleo e gás, o que pode contribuir significativamente para a pesquisa do usuário sobre gestão de emergências na indústria de óleo e gás, abordando os regulamentos brasileiros relacionados ao planejamento de emergências.

   Sumarização (Autogerado): O artigo aborda a regulação brasileira de segurança do processamento submarino de óleo e gás, comparando-a com regulações internacionais. Avalia a implementação do regulamento específico para equipamentos submarinos e propõe melhorias no arcabouço regulatório atual. Destaca a importância da investigação de acidentes para identificar causas e promover a segurança das instalações submarinas.

Objetivos (Autogerado): O artigo tem como objetivo avaliar a regulação brasileira de segurança submarina, comparando-a com regulações internacionais, analisar a implementação do regulamento específico para equipamentos submarinos e propor melhorias no arcabouço regulatório atual.

Conclusões (Autogerado): A implementação da regulação de equipamentos submarinos não foi suficiente para promover a segurança dessas instalações. São propostas melhorias no arcabouço regulatório, como simplificação, introdução de novos requisitos específicos e criação de um comitê técnico para melhorar a segurança dos equipamentos submarinos.

Como bem pontua a sumarização de relevância do segundo caso, ambos tratam de gestão de emergência a incidentes da indústria de óleo e gás. A partir do ranking de resultados e as sumarizações geradas, as métricas são consolidadas para cada um dos testes.

5.5 RESULTADOS

5.5.1 Validação de Fine-Tuning

Antes dos testes em si, um pequeno subconjunto de 100 dentre as 1000 consultas geradas na seção anterior foram selecionadas de maneira aleatória e utilizadas para ajustar o hiperparâmetro do total K de documentos recuperados através da Busca Vetorial. Essas configurações melhoram a performance e o custo do sistema. Para avaliar, as mesmas métricas utilizadas na validação foram analisadas.

Para ajustar o K para o total de documentos recuperados na Busca Vetorial, foram testados K=50, 100, 500. Este é um hiperparâmetro muito sensível, pois filtra os documentos que são passados para o *reranker*, que gera custo devido ao acesso à API. A Tabela 1 mostra os resultados tanto da Recuperação quanto do Reranker, respectivamente divididos por uma barra. Não há uma grande diferença entre nenhuma das medidas, o que permite a escolha por custo e um valor razoável para a qualidade dos resultados. O K=50 pareceu ser muito pequeno para realizar um reranking e captura de vários *chunks* por documento. Sendo assim, a opção foi pelo K=100 que balanceia as opções.

Tabela 1 – Fine-tuning do Total de Documentos Retornados na Busca Vetorial

Total	MR	MRR	P@1	CR
K=50	1.17 | 1.0	0.83 | 0.9	0.78 | 0.9	0.91
K=100	1.31 | 1.0	0.83 | 0.94	0.77 | 0.94	0.92
K=500	1.42 | 1.0	0.83 | 0.94	0.78 | 0.94	0.90

5.5.2 Validação do Componente de Recuperação

As métricas de recuperação mostradas na Tabela 2 são muito promissoras. Cada célula possui o valor da métrica para Busca Vetorial | Reranking. A recuperação com *reranking* alcançou um *Mean Rank* (MR) de 1.003, que é quase a primeira posição em todos os testes; com 9% das consultas não encontrando (NE) o documento correto entre os 5 primeiros resultados; uma excelente *Mean Reciprocal Rank* (MRR) de 0.91; uma *Precision@1* (P@1) de 0.91; e um *Context Relevance* (CR) de 0.90. Além disso, a etapa de reranking notavelmente aprimora a relevância dos documentos recuperados.

Tabela 2 – Validação do Componente de Recuperação para 1000 Consultas

MR	NE	MRR	P@1	CR
1.228 | 1.003	9% | 9%	0.85 | 0.91	0.81 | 0.91	0.65 | 0.9

Quanto ao tempo de execução da etapa de recuperação, o tempo médio de resposta para a busca vetorial foi de 0,25 segundos com desvio padrão de 0,05 e a média para o *reranker* foi de 0,64 segundos com desvio padrão de 0,84.

5.5.3 Validação do Componente de Geração

Quanto às métricas de geração mostradas na Tabela 3, a *Faithfulness* (F) alcançou 99,5%, o que garante que o gerador responde apenas com fatos do contexto. Isso é esperado, já que a temperatura dos modelos foi ajustada para 0,2, o que é menos criativo.

Como essa métrica é validada usando um LLM, um bom teste para verificar se o modelo está tendencioso a sempre responder "SIM" seria testar com uma resposta final e contextos aleatórios de outros documentos, ao invés dos contextos reais recuperados, o que deveria resultar em respostas "NÃO". Testando com 100 consultas, obteve-se 92% de precisão para respostas "NÃO".

Para a *Answer Relevance* (AR), há uma métrica para cada uma das respostas (Relevância, Resumo, Objetivos e Conclusões), todas elas obtendo pontuações muito boas na Tabela 3. A melhor similaridade de 0,15 foi para a relevância, como esperado, porque compara a consulta original com o documento. Para as outras, elas representam muito bem que os documentos recuperados são bastante semelhantes à consulta.

Tabela 3 – Validação do Componente de Geração para 1000 Consultas

F	AR (Rel)	AR (Sum)	AR (Obj)	AR (Con)
99.5%	0.15	0.19	0.27	0.27

Para o tempo de execução da etapa de geração, o tempo médio de resposta foi de 50 segundos com desvio padrão de 9,5.

5.5.4 Validação Ponta a Ponta

Quanto à *Answer Semantic Similarity* (AS), comparando o resumo original com cada resposta (Relevância, Resumo, Objetivos e Conclusões), as métricas mostradas na Tabela 4 também são muito boas, todas abaixo de 0,5. Isso mostra que o sistema completo fornece respostas muito próximas ao tópico original, mais relevantes para a busca.

É importante notar que, quando comparadas à *Answer Relevance*, essas métricas são um pouco mais altas, e isso ocorre porque a anterior usa a consulta, que é um resumo dos objetivos principais do resumo original, levando a um *embedding* mais refinado, mais próximo da resposta final.

Tabela 4 – Validação ponta a Ponta para 1000 Consultas

AS (Rel)	AS (Sum)	AS (Obj)	AS (Con)
0.22	0.23	0.35	0.30

Para concluir, vemos que as métricas apresentadas neste capítulo são muito positivas, com apenas uma passível de melhorias, a quantidade de testes em que o documento original não foi encontrado nos resultados. Isso ocorreu em 9% da base, o que pode ser analisado e melhorado. Ainda assim, todos os demais resultados demonstraram a qualidade dos resultados.

6 CONCLUSÃO

Este estudo propôs a utilização da arquitetura de Geração Aumentada por Recuperação para automatizar o levantamento bibliográfico de revisões de literatura em documentos em português. O objetivo foi aprimorar a produtividade da criação de pesquisas científicas, através de busca semântica por relevância e resumos que auxiliem pesquisadores a encontrar e filtrar mais facilmente artigos de interesse. Ao combinar as capacidades robustas de recuperação tradicional com a geração sofisticada de texto de grandes modelos de linguagem (LLMs), o sistema RAG resolve desafios de geração de conteúdo impreciso e a incapacidade de incorporar informações em tempo real, essenciais no cenário de pesquisa.

A etapa mais demorada da monografia foi a extração de texto de documentos em formato PDF. Foram testadas várias ferramentas, mas devido à falta de suporte adequado para o português, a melhor solução encontrada foi a criação de um parser de texto personalizado. Isso demostra um avanço necessário na publicação de pesquisas, garantindo formatos legíveis para máquinas. Iniciativas como a *Text Encoding Initiative*[1] podem auxiliar com formatações padronizadas que permitam melhor busca dentro de artigos.

Este desafio evidenciou um segundo problema: a ausência de bases de dados de pesquisas em português rotuladas por similaridade. A existência dessas bases é crucial para validar arquiteturas de sistemas de busca em português, que foi a dificuldade em gerar métricas além do primeiro resultado. Formalizar uma base que relacione artigos em português é essencial para futuros trabalhos, inclusive para padronizar e comparar pesquisas.

A arquitetura RAG, validada por meio de uma versão adaptada do *Framework RAGAS* e métricas da literatura, demonstrou resultados significativos tanto na recuperação quanto na geração de textos. Isso é promissor para o objetivo de simplificar a exploração de pesquisas, indicando que o sistema pode ser ainda mais ampliado e aprimorado.

Também é importante levantar o tópico de ética no uso de modelos de inteligência artificial para pesquisas científicas. Ainda que o sistema proposto não sirva para o propósito de utilizar as sumarizações diretamente no texto de uma pesquisa, os modelos são utilizados de ponta a ponta no sistema, sendo essencial entender as implicações do uso.

Primeiramente, quanto ao uso indiscriminado de resultados de modelos de linguagem, é do nosso entendimento que esse tipo de modelo ainda não foi maturado e testado suficientemente para produzir texto sem estar assistido. Em um futuro próximo, é possível que os problemas de alucinação e falta de factualidade sejam totalmente resolvidos e a produção seja tão confiável que de fato possa ser utilizada, mas no momento deve ser utilizada com cuidado. Por isso reforçamos que o sistema desenvolvido deve ser utilizado na fase de levantamento bibliográfico, não substituindo a leitura completa dos textos pós-seleção.

[1] https://tei-c.org/

Isso é especialmente preocupante para o contexto de medicina e biológicas, uma área com muita pesquisa relacionada ao uso de LLMs e RAG, como visto no Capítulo 3. Resultados recuperados de maneira errada e sumarizações não verídicas ou sem garantia de fatos possam impactar de maneiras catastróficas uma consulta, cirurgia ou pesquisa.

É importante notar que poucas são as pessoas que utilizam esse tipo de modelo com consciência de como eles são treinados e em qual função deveriam ser aplicados. Como vimos na Seção 2.4.4, ainda existem muitas limitações de uso dos modelos, e a principal é que um modelo de linguagem fundamentalmente é um gerador de probabilidades para a próxima palavra de uma frase. Não existe processo de lógica por trás que certifique ou se questione do que está sendo gerado. Os usos mais seguros para modelos de linguagem são os que o conteúdo a ser utilizado como insumo está explicitamente inserido no texto de entrada. Casos como sumarização de um texto, correção de texto, melhorar a articulação do texto e até mesmo tradução são menos problemáticos pois não dependem do conteúdo implícito do modelo.

Mesmo sem o uso da sumarização gerada, o sistema integra modelos de linguagem em cada uma das etapas, através dos *embeddings* e do *reranking* realizado. Isso poderia implicar que erros no processo de interpretação do texto realizado pelos modelos que geram os *embeddings* ou realizam o *reranking* afetariam os resultados finais ao remover algum documento relevante da listagem apresentada. Esse erro afetaria a pesquisa, devido a um enviesamento das publicações geradas por falta de exposição ao resultado. Uma possível garantia seria utilizar múltiplos modelos em paralelo, para garantir que o erro seja minimizado, e também utilizar a técnica de *exact match* junto da busca.

Quanto ao uso atual dessa tecnologia, em uma pesquisa realizada pela *Nature* com pós-doutores (NORDLING, 2023), 31% afirmou utilizar ferramentas de chat com modelos de linguagem no dia a dia, mas ainda veem um ceticismo muito alto na academia em relação ao seu uso. Um ponto importante citado é que muitos pesquisadores escrevem em línguas não-nativas, sendo esse tipo de ferramenta muito útil para melhorar o texto e corrigir erros. Em geral, os entrevistados disseram que veem nesse tipo de inteligência artificial uma grande capacidade para reduzir o trabalho braçal ou considerado penoso, que fazem em algumas etapas da pesquisa, mas no fundo o conteúdo deve ser responsabilidade do pesquisador e a ferramenta trabalhar ao redor disso.

Em resumo, o ponto mais importante dessa discussão de ética é que a inteligência artificial será inserida no contexto acadêmico em grande escala, mesmo que alguns pesquisadores sejam contra. Então é um momento de compreender a tecnologia envolvida, as aplicações que podem se desenvolver e as implicações relacionadas, para que problemas sejam corrigidos antes de entrarem em vigor e tenham qualidade garantida.

Para futuros desenvolvimentos, duas etapas podem ter um impacto substancial no desempenho geral do sistema: a *expansão de consulta* (WANG; YANG; WEI, 2023) e a

reescrita de consulta (MA et al., 2023), podendo organizar melhor as consultas, extrair mais informações e dividi-las em subconsultas, aumentando a precisão e a relevância dos resultados. Além disso, a técnica de *chain-of-thought* (WEI et al., 2022) para geração de respostas pode enriquecer as respostas ao adicionar mais etapas de raciocínio no modelo, identificando melhor os conceitos-chave para a verificação de relevância. Quanto à validação, é importante evoluir no comparativo de relevância para além de um único documento por teste e também realizar um comparativo com outras metodologias, como a própria busca por correspondência exata.

REFERÊNCIAS

AGARWAL, S. et al. Litllm: A toolkit for scientific literature review. **arXiv preprint arXiv:2402.01788**, 2024.

ALLAN, J. et al. Challenges in information retrieval and language modeling: report of a workshop held at the center for intelligent information retrieval, university of massachusetts amherst, september 2002. v. 37, n. 1, p. 31–47, 2003.

ALMEIDA, F.; XEXÉO, G. Word embeddings: A survey. **arXiv preprint arXiv:1901.09069**, 2019.

ALMEIDA, T. S. et al. Sabiá-2: A new generation of portuguese large language models. **arXiv e-prints**, p. arXiv–2403, 2024.

BAEZA-YATES, R.; RIBEIRO-NETO, B. et al. Modern information retrieval. ACM press New York, v. 463, n. 1999, 1999.

BROWN, T. et al. Language models are few-shot learners. **Advances in neural information processing systems**, v. 33, p. 1877–1901, 2020.

CHOMSKY, N. **Syntactic structures**. [S.l.: s.n.], 1957.

CONSTANTIN, A.; PETTIFER, S.; VORONKOV, A. Pdfx: fully-automated pdf-to-xml conversion of scientific literature. p. 177–180, 2013.

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

ES, S. et al. Ragas: Automated evaluation of retrieval augmented generation. **arXiv preprint arXiv:2309.15217**, 2023.

FRISONI, G. et al. Bioreader: a retrieval-enhanced text-to-text transformer for biomedical literature. p. 5770–5793, 2022.

GAO, Y. et al. Retrieval-augmented generation for large language models: A survey. **arXiv preprint arXiv:2312.10997**, 2023.

GARCIA, G. L. et al. Introducing bode: A fine-tuned large language model for portuguese prompt-based task. **arXiv preprint arXiv:2401.02909**, 2024.

HAMBARDE, K. A.; PROENCA, H. Information retrieval: recent advances and beyond. **IEEE Access**, IEEE, 2023.

HARRIS, Z. S. Distributional structure. **Word**, Taylor & Francis, v. 10, n. 2-3, p. 146–162, 1954.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, MIT press, v. 9, n. 8, p. 1735–1780, 1997.

JEONG, M. et al. Improving medical reasoning through retrieval and self-reflection with retrieval-augmented large language models. **arXiv preprint arXiv:2401.15269**, 2024.

JI, Z. et al. Survey of hallucination in natural language generation. **ACM Computing Surveys**, 2023.

JIANG, Z. et al. Active retrieval augmented generation. p. 7969–7992, 2023.

JOHNSON, J.; DOUZE, M.; JÉGOU, H. Billion-scale similarity search with gpus. **IEEE Transactions on Big Data**, IEEE, v. 7, n. 3, p. 535–547, 2019.

JONES, K. S. A statistical interpretation of term specificity and its application in retrieval. **Journal of documentation**, MCB UP Ltd, v. 28, n. 1, p. 11–21, 1972.

KOUBAA, A. Gpt-4 vs. gpt-3.5: A concise showdown. 04 2023.

LÁLA, J. et al. Paperqa: Retrieval-augmented generative agent for scientific research. **arXiv preprint arXiv:2312.07559**, 2023.

LEWIS, P. et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. **Advances in Neural Information Processing Systems**, v. 33, p. 9459–9474, 2020.

LI, H. et al. A survey on retrieval-augmented text generation. **arXiv preprint arXiv:2202.01110**, 2022.

LI, Y. et al. Chatcite: Llm agent with human workflow guidance for comparative literature summary. **arXiv preprint arXiv:2403.02574**, 2024.

LIU, N. F. et al. Lost in the middle: How language models use long contexts. **Transactions of the Association for Computational Linguistics**, 2024.

LOPEZ, P. Grobid: Combining automatic bibliographic data recognition and term extraction for scholarship publications. p. 473–474, 2009.

MA, X. et al. Query rewriting for retrieval-augmented large language models. **arXiv preprint arXiv:2305.14283**, 2023.

MANNING, C. Introduction to information retrieval. In: . [S.l.]: Cambridge University Press, 2008. cap. 8, p. 151–175.

MIKOLOV, T. et al. Efficient estimation of word representations in vector space. **arXiv preprint arXiv:1301.3781**, 2013.

MOREIRA, I.; CUNHA, M. Avaliação de ferramentas de extração automática de metadados na catalogação de artigos científicos do connepi. In: SBC. **Anais da XIX Escola Regional de Computação Bahia, Alagoas e Sergipe**. [S.l.], 2019. p. 386–395.

NAIK, A. et al. Literature-augmented clinical outcome prediction. p. 438–453, 2022.

NORDLING, L. **How ChatGPT is transforming the postdoc experience**. 2023. Accessed: 30/06/2024. Disponível em: https://www.nature.com/articles/d41586-023-03235-8.

OPENAI. **Prompt Engineering - OpenAI**. 2024. Accessed: 25/06/2024. Disponível em: https://platform.openai.com/docs/guides/prompt-engineering.

RACKAUCKAS, Z. Rag-fusion: a new take on retrieval-augmented generation. **arXiv preprint arXiv:2402.03367**, 2024.

REIMERS, N.; GUREVYCH, I. Sentence-bert: Sentence embeddings using siamese bert-networks. **arXiv preprint arXiv:1908.10084**, 2019.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **nature**, Nature Publishing Group UK London, v. 323, n. 6088, p. 533–536, 1986.

SAAD-FALCON, J. et al. Ares: An automated evaluation framework for retrieval-augmented generation systems. **arXiv preprint arXiv:2311.09476**, 2023.

SALTON, G. **Automatic Information Organization and Retrieval.** [S.l.]: McGraw Hill Text, 1968. ISBN 0070544859.

SALTON, G.; WONG, A.; YANG, C. S. A vector space model for automatic indexing. **Commun. ACM**, Association for Computing Machinery, New York, NY, USA, p. 613–620, 1975.

SCAO, T. L. et al. Bloom: A 176b-parameter open-access multilingual language model. 2023.

SHAO, Z. et al. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. p. 9248–9274, 2023.

TKACZYK, D. et al. Cermine–automatic extraction of metadata and references from scientific literature. p. 217–221, 2014.

TOUVRON, H. et al. Llama 2: Open foundation and fine-tuned chat models. **arXiv preprint arXiv:2307.09288**, 2023.

VASWANI, A. et al. Attention is all you need. **Advances in neural information processing systems**, v. 30, 2017.

WANG, L.; YANG, N.; WEI, F. Query2doc: Query expansion with large language models. p. 9414–9423, 2023.

WEI, J. et al. Chain-of-thought prompting elicits reasoning in large language models. **Advances in neural information processing systems**, v. 35, p. 24824–24837, 2022.

XIONG, G. et al. Benchmarking retrieval-augmented generation for medicine. **arXiv preprint arXiv:2402.13178**, 2024.

YAN, Z. Patterns for building llm-based systems products. **eugeneyan.com**, 2023. Disponível em: https://eugeneyan.com/writing/llm-patterns/.

ZAKKA, C. et al. Almanac—retrieval-augmented language models for clinical medicine. **NEJM AI**, Massachusetts Medical Society, v. 1, n. 2, p. AIoa2300068, 2024.

ZHU, Y. et al. Large language models for information retrieval: A survey. **arXiv preprint arXiv:2308.07107**, 2023.