

Modelos de Linguagem para Recuperação e Geração de Conhecimento Contextualizado

Aplicação de Retrieval-Augmented Generation (RAG)
em Reações Químicas e Análise de Dados Científicos

Jonas Gomes da Silva Junior



UFG

UNIVERSIDADE
FEDERAL DE GOIÁS

UNIVERSIDADE FEDERAL DE GOIÁS (UFG)
INSTITUTO DE INFORMÁTICA (INF)

JONAS GOMES DA SILVA JUNIOR

**Modelos de Linguagem para Recuperação e Geração de
Conhecimento Contextualizado**

Aplicação de Retrieval-Augmented Generation (RAG) em Reações Químicas e
Análise de Dados Científicos

Goiânia
2025



UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC no 1240/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei no 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo dos Trabalhos de Conclusão dos Cursos de Graduação disponibilizado no RI/UFG é de responsabilidade exclusiva dos autores. Ao encaminhar(em) o produto final, o(s) autor(a)(es)(as) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG)

Nome(s) completo(s) do(a)(s) autor(a)(es)(as): JONAS GOMES DA SILVA JUNIOR

Título do trabalho: Modelos de Linguagem para Recuperação e Geração de Conhecimento Contextualizado

Aplicação de Retrieval-Augmented Generation (RAG) em Reações Químicas e Análise de Dados Científicos

2. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador) Concorda com a liberação total do documento [X] SIM [] NÃO¹

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante: a) consulta ao(à)(s) autor(a)(es)(as) e ao(à) orientador(a); b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo do TCCG. O documento não será disponibilizado durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro.

Obs.: Este termo deve ser assinado no SEI pelo orientador e pelo autor.



Documento assinado eletronicamente por **Jonas Gomes Da Silva Junior**, **Discente**, em 15/01/2025, às 16:45, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Fernando Marques Federson, Professor do Magistério Superior**, em 16/01/2025, às 18:29, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **5089784** e o código CRC **A61B5A19**.

Referência: Processo nº 23070.001590/2025-26

SEI nº 5089784

JONAS GOMES DA SILVA JUNIOR

**Modelos de Linguagem para Recuperação e Geração de
Conhecimento Contextualizado**

Aplicação de Retrieval-Augmented Generation (RAG) em Reações Químicas e
Análise de Dados Científicos

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.

Orientador: Prof. Dr. Fernando Marques Federson

Goiânia

2025

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UFG.

JUNIOR, JONAS GOMES DA SILVA

Modelos de Linguagem para Recuperação e Geração de Conhecimento Contextualizado [manuscrito] : Aplicação de Retrieval Augmented Generation (RAG) em Reações Químicas e Análise de Dados Científicos / JONAS GOMES DA SILVA JUNIOR. - 2025.
60 f.

Orientador: Prof. Dr. Fernando Marques Federson.
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Goiás, Instituto de Informática (INF), Inteligência Artificial, Goiânia, 2025.

1. inteligência artificial. 2. geração aumentada de recuperação. 3. base de dados. I. Federson, Fernando Marques , orient. II. Título.

CDU 004

JONAS GOMES DA SILVA JUNIOR

**Modelos de Linguagem para Recuperação e Geração de
Conhecimento Contextualizado**

Aplicação de Retrieval-Augmented Generation (RAG) em Reações Químicas e
Análise de Dados Científicos

Relatório final de Trabalho de Conclusão de
Curso, apresentado à Universidade Federal
de Goiás, como parte das exigências para a
obtenção do título de Bacharel em Inteligência
Artificial.

Data da Aprovação: 17 de dezembro de 2024.



Prof. Dr. Fernando Marques Federson
Orientador (INF-UFG)



Prof. Dr. Aldo André Diaz Salazar
Coordenador de TCC do BIA (INF-UFG)



Prof. Dr. Anderson da Silva Soares
Coordenador do BIA (INF-UFG)



TAE Me. Raimunda Delfino Dos Santos Aguiar
(INF-UFG)

JONAS GOMES DA SILVA JUNIOR

Modelos de Linguagem para Recuperação e Geração de Conhecimento Contextualizado

Aplicação de Retrieval-Augmented Generation (RAG) em Reações Químicas e
Análise de Dados Científicos

RESUMO

Este Relatório de Conclusão de Curso tem como objetivo reunir os resultados da minha jornada para me tornar um especialista em **Geração Aumentada de Recuperação (RAG)**. Uma ilustração e sua narrativa descrevem os períodos de trabalho. Os Apêndices contêm os Termos de Aceite de Entrega e os resultados obtidos durante cada período de trabalho.

Palavras-chave: inteligência artificial, modelos grandes de linguagem, geração automática de datasets.

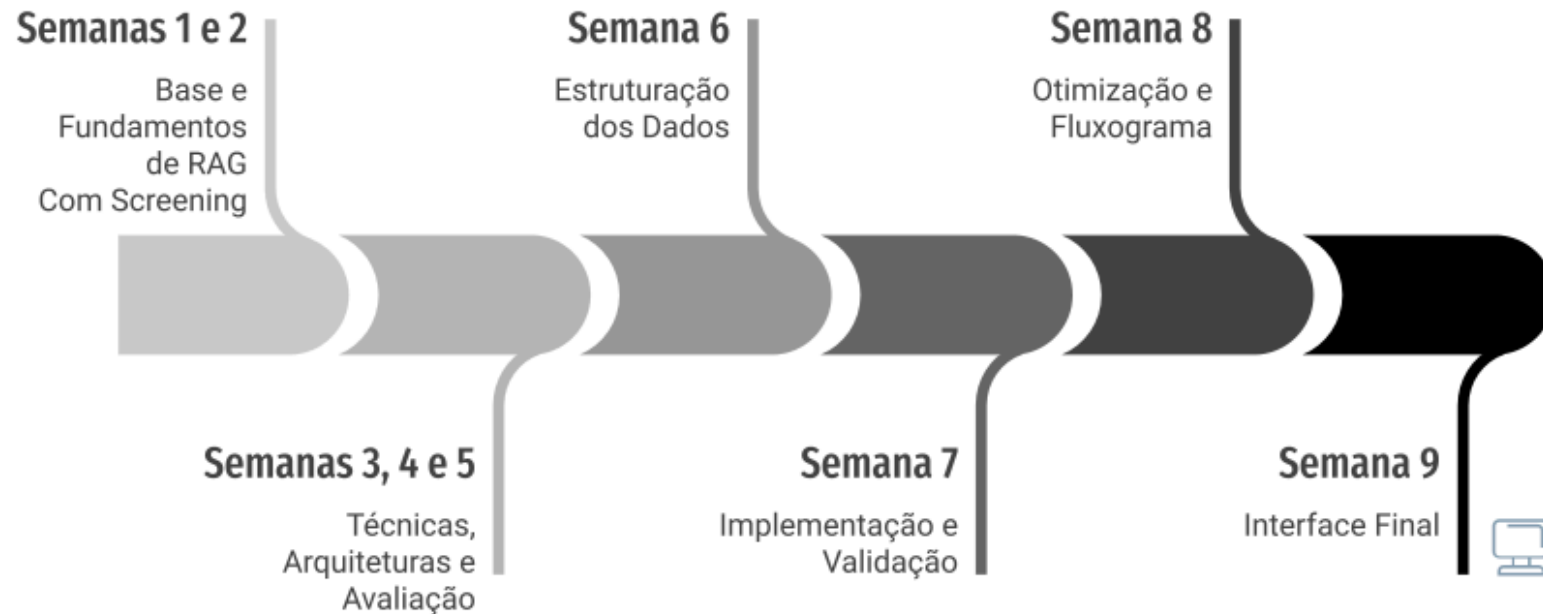
ABSTRACT

This Course Completion Report aims to bring together the results of my journey to become an expert in **Recovery Augmented Generation (RAG)**. An illustration and its narrative describe the work periods. The Appendices contain the Delivery Acceptance Terms and the results obtained during each work period.

Keywords: artificial intelligence, large language models, automatic dataset generation.

Goiânia
2025

Minha Jornada



Jonas Gomes Silva Junior

Especialista em: Geração Aumentada de Recuperação (RAG)

MINHA JORNADA

Nome: JONAS GOMES DA SILVA JUNIOR

Especialidade: Geração Aumentada de Recuperação (RAG)

Objetivo deste documento

Este documento tem como objetivo apresentar os resultados obtidos ao longo da disciplina de Residência em IA, evidenciando a trajetória de construção da minha especialização. Ao longo das semanas, cada conjunto de atividades e metas planejadas foi consolidado por meio de Termos de Aceite de Entrega, avaliados por uma banca examinadora. Aqui, serão descritos os principais resultados dos exercícios, bem como a relação destes com os Termos de Aceite e a documentação associada, oferecendo uma visão do progresso e da evolução do trabalho.

Minha Jornada

Iniciei minha jornada na **Semana 1** realizando um mapeamento preliminar de artigos científicos e conferências utilizando Screening, técnica usada a fim de definir com clareza a área de especialização na qual desejava aprofundar meus estudos. As pesquisas foram direcionadas à área de Processamento de Linguagem Natural (PLN) e modelos de linguagem de grande porte (LLMs), motivadas pelo meu interesse pessoal e pela sólida base oferecida pela disciplina de PLN cursada durante o Bacharelado. Esse conjunto de leituras iniciais forneceu subsídios importantes para a etapa seguinte. Na **Semana 2**, após avaliar diversos artigos, blogs e materiais de referência, decidi concentrar meus esforços na área de Geração Aumentada por Recuperação (Retrieval-Augmented Generation – RAG). Essa decisão foi fundamentada no potencial que a RAG apresenta ao integrar a recuperação de informações ao processo de geração de respostas contextuais, especialmente quando aplicada a modelos de linguagem de grande porte. O aprofundamento nesse tema, guiado por um artigo seminal da técnica, evidenciou que a combinação entre recuperação e geração não apenas eleva a eficácia dos sistemas baseados em IA, mas também oferece

soluções para lidar com problemas complexos. A RAG, portanto, surgiu como uma abordagem promissora, justificando minha escolha e delineando o foco das etapas subsequentes. Os materiais relacionados a estas duas Semanas podem ser encontrados no **Apêndice 1**.

Na **Semana 3**, motivado por questionamentos e discussões ocorridas durante um encontro no laboratório (“o gate”), investiguei as diferenças e as sinergias entre o Fine Tuning tradicional de LLMs e a abordagem RAG. Essa análise evidenciou que, longe de serem mutuamente excludentes, ambas as técnicas podem ser empregadas de modo complementar, resultando em modelos mais adaptados a contextos específicos. Ao longo das **Semanas 4 e 5**, aprofundei-me em estudos sobre técnicas avançadas de RAG, RAG Modular e diferentes métodos de avaliação de modelos. O Apêndice 2 apresenta a relação completa dos artigos lidos, acompanhada de observações que serviram de base para organizar o conhecimento adquirido e direcionar as etapas seguintes. Essas leituras tornaram-se fundamentais para compreender a flexibilidade arquitetural da RAG, permitindo sua adaptação a distintos cenários e problemas. Os materiais relacionados a estas três Semanas podem ser encontrados no **Apêndice 2**.

Com a área já definida, na **Semana 6**, decidi aplicar a RAG em um domínio que se beneficiaria de maior inteligência analítica: a área da química, uma área que não possui muitas aplicações práticas utilizando Modelos de Linguagem Grande. Ao realizar pesquisas nesse campo, encontrei o Open Reaction Database (ORD), uma infraestrutura de acesso aberto para compartilhamento de dados sobre reações orgânicas. Este repositório, que está sempre em atualização, gerou mais de 300.000 registros de reações químicas. Esse acervo de dados foi então processado e preparado para alimentar o pipeline de RAG. Os materiais relacionados a esta Semana podem ser encontrados no **Apêndice 3**.

Na **Semana 7**, implementei a RAG empregando o Qdrant, um banco de dados vetorial eficiente para lidar com embeddings e recuperação de informações relevantes. Com isso, estabeleci um fluxo que integrava dados, recuperação e geração de respostas contextuais, tornando o modelo apto a compreender e explorar o vasto conjunto de reações químicas

disponíveis. Os materiais relacionados a esta Semana podem ser encontrados no **Apêndice 4**.

Durante a **Semana 8**, refinei o pipeline, buscando otimizar o processo de recuperação e geração de contexto ao interagir com o modelo GPT-4. Paralelamente, elaborei um fluxograma detalhado do funcionamento de ponta a ponta da implementação, garantindo clareza e estruturando o raciocínio para futuras melhorias. Os materiais relacionados a esta Semana podem ser encontrados no **Apêndice 5**.

Por fim, na **Semana 9**, desenvolvi uma interface simples e intuitiva utilizando a ferramenta Streamlit. Esse frontend permite que usuários interajam facilmente com o “chatbot químico”, ao qual denominei RAI (“Reactions Artificial Intelligence”). Dessa forma, concluí o processo do projeto, consolidando a escolha da área de especialização em RAG e estabelecendo uma prova de conceito funcional e abrangente para futuras expansões e aprimoramentos. Os materiais relacionados a esta última Semana podem ser encontrados no **Apêndice 6**.

Em função de tudo que vivi nesta Jornada, gostaria de deixar registrado que o processo de Residência me fez desenvolver uma capacidade técnica e uma experiência de profunda imersão na área de desenvolvimento de um RAG, expandindo meus horizontes a respeito das possibilidades de aplicação de um conhecimento contextualizado utilizando Geração de Recuperação Aumentada.

APÊNDICE 1

Termo de Aceite de Entrega 1

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 18 de set. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

JONAS GOMES DA SILVA JUNIOR

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Nesta primeira etapa do trabalho, o foco foi na compreensão da história e dos fundamentos da técnica de Retrieval-Augmented Generation (RAG), além de estudar sua aplicação em modelos de linguagem grande (LLM).

Realizei a pesquisa e estudo do artigo seminal sobre RAG, incluindo a análise de componentes essenciais como o Recuperador (DPR) e o Gerador (BART), bem como o treinamento e experimentos descritos no artigo.

Produtos Gerados:

- Link para o primeiro trabalho formal sobre RAG publicado em 2020: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (Lewis et al., 2020) - <https://arxiv.org/pdf/2005.11401>
- Link para o documento de anotações com os principais pontos do estudo, como metodologia, modelos de RAG, Treinamento, Decodificação, Experimentos, Avaliação e Aplicações: Documento de anotações do estudo do Paper Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks - https://docs.google.com/document/d/1A6iAK8zrE19rqR0FoH903aBoWnzUu1o_qxM0WW91UOs
- Link para documentação de referência adicional da Databricks, que complementa a explicação dos problemas contornados pelo RAG, como a falta de atualização de conhecimento em tempo real e a ausência de controles de acesso: Documentação Databricks sobre RAG - <https://docs.databricks.com/pt/generative-ai/tutorials/ai-cookbook/rag-overview.html>

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Para a próxima entrega, pretendo aprofundar o estudo de aplicações práticas do RAG, especialmente em assistentes virtuais e sistemas educacionais, investigando implementações que utilizam fontes externas de conhecimento e como essas soluções estão sendo integradas nas empresas. Além disso, irei em busca de referências em artigos e papers de protótipos simples de RAG utilizando um **framework** para validar os conceitos teóricos estudados até o momento. A implementação incluirá a integração de um recuperador de informações com um modelo gerador, seguindo os princípios discutidos no artigo de Lewis et al., 2020.

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go! ▾

[Doc citado no Termo de Aceite de Entrega de 18 de Setembro]

Introdução ao RAG

O Retrieval-Augmented Generation (RAG) é uma técnica emergente no campo do Processamento de Linguagem Natural (NLP) que busca superar algumas limitações dos Grandes Modelos de Linguagem. Esses modelos, como o BERT e o GPT, por exemplo, aprendem uma vasta quantidade de conhecimento a partir de grandes corpos de dados durante o treinamento. No entanto, enfrentam dificuldades em expandir ou atualizar sua "memória" de maneira eficiente, além de não fornecerem explicações diretas para suas previsões, o que muitas vezes resulta em alucinações — situações em que o modelo gera informações imprecisas ou fictícias. Para resolver esse problema, pesquisadores da Meta AI propuseram o RAG, que combina técnicas de recuperação de informações com geração de

linguagem, criando um sistema mais robusto para tarefas que exigem conhecimento factual e atualizado.

O primeiro trabalho formal sobre RAG foi publicado em 2020: "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks" (Lewis et al., 2020) <https://arxiv.org/pdf/2005.11401>. Nesse artigo, os autores argumentam que, embora modelos de linguagem possam armazenar uma quantidade substancial de informações, eles não têm um mecanismo fácil para expandir ou revisar esse conhecimento em tempo real. Para superar isso, propuseram o uso de fontes externas de conhecimento que são acessadas durante o processo de geração de respostas.

Metodologia

O RAG é construído a partir da combinação de um recuperador de informações e um modelo gerador seq2seq pré-treinado, ambos treinados em um processo de ajuste fino de ponta a ponta. O conceito central é integrar uma memória paramétrica (um modelo seq2seq) com uma memória não-paramétrica (um índice denso da Wikipedia), que é acessada por um recuperador neural pré-treinado.

Modelos de RAG

RAG-Sequence: Este modelo utiliza o mesmo documento recuperado para gerar toda a sequência de resposta. Funciona tratando o documento recuperado como uma variável latente única, cuja probabilidade é marginalizada em relação à sequência de saída, utilizando uma aproximação top-K.

RAG-Token: Ao contrário do RAG-Sequence, o RAG-Token permite que o modelo desenhe um documento latente diferente para cada token da sequência gerada, marginalizando esses documentos à medida que a geração avança. Isso dá ao gerador mais flexibilidade para selecionar conteúdo de diferentes fontes ao compor uma resposta.

Recuperador: Dense Passage Retriever (DPR)

O Dense Passage Retriever (DPR) é utilizado como o componente de recuperação de informações do RAG. Ele consiste em dois codificadores baseados no BERTBASE: um codificador de consultas e um codificador de documentos. O DPR é treinado para recuperar documentos que contenham as respostas mais relevantes para perguntas específicas, utilizando datasets como o TriviaQA e o Natural Questions. Esse mecanismo de recuperação densa permite que o modelo busque documentos relevantes de forma eficiente, utilizando a técnica de Maximum Inner Product Search (MIPS) para resolver a busca por documentos de forma sublinear.

Gerador: BART

O componente gerador do RAG é baseado no BART, um transformador seq2seq pré-treinado com 400 milhões de parâmetros. O BART é responsável por combinar a entrada original com as informações recuperadas, gerando uma saída coerente e fluida. A entrada de consulta e os documentos recuperados são simplesmente concatenados antes de serem processados pelo BART, permitindo que o modelo gere respostas contextuais com base no conhecimento recuperado.

Treinamento

O treinamento do RAG é realizado de forma conjunta, sem supervisão direta sobre quais documentos específicos devem ser recuperados. Em vez disso, o modelo é ajustado para minimizar a log-verossimilhança marginal negativa de cada par de entrada/saída, utilizando o algoritmo de otimização Adam. Isso permite que o RAG aprenda quais documentos são mais relevantes para gerar uma resposta precisa, sem precisar de anotação explícita para a recuperação de documentos.

Decodificação

A fase de decodificação difere entre os dois modelos de RAG:

No RAG-Token, a probabilidade é calculada token a token, permitindo que o modelo escolha diferentes documentos latentes a cada passo de geração. Isso facilita a incorporação de informações de múltiplas fontes na resposta.

No RAG-Sequence, o processo é mais direto: o modelo utiliza o documento recuperado como variável latente e gera uma sequência completa de saída. Para encontrar a melhor resposta, o modelo realiza uma busca em feixe (beam search) para cada documento recuperado, pontuando as hipóteses de resposta de acordo com a probabilidade seq^2seq .

Experimentos

Para validar o desempenho do RAG, os autores conduziram experimentos em uma variedade de tarefas que exigem conhecimento factual, utilizando a Wikipedia como a base de dados de conhecimento não paramétrico. O dump de dezembro de 2018 da Wikipedia foi dividido em blocos de 100 palavras, totalizando cerca de 21 milhões de documentos. Os experimentos mostraram que o RAG supera outros modelos em tarefas como perguntas e respostas factuais, oferecendo respostas mais precisas e baseadas em fontes confiáveis.

Aplicações do RAG

Desde sua introdução, o RAG tem sido aplicado em diversas áreas, como por exemplo:

Assistentes virtuais - Utilizado para melhorar a precisão das respostas fornecidas por sistemas de perguntas e respostas em domínios como atendimento ao cliente e assistentes pessoais., Sistemas educacionais - O RAG pode ser usado para responder a perguntas complexas e fornecer explicações baseadas em fontes de conhecimento; Diagnóstico médico - Integrado a bases de dados médicas, o RAG tem o potencial de fornecer diagnósticos assistidos por IA, acessando diretamente artigos científicos e estudos médicos. Entre outros diversos exemplos.

Referências: Introdução ao RAG - DataBricks
<https://docs.databricks.com/pt/generative-ai/tutorials/ai-cookbook/rag-overview.html>

Termo de Aceite de Entrega 2

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 18 de set. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

JONAS GOMES DA SILVA JUNIOR

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Neste Gate procurei mais referências de artigos que fossem relevantes para o meu estudo em RAG, utilizando de um pequeno **Screening** para essa busca de artigos, e felizmente percebi que o artigo que eu estava usando já era um dos mais citados (3580 vezes - fonte: Google Scholar), pois foi o pioneiro, o início de todos.

Problemática: Modelos pré-treinados não podem expandir ou revisar facilmente sua memória, não podem fornecer insights diretos sobre suas previsões, não possuem referências atualizadas e podem produzir “alucinações”. Referência - <https://arxiv.org/pdf/2005.11401> - Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.

Com isso, o artigo propõe que a base de um RAG Simples utiliza de uma arquitetura que combina **Recuperador** um dos mais comuns é o DPR - Dense Passage Retriever e um **Gerador** normalmente um modelo seq2seq (abreviação de sequence-to-sequence) é uma arquitetura de aprendizado profundo usada para converter uma sequência de entrada em uma sequência de saída.

- Recuperador DPR- codifica a consulta e os documentos e retorna os documentos top-k similares a consulta.
- Gerador - concatena a consulta e os documentos recuperados e, em seguida, usa isso como contexto para gerar uma resposta coerente.

Fluxo de Dados em uma Arquitetura Simples de RAG:

Entrada: O usuário faz uma pergunta (consulta), como por exemplo:
"Qual planeta possui mais luas?"

Recuperação de Documentos: O Recuperador (DPR) transforma a consulta em um vetor.

Ele busca na base de dados externa (por exemplo, a Wikipedia) os documentos mais relevantes relacionados à consulta.

Geração da Resposta: O modelo de geração constrói a resposta baseando-se nas informações dos documentos, evitando inventar respostas e garantindo que elas estejam ancoradas nos dados recuperados.

Saída: uma resposta ao usuário, como "Saturno é o planeta com mais luas do Sistema Solar, com 146 luas em junho de 2023."

Obs: Até 2021 Júpiter era o planeta com mais lua, com 92.

Além disso, fui em busca de aplicações práticas e encontrei uma comunidade de Inteligência Artificial com foco em RAG chamada "DiamantAI" que ensina sobre as arquiteturas RAG, com implementações que vão desde uma arquitetura simples à utilização de Métodos Avançados de Recuperação.

Também procurei frameworks e ferramentas que dêem suporte para implementações RAG - encontrando como uma das principais ferramentas a Biblioteca Pytorch e também o Hugging Face Transformers que oferece API's e ferramentas para modelos pré-treinados. - <https://huggingface.co/docs/transformers/index>

Documento de resumo de estudo:  Resumo - Gate 2

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Através de referências atuais como artigos recentes e com as referências da comunidade buscar aprofundamento nas outras técnicas de RAG além do RAG Simples, explorando as diferenças e aplicações das técnicas em diferentes contextos.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: 

[Doc citado no Termo de Aceite de Entrega de 25 de Setembro]

Arquitetura básica de um RAG

O diferencial do RAG em comparação com modelos puramente gerativos está em sua capacidade de acessar bases de conhecimento externas, o que torna as respostas mais relevantes e baseadas em fatos verificáveis.

Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (Lewis et al., 2020) - <https://arxiv.org/pdf/2005.11401> - Continuando usando a referência inicial podemos tirar algumas análises sobre a forma básica de uma arquitetura de Rag:

Para isso ele utiliza de uma arquitetura que combina Recuperador pré-treinado (Query Encoder + Document Index) com um modelo seq2seq pré-treinado (Generator) e fazemos ajustes finos de ponta a ponta.

Houve um extenso trabalho anterior propondo arquiteturas para enriquecer sistemas com memória não paramétrica que são treinados do zero para tarefas específicas, por exemplo, redes de memória, redes aumentadas em pilha e camadas de memória. Em contraste, exploramos um cenário onde ambos os componentes de memória paramétrica e não paramétrica são pré-treinados e pré-carregados com amplo conhecimento. Crucialmente, ao usar mecanismos de acesso pré-treinados, a capacidade de acessar conhecimento está presente sem treinamento adicional.

Uma arquitetura simples de RAG (Retrieval-Augmented Generation) combina dois componentes principais: recuperação de informações e geração de texto, com o objetivo de fornecer respostas factuais e baseadas em documentos relevantes. Ou seja, temos:

Recuperador de Informações (Retriever)

O papel do recuperador é buscar documentos ou passagens de texto relevantes para a consulta do usuário. Ele utiliza uma técnica de busca semântica, onde tanto a consulta quanto os documentos são representados como vetores em um espaço de embeddings.

Um dos modelos mais comuns para essa função é o Dense Passage Retriever (DPR), que utiliza dois codificadores baseados em BERT:

Codificador de consulta: Converte a consulta do usuário (a pergunta) em um vetor.

Codificador de documentos: Converte os documentos em vetores.

O recuperador então busca documentos mais próximos no espaço vetorial, baseando-se na similaridade entre os embeddings da consulta e dos documentos.

Gerador de Texto (Generator)

O gerador de texto, normalmente um modelo seq2seq como o BART (Bidirectional and Auto-Regressive Transformer), recebe a consulta e os documentos recuperados.

Ele utiliza esses documentos como contexto para gerar uma resposta fluida e coerente.

O gerador concatena a consulta com os documentos mais relevantes e, usando seu mecanismo de codificação-decoding, gera a resposta de forma autorregressiva (token por token).

Fluxo de Dados em uma Arquitetura Simples de RAG

- Entrada: O usuário faz uma pergunta (consulta), como por exemplo: "Quem inventou a lâmpada elétrica?"
- Recuperação de Documentos: O Recuperador (DPR) transforma a consulta em um vetor.
- Ele busca em uma base de dados externa (por exemplo, a Wikipedia) os documentos mais relevantes relacionados à consulta.
- O DPR retorna os Top-K documentos (geralmente os mais relevantes), como trechos de texto ou passagens que possuam informações sobre a invenção da lâmpada.
- Geração da Resposta:
- O Gerador (BART) concatena a consulta e os documentos recuperados e, em seguida, usa isso como contexto para gerar uma resposta coerente.

- O modelo de geração constrói a resposta baseando-se nas informações dos documentos, evitando inventar respostas e garantindo que elas estejam ancoradas nos dados recuperados.
- Saída: O gerador fornece uma resposta ao usuário, como "A lâmpada elétrica foi inventada por Thomas Edison no final do século XIX, com base em experimentos de outros inventores como Humphry Davy e Joseph Swan."

Variantes Simples do RAG

Existem duas principais variantes de como o RAG pode utilizar os documentos recuperados para gerar a resposta:

RAG-Sequence: Neste modelo, o RAG usa o mesmo documento recuperado para gerar toda a sequência da resposta. O processo funciona de maneira simples: o gerador usa um único documento como referência e gera a resposta completa.

RAG-Token: Neste modelo, o RAG pode usar diferentes documentos para gerar cada token da resposta. Isso permite que o modelo combine informações de vários documentos, escolhendo qual documento é mais relevante em cada etapa da geração.

Exemplo Simples:

Pergunta: "Quem inventou a lâmpada?"

Recuperação: O Recuperador (DPR) busca na base de dados e encontra documentos que mencionam Thomas Edison, Humphry Davy e Joseph Swan.

Geração:

O Gerador (BART) concatena a pergunta com esses documentos e gera a resposta: "A lâmpada elétrica foi inventada por Thomas Edison em 1879, embora Humphry Davy tenha feito experimentos com lâmpadas elétricas no início do século XIX."

Principais Vantagens de uma Arquitetura Simples de RAG:

- **Atualização de conhecimento:** Como o modelo acessa uma base de dados externa (não-paramétrica), ele pode fornecer respostas atualizadas e baseadas em fatos.

- Redução de alucinações: Ao buscar documentos relevantes, o modelo gera respostas ancoradas em dados reais, minimizando a invenção de informações.
- Eficiência: Com DPR e BART, a arquitetura é relativamente simples e eficiente em termos de implementação, mas ainda poderosa para tarefas que exigem conhecimento factual.

Frameworks como:

- Hugging Face Transformers: Oferece modelos pré-treinados como DPR e BART, facilitando a implementação.
- PyTorch: Um framework popular que pode ser usado para treinar e ajustar os componentes do RAG.

APÊNDICE 2

Termo de Aceite de Entrega 3

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 10 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

JONAS GOMES DA SILVA JUNIOR

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Este gate foi focado no estudo dos métodos de avaliação de RAG - utilizando como referência o Survey [Retrieval-Augmented Generation for Large Language Models: A Survey](#)

AVALIAÇÃO DE MODELOS RAG

Question Answering (QA)

Tipos de QA:

- Hop único/múltiplo.
- Múltipla escolha.
- Domínio específico.
- Resposta longa

Métricas para QA:

- Exact Match (EM): Mede se a resposta é exatamente igual à resposta esperada.
- F1: Considera uma combinação de precisão e recall, útil quando as respostas são mais flexíveis e podem não ser exatas, mas ainda assim corretas..
 - Verificação de Fatos: Usa a acurácia para verificar se as informações geradas estão corretas.
 - Qualidade das Respostas: Métricas como BLEU e ROUGE, que avaliam a similaridade da resposta gerada com uma resposta de referência.

BLEU (Bilingual Evaluation Understudy) - Mede a correspondência entre n-gramas (sequências de palavras) do texto gerado e os n-gramas dos textos de referência.

Existem várias variantes do ROUGE:

- ROUGE-N: Mede a sobreposição de n-gramas (como BLEU).
- ROUGE-L: Mede a maior subsequência comum entre o texto gerado e o texto de referência, avaliando similaridade estrutural.
- ROUGE-S: Mede a co-ocorrência de pares de palavras no texto gerado e no texto de referência, mesmo que não estejam na mesma ordem (skip-bigram).

Avaliação da Qualidade da Recuperação

- Hit Rate: Mede a frequência com que a recuperação traz informações corretas.
- MRR (Mean Reciprocal Rank): Mede a posição da primeira resposta correta.
- NDCG (Normalized Discounted Cumulative Gain): Avalia a qualidade da ordenação das respostas.

Qualidade da Geração

- Fidelidade: A resposta está em conformidade com o contexto recuperado?
- Relevância: A resposta é relevante para a pergunta feita?
- Não Nocividade: O modelo evita gerar conteúdo inapropriado ou prejudicial quando não há rótulos para isso?

São usadas três pontuações de qualidade

- Relevância do Contexto: Avalia a precisão do contexto que foi recuperado para gerar a resposta.
- Fidelidade da Resposta: Avalia se a resposta gerada é consistente com as informações do contexto.
- Relevância da Resposta: Mede o quanto a resposta se alinha com a pergunta feita.

Critérios também avaliados

- Robustez ao Ruído: O modelo deve lidar com documentos que podem conter informações irrelevantes ou confusas.
- Rejeição de Negativos: Evitar gerar respostas que não contêm informação útil ou relevante.
- Integração de Informações: Combinar informações de múltiplos documentos de forma coerente.
- Robustez Contrafactual: O modelo deve reconhecer e ignorar informações incorretas ou imprecisas presentes nos documentos.

Frameworks como RALLE são utilizadas. - <https://arxiv.org/pdf/2308.10633>

Anotacoes de estudo -
<https://docs.google.com/document/d/1Mlj-BijAQ0ejuGYgP8a4F21xymKMA0F-aWYmJE4fHmo/edit?usp=drivesdk>

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Aprofundar sobre a estrutura das diferentes técnicas utilizadas no RAG avançado e Modular e também estudar sobre LlamaIndex2, LangChain3 e HayStack.

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go! ▾

[Doc citado no Termo de Aceite de Entrega de 03 de Outubro]

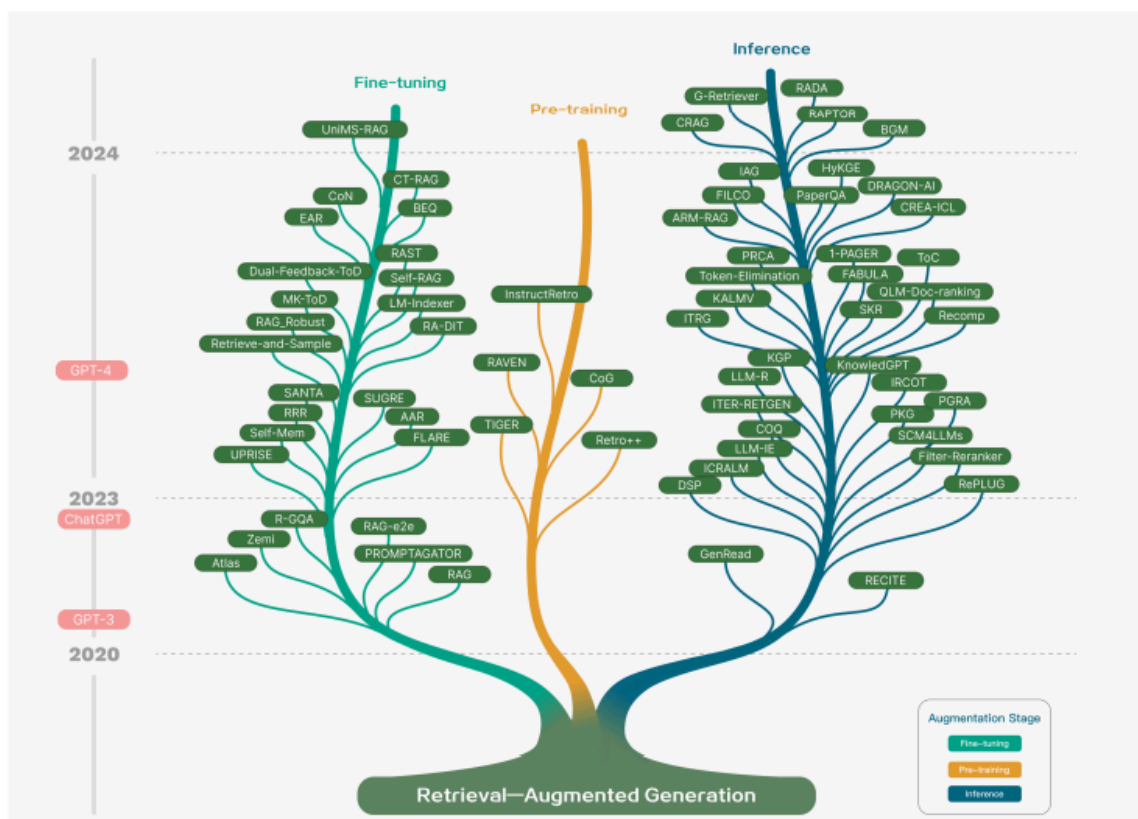
Fine Tuning vs. Retrieval Augmented Generation

O RAG recupera informações relevantes de um corpus de documentos e aprimora a geração de respostas do LM por meio da implementação de aprendizagem em contexto. Por outro lado, a abordagem FT atualiza os pesos do modelo para se tornar adepto de relembrar informações específicas e aprimorar suas capacidades de memorização durante a

inferência. - Referência: Fine Tuning vs. Retrieval Augmented Generation for Less Popular Knowledge - <https://arxiv.org/pdf/2403.01432>

Entretanto, combinar fine-tuning com RAG é uma abordagem poderosa que pode melhorar a performance do sistema tanto na recuperação quanto na geração de respostas. O fine-tuning pode ser aplicado ao modelo gerador, ao modelo recuperador ou ambos, tornando o sistema mais eficaz em contextos especializados ou em tarefas muito técnicas. Essa combinação oferece o melhor dos dois métodos: o refinamento específico de fine-tuning com a flexibilidade e atualização contínua da recuperação de informações do RAG.

A tecnologia RAG se desenvolveu rapidamente nos últimos anos, e a árvore tecnológica resumindo a pesquisa relacionada é mostrada na Figura abaixo. A trajetória de desenvolvimento do RAG na era de grandes modelos exibe várias características de estágio distintas.



RAG Naive (ingênuo) - indexação, recuperação e geração:

- A indexação começa com a limpeza e extração de dados brutos em diversos formatos como PDF, HTML, Word e Markdown, que são então convertidos em um formato de texto simples uniforme. Para acomodar as limitações de contexto dos modelos de linguagem, o texto é segmentado em pedaços menores e digeríveis. Os pedaços são então codificados em representações vetoriais usando um modelo de incorporação e armazenados em um banco de dados vetorial.
- Recuperação. Após o recebimento de uma consulta do usuário, o sistema RAG emprega o mesmo modelo de codificação utilizado durante a fase de indexação para transformar a consulta em uma representação vetorial. Em seguida, ele calcula as pontuações de similaridade entre o vetor de consulta e o vetor de blocos dentro do corpus indexado. O sistema prioriza e recupera os principais blocos K que demonstram a maior similaridade com a consulta.
- Geração. A consulta proposta e os documentos selecionados são sintetizados em um prompt coerente para o qual um grande modelo de linguagem é encarregado de formular uma resposta. A abordagem do modelo para responder pode variar dependendo dos critérios específicos da tarefa, permitindo que ele utilize seu conhecimento paramétrico inerente ou restrinja suas respostas às informações contidas nos documentos fornecidos.

Advanced RAG introduz melhorias específicas para superar as limitações do Naive RAG. Com foco em melhorar a qualidade da recuperação, ele emprega estratégias de pré-recuperação e pós-recuperação. Para lidar com os problemas de indexação, o Advanced RAG refina suas técnicas de indexação por meio do uso de uma abordagem de janela deslizante, segmentação refinada e incorporação de metadados. Além disso, ele incorpora vários métodos de otimização para agilizar o processo de recuperação.

- Pré-recuperação. Nesta fase, o foco principal é na otimização da estrutura de indexação e da consulta original. O objetivo da otimização da indexação é melhorar a qualidade do conteúdo que está sendo indexado. Isso envolve estratégias: melhorar a granularidade dos dados, otimizar estruturas de índice, adicionar metadados, otimização de alinhamento e recuperação mista. Enquanto o objetivo da otimização da consulta é tornar a pergunta original do usuário mais clara e mais adequada para a tarefa de recuperação. Métodos comuns incluem reescrita de consulta, transformação de consulta, expansão de consulta e outras técnicas.

- Processo de Pós-Recuperação. Uma vez que o contexto relevante é recuperado, é crucial integrá-lo efetivamente com a consulta. Os principais métodos no processo de pós-recuperação incluem reclassificação de blocos e compressão de contexto. Reclassificar as informações recuperadas para realocar o conteúdo mais relevante para as bordas do prompt é uma estratégia fundamental. Este conceito foi implementado em estruturas como LlamaIndex2, LangChain3 e HayStack. Alimentar todos os documentos relevantes diretamente em LLMs pode levar à sobrecarga de informações, diluindo o foco em detalhes importantes com conteúdo irrelevante. Para mitigar isso, os esforços de pós-recuperação concentram-se na seleção das informações essenciais, enfatizando seções críticas e encurtando o contexto a ser processado.

A arquitetura RAG modular avança além dos dois paradigmas RAG anteriores, oferecendo adaptabilidade e versatilidade aprimoradas. Ela incorpora diversas estratégias para melhorar seus componentes, como adicionar um módulo de pesquisa para pesquisas de similaridade e refinar o recuperador por meio de ajuste fino. Inovações como módulos RAG reestruturados e pipelines RAG reorganizados foram introduzidas para enfrentar desafios específicos. Apesar de sua distinção, o RAG Modular se baseia nos princípios fundamentais do RAG Avançado e Naive, ilustrando uma progressão e refinamento dentro da família RAG:

- Novos Módulos: A estrutura RAG Modular introduz componentes especializados adicionais para aprimorar as capacidades de recuperação e processamento. O módulo Search se adapta a cenários específicos, permitindo pesquisas diretas em várias fontes de dados, como mecanismos de pesquisa, bancos de dados e gráficos de conhecimento, usando código gerado pelo LLM e linguagens de consulta.
 - O RAGFusion aborda as limitações tradicionais de pesquisa empregando uma estratégia de múltiplas consultas que expande as consultas do usuário em diversas perspectivas, utilizando pesquisas de vetores paralelos e reclassificação inteligente para descobrir conhecimento explícito e transformador.
 - O módulo Memory aproveita a memória do LLM para orientar a recuperação, criando um pool de memória ilimitado que alinha o texto mais de perto com a distribuição de dados por meio de auto aprimoramento iterativo.
 - O roteamento no sistema RAG navega por diversas fontes de dados, selecionando o caminho ideal para uma consulta, seja envolvendo resumo,

pesquisas específicas em bancos de dados ou mesclando diferentes fluxos de informações.

- O módulo Predict visa reduzir a redundância e o ruído gerando contexto diretamente por meio do LLM, garantindo relevância e precisão.
- Por fim, o módulo Task Adapter adapta o RAG a várias tarefas posteriores, automatizando a recuperação rápida para entradas de disparo zero e criando recuperadores específicos para tarefas por meio da geração de consultas de disparos poucos.
- Novos padrões: o RAG modular oferece adaptabilidade notável ao permitir a substituição ou reconfiguração de módulos para abordar desafios específicos. Isso vai além das estruturas fixas do RAG Naive e Advanced, caracterizadas por um mecanismo simples de “Recuperação” e “Ler”. Além disso, o RAG modular expande essa flexibilidade integrando novos módulos ou ajustando o fluxo de interação entre os existentes, aprimorando sua aplicabilidade em diferentes tarefas. Inovações como o modelo Rewrite-Retrieve-Read aproveitam as capacidades do LLM para refinar consultas de recuperação por meio de um módulo de reescrita e um mecanismo de feedback LM para atualizar o modelo de reescrita, melhorando o desempenho da tarefa.

Da mesma forma, abordagens como Generate-Read substituem a recuperação tradicional por conteúdo gerado pelo LLM, enquanto ReciteRead enfatiza a recuperação de pesos do modelo, aprimorando a capacidade do modelo de lidar com tarefas intensivas em conhecimento.

Estratégias de recuperação híbrida integram palavras-chave, semânticas e pesquisas vetoriais para atender a diversas consultas. Além disso, empregar subconsultas e incorporações hipotéticas de documentos (HyDE) busca melhorar a relevância da recuperação ao focar em incorporar similaridades entre respostas geradas e documentos reais.

Ajustes no arranjo e interação do módulo, como a estrutura Demonstrate-Search-Predict (DSP) e o fluxo iterativo Retrieve-Read-Retrieve-Read do ITERRETGEN, demonstram o uso dinâmico de saídas do módulo para reforçar a funcionalidade de outro módulo, ilustrando uma compreensão sofisticada de aprimoramento da sinergia do módulo.

A orquestração flexível do Modular RAG Flow demonstra os benefícios da recuperação adaptativa por meio de técnicas como FLARE e Self-RAG. Essa abordagem

transcende o processo fixo de recuperação do RAG ao avaliar a necessidade de recuperação com base em diferentes cenários. Outro benefício de uma arquitetura flexível é que o sistema RAG pode se integrar mais facilmente a outras tecnologias (como ajuste fino ou aprendizado por reforço) . Por exemplo, isso pode envolver ajuste fino do recuperador para melhores resultados de recuperação, ajuste fino do gerador para saídas mais personalizadas ou envolvimento em ajuste fino colaborativo.

[Doc citado no Termo de Aceite de Entrega de 10 de Outubro]

AVALIAÇÃO DE MODELOS RAG

O rápido avanço e a crescente adoção do RAG no campo de Processamento de Linguagem Natural impulsionaram a avaliação dos modelos de RAG para o centro das pesquisas na comunidade de LLMs. O principal objetivo dessa avaliação é compreender e otimizar o desempenho dos modelos de RAG em diversos cenários de aplicação.

A. Tarefa Downstream

A principal tarefa do RAG continua sendo o Question Answering (QA), incluindo QA tradicional de hop único/múltiplo, múltipla escolha, QA específico de domínio, bem como cenários de resposta longa, adequados para o RAG. Além de QA, o RAG está se expandindo continuamente para várias outras tarefas downstream, como extração de informações (IE), geração de diálogos, busca de código, etc. As principais tarefas downstream do RAG e seus datasets correspondentes estão resumidos na Tabela II.

B. Objetivo da Avaliação

Historicamente, as avaliações dos modelos de RAG focam em sua execução em tarefas downstream específicas. Essas avaliações empregam métricas estabelecidas adequadas para as tarefas. Por exemplo, as avaliações de QA podem usar as pontuações de Exact Match (EM) e F1, enquanto as tarefas de verificação de fatos costumam usar a acurácia como a principal métrica. Métricas como BLEU e ROUGE também são comumente usadas para avaliar a qualidade das respostas. Ferramentas como o RALLE, projetadas para a avaliação automática de aplicações de RAG, também baseiam suas avaliações nessas métricas específicas de tarefas. Ainda assim, há uma escassez notável de pesquisas

dedicadas a avaliar as características distintas dos modelos de RAG. Os principais objetivos de avaliação incluem:

Qualidade da Recuperação: Avaliar a qualidade da recuperação é crucial para determinar a eficácia do contexto fornecido pelo componente recuperador. Métricas padrão de motores de busca, sistemas de recomendação e sistemas de recuperação de informações são usadas para medir o desempenho do módulo de recuperação do RAG, como a taxa de acerto (Hit Rate), MRR e NDCG.

Qualidade da Geração: A avaliação da qualidade da geração se concentra na capacidade do gerador de sintetizar respostas coerentes e relevantes a partir do contexto recuperado. Essa avaliação pode ser categorizada com base nos objetivos do conteúdo: conteúdo não rotulado e rotulado. Para conteúdo não rotulado, a avaliação abrange a fidelidade, relevância e não nocividade das respostas geradas. Para conteúdo rotulado, o foco está na precisão das informações geradas pelo modelo. Além disso, as avaliações de qualidade da recuperação e da geração podem ser conduzidas por métodos de avaliação manuais ou automáticos.

C. Aspectos da Avaliação

As práticas contemporâneas de avaliação de modelos de RAG enfatizam três principais pontuações de qualidade e quatro habilidades essenciais, que juntas informam a avaliação dos dois principais alvos do modelo de RAG: recuperação e geração.

1. **Pontuações de Qualidade:** As pontuações de qualidade incluem relevância do contexto, fidelidade da resposta e relevância da resposta. Essas pontuações avaliam a eficiência do modelo RAG a partir de diferentes perspectivas no processo de recuperação e geração de informações:
 - **Relevância do Contexto** avalia a precisão e especificidade do contexto recuperado, garantindo relevância e minimizando custos de processamento associados a conteúdos desnecessários.
 - **Fidelidade da Resposta** assegura que as respostas geradas permanecem fiéis ao contexto recuperado, mantendo consistência e evitando contradições.
 - **Relevância da Resposta** exige que as respostas geradas sejam diretamente pertinentes às perguntas feitas, abordando efetivamente a questão principal.

2. **Habilidades Necessárias:** A avaliação de RAG também abrange quatro habilidades indicativas de sua adaptabilidade e eficiência:

- **Robustez ao Ruído** avalia a capacidade do modelo de lidar com documentos ruidosos relacionados à pergunta, mas que carecem de informações substanciais.

- Rejeição de Negativos mede a capacidade do modelo de evitar responder quando os documentos recuperados não contêm o conhecimento necessário para responder a uma pergunta.
- Integração de Informações avalia a habilidade do modelo em sintetizar informações de múltiplos documentos para lidar com perguntas complexas.
- Robustez Contrafactual testa a habilidade do modelo de reconhecer e ignorar imprecisões conhecidas dentro dos documentos, mesmo quando instruído sobre possíveis desinformações.

A relevância do contexto e a robustez ao ruído são importantes para avaliar a qualidade da recuperação, enquanto a fidelidade da resposta, a relevância da resposta, a rejeição de negativos, a integração de informações e a robustez contrafactual são importantes para avaliar a qualidade da geração.

Termo de Aceite de Entrega 4

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 17 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

JONAS GOMES DA SILVA JUNIOR

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

AVALIAÇÃO DE MODELOS RAG

Question Answering (QA):

Métricas para QA:

- Exact Match (EM): Mede se a resposta é exatamente igual à resposta esperada.
- F1: Considera uma combinação de precisão e recall
- Verificação de Fatos: Usa a acurácia para verificar se as informações geradas estão corretas.
- Qualidade das Respostas: Métricas como BLEU e ROUGE, avaliam a similaridade da resposta gerada com uma resposta de referência.

BLEU (Bilingual Evaluation Understudy) - Mede a correspondência entre n-gramas (sequências de palavras) do texto gerado e os n-gramas dos textos de referência.

Existem várias variantes do ROUGE:

ROUGE-N: Mede a sobreposição de n-gramas (como BLEU).

ROUGE-L: Mede a maior subsequência comum entre o texto gerado e o texto de referência, avaliando similaridade estrutural.

ROUGE-S: Mede a co-ocorrência de pares de palavras no texto gerado e no texto de referência, mesmo que não estejam na mesma ordem (skip-bigram).

Avaliação da Qualidade da Recuperação

- Hit Rate: Mede a frequência com que a recuperação traz informações corretas.
- MRR (Mean Reciprocal Rank): Mede a posição da primeira resposta correta.
- NDCG (Normalized Discounted Cumulative Gain): Avalia a qualidade da ordenação das respostas.

Qualidade da Geração

- Fidelidade: A resposta está em conformidade com o contexto recuperado?
- Relevância: A resposta é relevante para a pergunta feita?
- Não Nocividade: O modelo evita gerar conteúdo inapropriado ou prejudicial quando não há rótulos para isso?

São usadas três pontuações de qualidade Gerais

- Relevância do Contexto: Avalia a precisão do contexto que foi recuperado para gerar a resposta.
- Fidelidade da Resposta: Avalia se a resposta gerada é consistente com as informações do contexto.
- Relevância da Resposta: Mede o quanto a resposta se alinha com a pergunta feita.

Todas essas avaliações vieram do paper [Retrieval-Augmented Generation for Large Language Models: A Survey](#).

Técnicas de RAG avançado com LangChain - É um Framework que oferece ferramentas poderosas para construir sistemas RAG avançados.

Indexação hierárquica - organiza a base de conhecimento em uma estrutura semelhante a uma árvore, permitindo uma recuperação mais rápida e eficiente de documentos relevantes.

Quantização vetorial - comprime as representações vetoriais de documentos para reduzir os requisitos de armazenamento e melhorar a velocidade de pesquisa. Utiliza algoritmos como PCA (Análise de Componentes Principais) ou técnicas mais avançadas de compressão para reduzir a dimensão dos vetores sem perder informações cruciais.

LangChain oferece uma ampla gama de técnicas de recuperação, como busca semântica e expansão de

consulta, que podem ser facilmente integradas e experimentadas para otimizar o desempenho da recuperação.

O LangChain também suporta integração com bancos de dados de vetores populares, como Elasticsearch , permitindo armazenamento e recuperação eficientes de representações de vetores. Ao alavancar os recursos de otimização de recuperação do LangChain, os desenvolvedores podem ajustar seus sistemas RAG para atingir alta precisão e eficiência de recuperação.

Pós- Recuperação - Técnicas de otimização pós-recuperação focam em refinar as informações recuperadas antes de passá-las para o estágio de geração de resposta. O LlamalIndex oferece métodos como feedback de relevância, que permite que o sistema RAG refine iterativamente os resultados de recuperação com base no feedback do usuário, melhorando a relevância do contexto recuperado ao longo do tempo.

Outra técnica de otimização pós-recuperação é a filtragem de informações, que remove informações irrelevantes ou redundantes do texto recuperado. Ao aplicar essas otimizações, os sistemas RAG podem fornecer respostas mais concisas e direcionadas às consultas do usuário.

CRAG - Corrective Retrieval-Augmented Generation (CRAG) é uma técnica avançada que visa melhorar a precisão factual das respostas geradas. O CRAG funciona recuperando e integrando iterativamente informações relevantes da base de conhecimento para corrigir e refinar a saída gerada.

 Gate5 - Documento de Anotações

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Continuar explorando técnicas de RAG avançado e possibilidades de diferentes métodos e soluções dentro do LangChain

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: 

[Doc citado no Termo de Aceite de Entrega de 17 de Outubro]

Implementando RAG avançado com LlamaIndex e LangChain

LlamaIndex e LangChain são duas bibliotecas populares de código aberto que oferecem ferramentas poderosas para construir sistemas RAG avançados. Essas bibliotecas fornecem uma variedade de recursos e otimizações para simplificar o processo de desenvolvimento e melhorar o desempenho dos aplicativos RAG.

Otimização de Indexação

O LlamaIndex oferece várias técnicas de indexação para otimizar o processo de recuperação em sistemas RAG. Uma dessas técnicas é a indexação hierárquica, que organiza a base de conhecimento em uma estrutura semelhante a uma árvore, permitindo uma recuperação mais rápida e eficiente de documentos relevantes.

Outra otimização de indexação disponível no LlamaIndex é a quantização vetorial, que comprime as representações vetoriais de documentos para reduzir os requisitos de armazenamento e melhorar a velocidade de pesquisa. Ao alavancar essas otimizações de indexação, os desenvolvedores podem construir sistemas RAG escaláveis capazes de lidar com grandes bases de conhecimento.

Otimização de Recuperação

LangChain fornece uma estrutura flexível e modular para construir pipelines de recuperação em sistemas RAG. Ele oferece uma ampla gama de técnicas de recuperação, como busca semântica e expansão de consulta, que podem ser facilmente integradas e experimentadas para otimizar o desempenho da recuperação.

O LangChain também suporta integração com bancos de dados de vetores populares, como Pinecone e Elasticsearch, permitindo armazenamento e recuperação eficientes de representações de vetores. Ao alavancar os recursos de otimização de recuperação do LangChain, os desenvolvedores podem ajustar seus sistemas RAG para atingir alta precisão e eficiência de recuperação.

Otimização pós-recuperação

Técnicas de otimização pós-recuperação focam em refinar as informações recuperadas antes de passá-las para o estágio de geração de resposta. O LlamaIndex oferece métodos como feedback de relevância, que permite que o sistema RAG refine iterativamente os resultados de recuperação com base no feedback do usuário, melhorando a relevância do contexto recuperado ao longo do tempo.

Outra técnica de otimização pós-recuperação é a filtragem de informações, que remove informações irrelevantes ou redundantes do texto recuperado. Ao aplicar essas otimizações, os sistemas RAG podem fornecer respostas mais concisas e direcionadas às consultas do usuário.

Implementação do CRAG

Corrective Retrieval-Augmented Generation (CRAG) é uma técnica avançada de RAG que visa melhorar a precisão factual das respostas geradas. O CRAG funciona recuperando e integrando iterativamente informações relevantes da base de conhecimento para corrigir e refinar a saída gerada.

O LlamaIndex fornece uma implementação do CRAG, que mostrou resultados promissores em benchmarks como o MTEB leaderboard. Ao alavancar o CRAG, os desenvolvedores podem construir sistemas RAG que geram respostas mais precisas e confiáveis, mesmo para consultas complexas que exigem várias etapas de recuperação.

Algumas outras Técnicas avançadas de RAG

Técnicas avançadas de RAG visam abordar as limitações do Naive RAG incorporando mecanismos de recuperação e geração mais sofisticados. Isso pode incluir expansão de consulta, onde termos adicionais são adicionados à consulta do usuário para melhorar a precisão da recuperação ou recuperação iterativa, onde o modelo recupera documentos em vários estágios para refinar o contexto.

Os sistemas RAG avançados também podem empregar técnicas como mecanismos de atenção para ajudar o modelo a focar nas partes mais relevantes dos documentos

recuperados durante a geração. Ao atender seletivamente a diferentes aspectos do contexto, o modelo pode gerar respostas mais precisas e contextualmente relevantes.

Dutos RAG modulares

Os pipelines RAG modulares dividem o processo de recuperação e geração em componentes separados e especializados. Isso permite maior flexibilidade e personalização do sistema RAG para atender às necessidades específicas da aplicação.

Um pipeline RAG modular típico pode incluir estágios para expansão de consulta, recuperação, reclassificação e geração, cada um manipulado por um módulo dedicado. Essa abordagem modular permite o uso de modelos ou técnicas especializadas em cada estágio, potencialmente levando a um desempenho geral melhorado.

Os pipelines RAG modulares também facilitam a experimentação com diferentes configurações e a identificação de gargalos ou áreas para melhoria dentro do sistema. Ao otimizar cada módulo de forma independente, os desenvolvedores podem criar sistemas RAG altamente eficientes e eficazes, adaptados ao seu caso de uso específico.

Recuperação de janela de sentença

A recuperação de janela de sentenças foca na recuperação de pedaços menores e mais direcionados de texto, como sentenças individuais ou passagens curtas, em vez de documentos inteiros. Essa abordagem ajuda a reduzir o ruído e a melhorar a relevância do contexto recuperado, levando, em última análise, a respostas geradas mais precisas.

Ao dividir o processo de recuperação de documentos em unidades menores, o sistema RAG pode identificar melhor as informações mais pertinentes para responder à consulta do usuário. Este método é particularmente eficaz para lidar com consultas complexas que exigem detalhes ou fatos específicos.

Conjunto e reclassificação de retrievers

O Retriever Ensembling envolve a combinação de vários modelos de recuperação para melhorar a precisão geral da recuperação. Ao alavancar os pontos fortes de diferentes

abordagens de recuperação, como BERT ou pesquisa semântica , o sistema RAG pode identificar com mais eficiência os documentos mais relevantes.

Técnicas de reclassificação são então aplicadas para refinar ainda mais os resultados recuperados com base em critérios adicionais, como pontuações de relevância ou diversidade. Isso garante que o conjunto de documentos mais informativo e diverso seja selecionado para o estágio de geração de resposta subsequente.

Geração e síntese de respostas

A geração de resposta em sistemas RAG envolve a integração do contexto recuperado com a consulta original do usuário para produzir uma resposta coerente e informativa. Técnicas avançadas, como mecanismos de atenção e planejamento de conteúdo, são empregadas para focar nas partes mais relevantes do texto recuperado e garantir um fluxo lógico de informações na resposta gerada.

Ao alavancar o poder de modelos de linguagem grandes (LLMs) , os sistemas RAG podem sintetizar as informações recuperadas em uma resposta natural e contextualmente relevante. Essa abordagem sofisticada permite que os sistemas RAG lidem com uma gama diversificada de tarefas de raciocínio e forneçam respostas precisas com base no conhecimento disponível.

Refinamento do conhecimento

Técnicas de refinamento de conhecimento visam melhorar a qualidade e relevância das informações recuperadas pelo sistema RAG. Isso pode envolver métodos como vinculação de entidades, que identifica e desambigua entidades nomeadas dentro do contexto recuperado, e integração de gráfico de conhecimento, que incorpora conhecimento estruturado para aprimorar o processo de recuperação.

Ao refinar o conhecimento recuperado, os sistemas RAG podem fornecer respostas mais precisas e informativas às consultas do usuário. Isso é particularmente importante para domínios que exigem um alto nível de precisão, como assistência médica ou finanças , onde a confiabilidade das respostas geradas é crítica.

APÊNDICE 3

Termo de Aceite de Entrega 5

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 31 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

JONAS GOMES DA SILVA JUNIOR

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Neste Gate procurei definir como iria contribuir para a comunidade científica com minhas pesquisas durante a residência, assim iniciei uma pesquisa e nela me deparei com artigos interessantes tais como “Para os químicos, a revolução da IA ainda não aconteceu - [For chemists, the AI revolution has yet to happen](#), 2023”

Com isso, procurei fonte de dados com que eu pudesse trabalhar, e acabei me deparando com o Open Reaction Database (ORD) que é um esquema de infraestrutura de acesso aberto para estruturar e compartilhar dados de reações orgânicas, incluindo um repositório de dados centralizado, iniciado por volta de 2020 e funciona com contribuições abertas até os dias de hoje. [the Open Reaction Database!](#)

Assim, iniciei uma Análise Exploratória dos Dados, além disso, estruturei os dados disponíveis em Jsons para facilitar o início da minha pesquisa utilizando RAG para especializar um modelo linguagem em Reações Químicas.

Os Jsons foram estruturados e compostos por:

- Reaction ID (ID da Reação) - Um identificador único para a reação
- Identifiers (Identificadores) - Contêm informações como o tipo e os detalhes da reação.
- Inputs (Entradas) - Descreve os reagentes e outros componentes envolvidos na reação.
- Conditions (Condições) - Descreve as condições da reação, como temperatura.
- Notes (Notas) - Incluem detalhes do procedimento experimental.
- Outcomes (Resultados) - Contêm informações sobre os produtos da reação.
- Provenance (Proveniência) - Contêm informações sobre a origem do experimento

Assim, iniciei a parte de geração testando como Modelos de Linguagem Grande reagem aos dados antes de serem expostos ao RAG, onde demonstraram que não são tão precisos em perguntas objetivas em relação a reações químicas, portanto, a utilização de técnica de RAG será útil nesse contexto.

Além disso, iniciei uma estratégia de Criação de um Benchmark de modelos para avaliar e comparar a implementação de RAG's com técnicas diferentes, utilizando métricas como F1 Score, BLEU, e ROUGE para avaliar a precisão das respostas geradas, comparando com as respostas desejadas.


Motivação e Definição de Tarefas:

Design inverso é o oposto da abordagem tradicional, onde os químicos normalmente começam com os materiais ou substâncias disponíveis e então investigam suas propriedades. No design inverso, a meta é começar pelos requisitos finais (como a cor, condutividade, solubilidade, ou eficiência catalítica) e trabalhar "para trás" para encontrar moléculas ou combinações de reagentes que possam alcançar esses objetivos.

Resposta a Perguntas - O modelo poderá responder perguntas específicas sobre reações, como: "Qual o rendimento esperado desta reação com um dado catalisador?".

Análise de Condições Experimentais - Solicitar ao modelo uma recomendação sobre as melhores condições experimentais, dado um conjunto de reagentes.

Geração de Descrições de Procedimentos - Solicitar ao modelo a criação de uma descrição de procedimento para uma reação com base nas condições e reagentes fornecidos.

Anotações -  Gate - Anotações.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Trabalhar no Benchmark de modelos para avaliar e comparar a implementação de RAGs com técnicas diferentes, utilizando métricas como F1 Score, BLEU, e ROUGE para avaliar a precisão das respostas geradas, comparando com as respostas desejadas.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: 

[Doc citado no Termo de Aceite de Entrega de 31 de Outubro]

Definição de Escopo

A matéria que foi meu ponto de partida neste trabalho, intitulada “For chemists, the AI revolution has yet to happen”, argumenta que, para fazer previsões químicas precisas, um sistema de IA precisa ter conhecimento suficiente das estruturas químicas específicas com as quais diferentes reações funcionam. Químicos que descobrem uma nova reação geralmente publicam resultados explorando essas reações, mas frequentemente esses dados não são exaustivos. A menos que os sistemas de IA tenham conhecimento abrangente, eles podem acabar sugerindo materiais iniciais com estruturas que impedem reações de funcionar ou que levam a produtos incorretos.

Um exemplo de progresso misto está no que os pesquisadores de IA chamam de "design inverso". Em química, isso envolve começar com propriedades físicas desejadas e, em seguida, identificar substâncias que possuem essas propriedades e que podem, idealmente, ser sintetizadas de forma barata. Por exemplo, o design inverso baseado em IA ajudou cientistas a selecionar materiais ótimos para produzir diodos emissores de luz orgânicos fosforescentes azuis.

Design inverso, no contexto da química e desenvolvimento de materiais, é um processo no qual os pesquisadores começam com as propriedades desejadas de um material ou molécula e, a partir dessas especificações, tentam identificar ou projetar substâncias que possam atingir essas propriedades.

Em outras palavras, o design inverso é o oposto da abordagem tradicional, onde os químicos normalmente começam com os materiais ou substâncias disponíveis e então investigam suas propriedades. No design inverso, a meta é começar pelos requisitos finais (como a cor, condutividade, solubilidade, ou eficiência catalítica) e trabalhar "para trás" para encontrar moléculas ou combinações de reagentes que possam alcançar esses objetivos.

Por exemplo:

OLEDs (Diodos Emissores de Luz Orgânicos): Se um cientista quer desenvolver um novo material para um OLED que emita luz azul e seja estável, ele pode começar com essa especificação. Usando métodos computacionais e inteligência artificial, é possível buscar entre milhares de compostos químicos aqueles que atendem às propriedades físicas desejadas.

Medicamentos: Quando se deseja um fármaco que tenha propriedades específicas, como boa absorção e baixa toxicidade, o design inverso é usado para identificar ou até mesmo criar uma estrutura molecular que atenda a essas especificações.

No campo da inteligência artificial aplicada à química, o design inverso pode ser facilitado pelo uso de algoritmos de aprendizado de máquina que analisam grandes quantidades de dados experimentais, identificando padrões e características que se

relacionam com as propriedades desejadas, e ajudando a propor estruturas candidatas que sejam promissoras.

Assim, o design inverso se torna uma ferramenta poderosa para acelerar a inovação, pois ajuda a diminuir o número de experimentos e tentativas que seriam necessários para encontrar um material com as propriedades desejadas, oferecendo uma abordagem mais eficiente e orientada para a resolução do problema.

Com isso em mente, procurei fontes de dados com as quais eu pudesse trabalhar e acabei encontrando o Open Reaction Database (ORD), que é um esquema e infraestrutura de acesso aberto para estruturar e compartilhar dados de reações orgânicas, incluindo um repositório de dados centralizado, iniciado por volta de 2020 e que funciona com contribuições abertas até os dias de hoje.

O ORD suporta tecnologias convencionais e emergentes, de reações de bancada a experimentos automatizados de alto rendimento e química de fluxo. A visão do projeto é que uma representação de dados consistente e uma infraestrutura que suporte o compartilhamento de dados permitirão aplicações futuras que melhorarão muito o estado da arte em relação ao planejamento de síntese auxiliado por computador, predição de reações e outras tarefas de química preditiva.

O banco de dados cresceu para incluir mais de 2 milhões de reações, incluindo um grande conjunto de dados extraídos de fontes do USPTO, e recebeu contribuições de usuários acadêmicos e industriais, tanto de trabalhos publicados quanto não publicados.

Assim, iniciei uma análise exploratória dos dados e estruturei os dados disponíveis em arquivos JSON para facilitar o início da minha pesquisa, utilizando RAG (Retrieval-Augmented Generation) para especializar um modelo de linguagem em reações químicas.

Os arquivos JSON foram estruturados e compostos por:

1. Reaction ID (ID da Reação): Um identificador único para a reação.
2. Identifiers (Identificadores): Contêm informações como o tipo e os detalhes da reação.
3. Inputs (Entradas): Descrevem os reagentes e outros componentes envolvidos na reação.
4. Conditions (Condições): Descrevem as condições da reação, como temperatura.
5. Notes (Notas): Incluem detalhes do procedimento experimental.
6. Outcomes (Resultados): Contêm informações sobre os produtos da reação.
7. Provenance (Proveniência): Contêm informações sobre a origem do experimento.

Em seguida, iniciei uma estratégia de criação de um benchmark de modelos para avaliar e comparar a implementação de RAG com diferentes técnicas, utilizando métricas como F1 Score, BLEU e ROUGE para avaliar a precisão das respostas geradas, comparando-as com as respostas desejadas.

Motivação e Definição de Tarefas

- Resposta a Perguntas: O modelo deve ser capaz de responder perguntas específicas sobre reações, como: "Qual o rendimento esperado desta reação com um dado catalisador?".
- Análise de Condições Experimentais: Solicitar ao modelo uma recomendação sobre as melhores condições experimentais, dado um conjunto de reagentes.
- Geração de Descrições de Procedimentos: Solicitar ao modelo a criação de uma descrição de procedimento para uma reação com base nas condições e reagentes fornecidos.
- Dessa forma, espero criar uma base para comparar a eficácia de diferentes abordagens e melhorar a precisão do modelo na predição e descrição de reações químicas.

APÊNDICE 4

Termo de Aceite de Entrega 6

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 6 de nov. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

JONAS GOMES DA SILVA JUNIOR

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante este Stage finalizei a estruturação dos dados e migrei para um banco de dados vetorial, para facilitar a integração com o modelo de Embeddings, para isso, escolhi o Qdrant.

Avanços:

- **Estrutura dos Dados:** Todos os arquivos baixados foram salvos em json (371.538)
- **Localção dos dados para um banco de dados vetorial - Qdrant.**

O Qdrant destaca-se como a opção ideal por oferecer:

 - ❖ Cluster, 4gb de espaço e 1gb de RAM interna gratuitamente.
 - ❖ Projetado para alta performance em busca de similaridade.
 - ❖ Possui uma API intuitiva e suporta múltiplas linguagens de programação, facilitando a integração com o pipeline.
 - ❖ Suporte a Filtros e Metadados: Permite a combinação de buscas variadas com filtros baseados em metadados, aumentando a precisão dos resultados.
- **Vetorização:** Início da implementação e do benchmark com o uso do modelo de recuperação **BGE-M3** para transformar cada segmento em vetores, permitindo comparações semânticas no banco vetorial.

Resolução de erros na construção do Pipeline de Recuperação

Anotações -  Entregável

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Finalizar a construção do pipeline de recuperação utilizando o modelo BGE-M3 e prosseguir com o BenchMark para comparar com outros modelos de recuperação.

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: 


[Doc citado no Termo de Aceite de Entrega de 06 de Novembro]

Uso de Dados em um Banco de Dados Vetorial

Cluster Name

rag_quimico

 vCPUs: 0.5  RAM: 1GB  DISK: 4GB x 1 Node

Your Cluster is Healthy 

Migração dos dados para um Banco de Dados Vetorial como o Qdrant para Gerenciamento de Dados de Reações Químicas

Contextualização do Problema

Durante o desenvolvimento deste Gate, foi acumulado um extenso conjunto de dados composto por milhares de arquivos JSON (371.538) detalhando diversas discussões químicas. Cada arquivo contém informações complexas, incluindo identificadores, componentes, condições de ocorrência, procedimentos e resultados.

Limitações de Bancos de Dados Tradicionais

Bancos de dados relacionados ou tradicionais não são adequados para este cenário devido a:

- Complexidade dos Dados
- Necessidade de Busca Semântica : Para responder a perguntas específicas, é necessário realizar pesquisas baseadas em similaridade semântica, algo que bancos de dados tradicionais não suportam de forma eficiente.
- Escalabilidade : O volume de dados exige uma solução que possa escalar horizontalmente sem perda de desempenho.

Vantagens do Uso de um Banco de Dados Vetorial

Os bancos de dados específicos foram desenvolvidos para lidar com a indexação e busca de dados de alta dimensionalidade, permitindo buscas por similaridade com alto desempenho. As principais vantagens incluem:

- Indexação de Embeddings : Permite armazenar vetores de embeddings gerados por modelos de linguagem, facilitando a comparação semântica entre consultas e documentos.
- Busca por Similaridade : Suporta algoritmos de busca aproximados que são altamente eficientes para encontrar itens semelhantes em grandes conjuntos de dados.
- Integração com Modelos de Aprendizado de Máquina : Facilitam a integração com pipelines de IA, essencial para o sistema RAG proposto.

Por que o Qdrant

O Qdrant destaca-se como a opção ideal por oferecer:

- Cluster Gratuito : Oferece a possibilidade de utilizar um cluster gratuitamente, permitindo iniciar o desenvolvimento e os testes sem incorrer em custos adicionais.
- Alto Desempenho : Projetado para alta performance em busca de similaridade, essencial para responder rapidamente às consultas dos usuários.
- Facilidade de Uso : Possui uma API intuitiva e suporta múltiplas linguagens de programação, facilitando a integração com o sistema existente.
- Suporte a Filtros e Metadados : Permite a combinação de buscas variadas com filtros baseados em metadados, aumentando a precisão dos resultados.

Implementação no Projeto

A utilização do Qdrant permite: Armazenamento Eficiente - Os dados das respostas serão convertidos em encaixes aleatórios e armazenados de forma otimizada, além disso uma busca semântica que possibilita que o sistema responda a integração a api, encontrando as respostas mais relevantes com base no contexto fornecido e escalabilidade - apoiará o crescimento do conjunto de dados sem transferência significativa de desempenho.

A adoção de um banco de dados vetorial como o Qdrant é crucial para o sucesso do projeto. Ele não apenas resolve as limitações impostas pelos bancos de dados tradicionais, como também fornece uma infraestrutura robusta para implementar um sistema RAG eficiente. Essa escolha tecnológica se alinha com os objetivos do projeto, garantindo a capacidade de fornecer respostas precisas e rápidas a consultas sobre respostas químicas, potencializando o uso dos modelos de linguagem e o valor do conjunto de dados acumulado.

Próximos Passos: Finalizar a instalação e configuração do Qdrant na API e Integrar o Qdrant com os modelos de recuperação para implementar o sistema de perguntas e respostas.

Referência: Documentação do Qdrant: <https://qdrant.tech/>

APÊNDICE 5

Termo de Aceite de Entrega 7

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 27 de nov. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

JONAS GOMES DA SILVA JUNIOR

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Nesta Semana, tive alguns desafios na execução da minha implementação, porém foram contornados, entre eles:

- Desafios com **Qdrant** na ingestão dos dados utilizando o cluster gratuito
- Desafio com o modelo de Recuperação 'multi-qa-mpnet-base-dot-v1'
- **Validação da Recuperação do RAG** - [Validação RAG](#)
- **Geração de Fluxograma inicial Ponta a Ponta** - https://miro.com/app/board/uXjVL_WBMSl=?share_link_id=983718286698

Fluxo do Sistema Implementado:

1. Processamento da Pergunta

- O usuário faz uma pergunta específica sobre uma **ocorrência química**.

2. Transformação em Vetor de Consulta

- A pergunta é convertida em um **vetor** usando um modelo embedding

3. Busca no Qdrant

- O vetor de consulta é usado para realizar uma busca no Qdrant.

- O Qdrant retorna os documentos mais relevantes com base na similaridade semântica dos vetores.
- Cada documento inclui o payload com informações fornecidas sobre uma ocorrência química.

4. Construção do Contexto

- Os dados recuperados são processados para extrair informações relevantes.
- O contexto pode incluir:
 - Identificadores de reação.
 - Lista de reagentes e produtos.
 - Condições de ocorrência (temperatura, solventes, ocorrências).
 - Resultados e rendimentos.

5. Geração da Resposta

- O contexto e a pergunta são fornecidos ao modelo GPT-4.
- O modelo usa essas informações para gerar uma resposta detalhada e precisa.

6. Resposta ao Usuário

- A resposta gerada é enviada ao usuário.
- Se o contexto não estivesse disponível, a resposta poderia ser genérica ou menos precisa, ou até mesmo nem fornecida por limitação de contexto.

Validação do Contexto (Retrieval):

Perguntas Específicas buscando um resultado de uma reação química específica, por exemplo:

Pergunta : "Qual é o papel da trifetilfosfina ($C_{18}H_{15}P$) na ocorrência com ID ord-5ca9c12179f747729d6cfe74d56c10fe?"

Resposta com RAG : "O papel da trifetilfosfina ($C_{18}H_{15}P$) na fato com ID ord-5ca9c12179f747729d6cfe74d56c10fe é o de reagente. Ela reage com HBr para formar o produto brometo de trifetilfosfônio."

Resposta sem RAG : "A notação que você revelou é uma ocorrência química escrita no formato SMILES, que é uma maneira de representar estruturas químicas em forma textual."

No entanto, sem o contexto específico ou os nomes reais dos compostos, é difícil dizer qual é o papel desse composto na ocorrência.

Geralmente, o papel de um composto de uma ocorrência química pode ser:

- **Reagente** : substância consumida na ocorrência.
- **Produto** : substância formada na ocorrência.
- **Catalisador** : substância que acelera a ocorrência sem ser consumido.
- **Solvente** : substância que apresenta ocorrência.

Se você puder fornecer mais informações sobre as complicações envolvidas ou o tipo de fato, poderei dar uma resposta mais específica.”

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Continuar explorando modelos de embeddings e otimizando meu sistema RAG para alcançar resultados e comparações entre os modelos.

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

LEONARDO ANTÔNIO ALVES: Go! ▾

[Doc citado no Termo de Aceite de Entrega de 27 de Novembro]

Processo de Validação da Utilização do RAG

Introdução

O **RAG** (Retrieval-Augmented Generation) é uma técnica que combina a recuperação de informações relevantes de uma base de dados (recuperação) com a geração de texto por um modelo de linguagem (geração), como o GPT-4. O objetivo é melhorar a qualidade e a precisão das respostas fornecidas pelo modelo, fornecendo-lhe o contexto adicional e específico sobre o assunto em questão.

Utilização do Qdrant no RAG

O **Qdrant** é um banco de dados de vetores otimizado para operações de similaridade semântica. Ele permite armazenar vetores de embutimentos de alta dimensionalidade e buscas eficientes de conexões mais realizadas. No contexto do RAG, o Qdrant é usado para armazenar e recuperar informações relevantes que serão fornecidas como contexto para o modelo de linguagem.

Como o Qdrant Auxilia no Sistema Implementado

1. Armazenamento de Dados em Forma de Vetores

- Possuímos uma coleção de dados sobre **reações químicas**, que inclui detalhes como reagentes, produtos, condições de ocorrência, etc.
- Esses dados são transformados em **vetores de embeddings** usando um modelo de embeddings (por exemplo, no momento *'multi-qa-mpnet-base-dot-v1'*).
- Os vetores e os dados associados (payloads) são armazenados no Qdrant.

2. Busca Semântica Eficiente

- Quando uma pergunta é feita, ela é transformada em um **vetor de embedding** usando o mesmo modelo.
- Esse vetor de consulta é usado para realizar uma busca no Qdrant, que retorna os documentos (reações) desses vetores são semanticamente mais próximos do vetor da pergunta.
- O Qdrant é otimizado para essas operações, permitindo buscas rápidas mesmo em grandes conjuntos de dados.

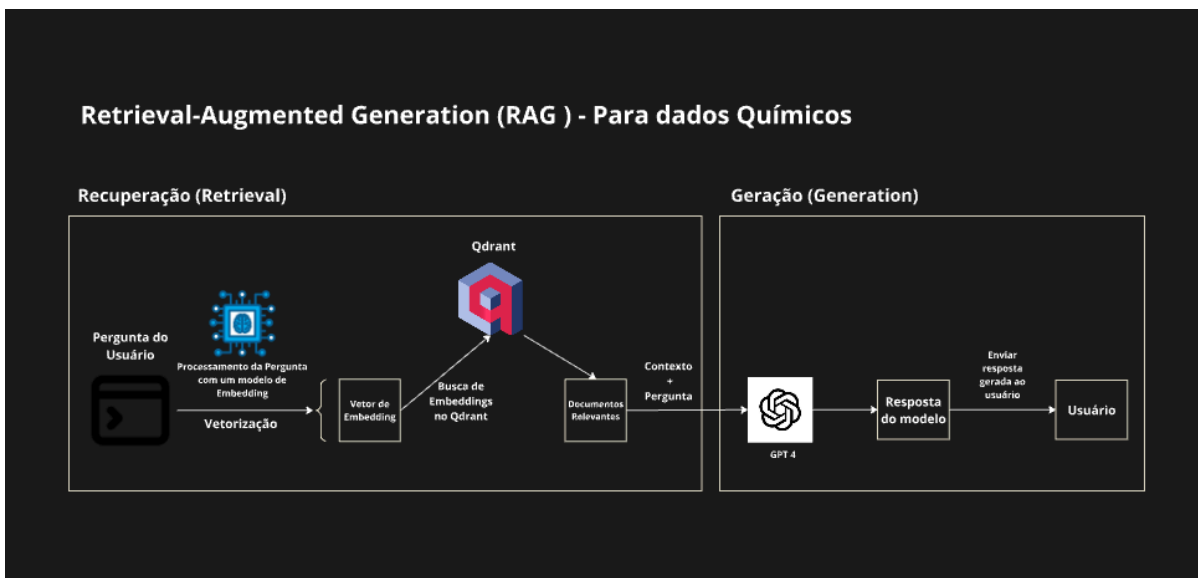
3. Fornecimento de Contexto Relevante

- Os dados recuperados do Qdrant são usados para construir um **contexto detalhado** sobre o assunto da pergunta.
- Esse contexto inclui informações específicas que são relevantes para responder à pergunta do usuário.

4. Melhoria da Resposta do Modelo de Linguagem

- O contexto é fornecido ao modelo GPT-4 junto com a pergunta.
- Com esse contexto adicional, o modelo pode gerar respostas mais precisas, informativas e relevantes.
- Isso é especialmente útil para perguntas que exigem conhecimentos específicos de que o modelo pode não ter sido adquirido durante seu treinamento inicial.

Fluxo do Sistema



Disponível em: https://miro.com/app/board/uXjVL_WBMsI=?share_link_id=983718286698

1. Processamento da Pergunta

- O usuário faz uma pergunta específica sobre uma **ocorrência química**.
- **Exemplo** : "Qual é o papel da trifenilfosfina na ocorrência com ID ord-12345?"

2. Transformação em Vetor de Consulta

- A pergunta é convertida em um **vetor de incorporação** usando o modelo *multi-qa-mpnet-base-dot-v1*.

3. Busca no Qdrant

- O vetor de consulta é usado para realizar uma busca no Qdrant.
- O Qdrant retorna os documentos mais relevantes com base na similaridade semântica dos vetores.
- Cada documento inclui o payload com informações fornecidas sobre uma ocorrência química.

4. Construção do Contexto

- Os dados recuperados são processados para extrair informações relevantes.
- O contexto pode incluir:
 - Identificadores de reação.
 - Lista de reagentes e produtos.
 - Condições de ocorrência (temperatura, solventes, ocorrências).
 - Resultados e rendimentos.

- O contexto é formatado de maneira estruturada para ser entendido pelo modelo GPT-4.

5. Geração da Resposta

- O contexto e a pergunta são fornecidos ao modelo GPT-4.
- O modelo usa essas informações para gerar uma resposta detalhada e precisa.
- **Exemplo de resposta** : "Na ocorrência com ID ord-12345, a trifenilfosfina atua como um reagente nucleófilo que reage com..."

6. Resposta ao Usuário

- A resposta gerada é enviada ao usuário.
- Se o contexto não estivesse disponível, a resposta poderia ser genérica ou menos precisa.

Benefícios do Uso do Qdrant no RAG

- **Eficiência** : O Qdrant é otimizado para buscas rápidas em grandes conjuntos de vetores, garantindo que o sistema responda rapidamente às consultas dos usuários.
- **Escalabilidade** : Pode lidar com milhões de documentos, permitindo que você expanda sua base de conhecimento sem perda de desempenho.
- **Precisão** : Uma busca baseada em similaridade semântica garante que os documentos recuperados sejam relevantes para a pergunta, melhorando a qualidade do contexto fornecido.
- **Flexibilidade** : Suporta atualizações dinâmicas, permitindo que novos dados sejam atualizados ou atualizados conforme necessário.

Exemplos Reais com Diferentes Formas de Perguntar sobre o Mesmo Componente Químico

Exemplo 1

Pergunta : "Qual é o papel da trifenilfosfina na ocorrência com ID ord-5ca9c12179f747729d6cfe74d56c10fe?"

Resposta com RAG : O papel da trifenilfosfina ($C_{18}H_{15}P$) na fato com ID ord-5ca9c12179f747729d6cfe74d56c10fe é o de reagente. Ela reage com HBr para formar o produto brometo de trifenilfosfônio.

Resposta sem RAG : A sequência que você apareceu parece ser uma representação SMILES (Simplified Molecular Input Line Entry System) de uma ocorrência química. Este sistema é usado para descrever a estrutura de espécies químicas usando sequências ASCII.

Infelizmente, sem informações mais específicas sobre a ocorrência (como um nome ou uma referência bibliográfica), é difícil fornecer uma interpretação detalhada do papel desta molécula.

No entanto, posso dizer que a sequência SMILES descreve uma ocorrência entre três moléculas:

1. Uma molécula orgânica complexa contendo carbono (C), hidrogênio (H) e fósforo (P).
2. Uma molécula simples de ácido bromídrico (HBr).
3. Uma molécula orgânica contendo carbono, oxigênio e oxigênio, com um éter e um grupo funcional carbonila.

A reação produz:

1. Um íon brometo (Br⁻).
2. Uma molécula orgânica complexa semelhante à molécula inicial, mas com uma carga positiva no átomo de fósforo.

A notação "[f:4.5]" não é padrão no SMILES e pode ser específico para o software ou banco de dados usado. Poderia representar uma propriedade da ocorrência ou molécula, como uma frequência ou constante de força, mas sem mais informações, é difícil dizer com certeza.

Em geral, o SMILES é um sistema para representar estruturas químicas e não fornece, por si só, informações sobre o papel de cada molécula em um acontecimento. Isso seria determinado pelo químico com base nas condições de fato.

Exemplo 2

Pergunta : "Qual é o papel da trifetilfosfina (C₁₈H₁₅P) na ocorrência com ID ord-5ca9c12179f747729d6cfe74d56c10fe?"

Resposta com RAG : O papel da trifetilfosfina (C₁₈H₁₅P) na fato com ID ord-5ca9c12179f747729d6cfe74d56c10fe é o de reagente. Ela reage com HBr para formar o produto brometo de trifetilfosfônio.

Resposta sem RAG : A notação que você revelou é uma ocorrência química escrita no formato SMILES, que é uma maneira de representar estruturas químicas em forma textual.

No entanto, sem o contexto específico ou os nomes reais dos compostos, é difícil dizer qual é o papel desse composto na ocorrência.

Geralmente, o papel de um composto de uma ocorrência química pode ser:

- **Reagente** : substância consumida na ocorrência.

- **Produto** : substância formada na ocorrência.
- **Catalisador** : substância que acelera a ocorrência sem ser consumido.
- **Solvente** : substância que apresenta ocorrência.

Se você puder fornecer mais informações sobre as complicações envolvidas ou o tipo de fato, poderei dar uma resposta mais específica.

Exemplo 3

Pergunta : "Qual é o papel de [C:1]1(P :7 [C:8]2[CH:13]=[CH:12][CH:11]=[CH:10][CH:9]=2)[CH:6]=[CH:5][CH:4]=[CH:3][CH:2]=1.[BrH:20].C(OCC)C>C1(C)C=CC=CC=1>[Br-:20].[C:14]1(PH +:7 [C:8]2[CH:13]=[CH:12][CH:11]=[CH:10][CH:9]=2)[CH:15]=[CH:16][CH:17]=[CH:18][CH:19]=1 |f:4.5| EU IA ord-5ca9c12179f747729d6cfe74d56c10fe?"

Resposta com RAG : A molécula com a estrutura complexa encontrada corresponde ao reagente trifetilfosfina e ao produto brometo de trifetilfosfônio na ocorrência com ID ord-5ca9c12179f747729d6cfe74d56c10fe. Nesta verdade, a trifetilfosfina (C₁₈H₁₅P) reage com HBr para formar brometo de trifetilfosfônio.

Resposta sem RAG : A sequência que você apareceu é uma representação de uma ocorrência química no formato SMILES. Esta é uma maneira de descrever uma ocorrência química em um formato linear que pode ser interpretada por um computador.

A ocorrência descrita parece ser uma ocorrência de substituição, onde um íon brometo ([Br⁻]) é substituído por um grupo fosfônio ([PH⁺]) em um composto aromático policíclico. Esta ocorrência pode fazer parte de um esquema maior sintético para introduzir um grupo contendo correspondência no composto aromático, que pode ser usado para posterior funcionalização ou como parte de uma descoberta.

A notação "|f:4.5|" não tem um significado padrão na notação SMILES e pode ser específico para o banco de dados ou software que gerou esta sequência. Pode representar algum tipo de classificação ou classificação para esta razão.

Comparação Geral

- **Com RAG** : As respostas são específicas e fornecidas, fornecemos informações precisas sobre o papel do composto na evidência, graças ao contexto adicional fornecido pelo Qdrant.
- **Sem RAG** : As respostas são genéricas e menos precisas, pois o modelo não tem acesso às informações específicas da ocorrência.

Conclusão do uso do Qdrant

A utilização do Qdrant no sistema RAG demonstrou boa performance na qualidade e precisão das respostas fornecidas pelo modelo GPT-4. Ao fornecer contexto relevante e específico, o modelo é capaz de gerar respostas que atendam melhor às necessidades dos usuários, especialmente em domínios especializados como a química.

APÊNDICE 6

Termo de Aceite de Entrega 8

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 4 de dez. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

JONAS GOMES DA SILVA JUNIOR

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Este Stage foi focado em desenvolver a Interface Final, para isso escolhi a ferramenta StreamLit.



RAI - Seu Chatbot Químico

Digite sua pergunta sobre reações químicas:

Obter Resposta

Desenvolvido com [Qdrant](#) e [GPT-4](#).

Elementos principais: campo de entrada de texto para perguntas, botão para gerar respostas, área de exibição dos resultados.

Documentação com explicação da interface: [Inteface - streamlit](#)

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: [Go!](#)

Desenvolvimento da Interface Final

O desenvolvimento da interface para o chatbot para análise de respostas químicas foi realizado utilizando o framework Streamlit para construir uma interface interativa que facilita a interação do usuário com o sistema. A interface foi projetada para permitir que o usuário faça perguntas específicas relacionadas às respostas químicas e receba respostas fornecidas e contextualizadas. Além disso, ela oferece a possibilidade de visualizar os dados recuperados e analisar o contexto utilizado para gerar as respostas, proporcionando maior transparência e interação.



RAI - Seu Chatbot Químico

Digite sua pergunta:

Qual é a ligação entre hidrogenio e oxigenio para formação da água?

Obter Resposta

Resposta com RAG

A ligação entre hidrogênio e oxigênio para formar água é uma ligação covalente.

Na água (H₂O), cada átomo de oxigênio forma duas ligações covalentes com dois átomos de hidrogênio. Na ligação covalente, os átomos compartilham elétrons para atingir a configuração de gás nobre e, assim, tornar-se estável.

O oxigênio tem 6 elétrons em sua camada mais externa e precisa de mais 2 elétrons para atingir a estabilidade (configuração de gás nobre). Cada hidrogênio, por outro lado, tem 1 elétron em sua camada mais externa e precisa de mais 1 elétron para atingir a estabilidade. Portanto, cada hidrogênio compartilha seu elétron com o oxigênio e, em troca, o oxigênio compartilha um de seus elétrons com cada hidrogênio. Isso resulta em uma molécula de água estável com ligações covalentes entre os átomos de hidrogênio e oxigênio.

Desenvolvido com [Qdrant](#) e [GPT-4](#).

```
Inserção completa! Total de documentos inseridos: 100000  
Total de documentos na coleção 'chemical_reactions': 100550  
WARNING:root:Unable to close http connection. Connection was interrupted on the server side
```

Print do prompt expando a quantidade de documentos presentes no cluster: 100.550 documentos.

A base do sistema está estruturada em um banco de dados vetorial hospedado no Qdrant, que contém um total de 100.000 arquivos de reações químicas. Os dados foram inseridos no Qdrant em formato JSON, sendo previamente estruturados com informações essenciais, como identificadores de soluções, reagentes, condições experimentais,

resultados e proveniência. A vetorização dos dados foi realizada utilizando o modelo BGE-M3 e multi-qa-mpnet-base-dot-v1 , permitindo uma recuperação eficiente baseada em similaridade semântica.

A interface desenvolvida no Streamlit possui funcionalidades intuitivas. O usuário insere suas perguntas por meio de um campo de entrada, e o sistema processa a consulta convertendo-a em vetores. Esses vetores são enviados ao Qdrant, que busca os documentos mais relevantes com base na semelhança com a consulta do usuário. O pipeline do chatbot combina os documentos recuperados para construir um contexto detalhado, que é então utilizado por um modelo de linguagem (GPT) para gerar uma resposta coerente e informativa. Após esse processo, uma resposta é exibida na interface, juntamente com uma opção de visualizar os documentos recuperados para análise adicional.

Durante o desenvolvimento, foram enfrentados alguns desafios técnicos. Um deles foi relacionado à ingestão de um grande volume de dados no cluster gratuito do Qdrant, o que exigiu configurações e otimizações para garantir o desempenho esperado. Outro desafio foi a integração entre a interface do Streamlit e o backend do sistema, especialmente para minimizar a latência na troca de dados e manter a interatividade do chatbot.

Os resultados obtidos foram positivos. O chatbot demonstrou a capacidade de responder de forma precisa e contextualizada a perguntas relacionadas a reações químicas, aproveitando a estrutura do Qdrant para realizar buscas rápidas e eficazes. A interface do Streamlit mostrou-se uma solução eficiente para fornecer uma experiência amigável ao usuário, permitindo tanto a interação direta com o sistema quanto a análise detalhada dos dados recuperados.

Este trabalho conclui o desenvolvimento de uma solução completa para análise e interação com dados de reações químicas, evidenciando a aplicabilidade prática de técnicas de Retrieval-Augmented Generation (RAG) e demonstrando a sinergia entre ferramentas como Streamlit e Qdrant para resolver problemas específicos com eficiência e precisão.