Gabriel Matheus Santos de Jesus

A aplicação de modelos LLM para facilitar a busca em páginas de perguntas frequentes (FAQs)

Universidade Federal de Pernambuco
graduacao@cin.ufpe.br
portal.cin.ufpe.br/graduacao

Recife
2024

Gabriel Matheus Santos de Jesus

A aplicação de modelos LLM para facilitar a busca em páginas de perguntas frequentes (FAQs)

Trabalho apresentado ao Programa de Graduação em Engenharia da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Engenharia da Computação.

Área de Concentração: *Processamento de Linguagem Natural*
Orientador: *Luciano de Andrade Barbosa*

Recife
2024

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Jesus, Gabriel Matheus Santos de.
A aplicação de modelos LLM para facilitar a busca em páginas de perguntas frequentes (FAQs) / Gabriel Matheus Santos de Jesus. - Recife, 2024.
42 p : il., tab.

Orientador(a): Luciano de Andrade Barbosa
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Informática, Engenharia da Computação - Bacharelado, 2024.

1. Grandes Modelos de Linguagem. 2. Perguntas Frequentes. 3. Geração de Respostas. 4. Web Scraping. 5. Recuperação de informação. I. Barbosa, Luciano de Andrade. (Orientação). II. Título.

000 CDD (22.ed.)

Gabriel Matheus Santos de Jesus

A aplicação de modelos LLM para facilitar a busca em páginas de perguntas frequentes (FAQs)

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia da Computação da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de bacharel em Engenharia da Computação.

Aprovado em: 20 / 03 / 2024

BANCA EXAMINADORA

_____

Prof. Dr. Luciano de Andrade Barbosa (Orientador)

Universidade Federal de Pernambuco

_____

Prof. Dr. Vinicius Cardoso Garcia (Examinador Interno)

Universidade Federal de Pernambuco

Quero dedicar esse trabalho à minha família e amigos, cujo apoio incondicional e amizade foram essenciais para alcançar este marco em minha jornada acadêmica.

AGRADECIMENTOS

Quero expressar minha sincera gratidão a todos que contribuíram para a realização deste trabalho. com certeza este não teria sido possível sem o apoio, orientação e incentivo de muitas pessoas incríveis.

Primeiramente, quero agradecer ao meu orientador Luciano Barbosa, pela orientação dedicada, paciência e valiosos insights ao longo deste processo. Sua expertise e feedback foram indispensáveis para o desenvolvimento deste trabalho.

À minha família e amigos, meu profundo agradecimento pela compreensão, incentivo constante e apoio emocional. Vocês foram a base sólida que me permitiu superar desafios e seguir em frente.

Aos colegas de classe, pela troca de experiências, debates construtivos e amizade ao longo desta jornada acadêmica, agradeço sinceramente.

Por último, mas não menos importante, expresso minha gratidão para todas as pessoas que, direta ou indiretamente, contribuíram para este trabalho. Cada pequeno gesto e palavra de encorajamento não passaram despercebidos.

Este trabalho é dedicado a todos vocês, que de alguma maneira, fizeram parte dessa trajetória acadêmica. Obrigado por tornarem este processo enriquecedor e inesquecível.

RESUMO

Diante das constantes inovações e crescentes possibilidades proporcionadas pelos *Large Language Models* (LLM), é cada vez mais comum que as pessoas busquem assistência para questões cotidianas através de simples perguntas, em vez de percorrer extensas páginas online em busca de respostas. Esse cenário é particularmente aplicável à busca por informações em páginas de *Frequently Asked Questions* (FAQ)s, cada vez mais presentes em diversos serviços. Este trabalho visa aprimorar a interação desses usuários, simplificando a consulta às informações nas páginas de FAQs. A proposta é permitir que os mesmos digitem suas perguntas, permitindo que o LLM compreenda o contexto e gere respostas claras e concisas relacionadas à necessidade informada de forma rápida e direta.

Palavras-chave: Grandes Modelos de Linguagem, Perguntas Frequentes, Geração de Respostas, Web Scraping, Recuperação de informação

ABSTRACT

In the face of constant innovations and growing possibilities provided by Large Language Models (LLM), it is increasingly common for people to seek assistance with everyday issues through simple questions, rather than navigating extensive online pages in search of answers. This scenario is particularly applicable to the search for information on Frequently Asked Questions (FAQs) pages, which are increasingly present in various services. This work aims to enhance the interaction of these users by simplifying the process of querying information on FAQs. The proposal is to allow users to type their questions, enabling the LLM to understand the context and generate clear and concise responses related to the informed need quickly and directly.

Keywords: Large Language Models, Frequently Asked Questions, Answer Generation, Web Scraping, Information Retrieval

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ACRÔNIMOS

ANN	*Aproximate Nearest Neighbors*
CSV	*Comma-Separated Values*
DOM	*Document Object Model*
ER	Expressões Regulares
FAISS	*Facebook AI Similarity Search*
FAQ	*Frequently Asked Questions*
HNSW	*Hierarchical Navigable Small World*
HTML	*HyperText Markup Language*
IA	Inteligência Artificial
JSON	*JavaScript Object Notation*
LCS	*Longest Common Subsequence*
LLM	*Large Language Models*
LSH	*Locality Sensitive Hashing*
PLN	Processamento de Linguagem Natural
URL	*Uniform Resource Locator*
XML	*eXtensible Markup Language*

SUMÁRIO

1 INTRODUÇÃO

1.1 MOTIVAÇÃO E JUSTIFICATIVA

Atualmente, a interseção entre tecnologia e a vida cotidiana está se aprofundando de maneira significativa. Uma das inovações que vem redefinindo essa relação são os *Large Language Models* (LLM), os quais além de várias outras capacidades impressionantes, conseguem compreender a linguagem humana e gerar respostas concisas e precisas [3]. Essas transformações impactam diretamente a forma como acessamos e interagimos com informações. Segundo pesquisa, mais de 70% dos consumidores de IA generativa confiam no conteúdo criado [4], marcando assim um ponto de inflexão no comportamento digital global.

Essa transformação é evidenciada pela crescente preferência em recorrer aos LLMs para esclarecimentos rápidos, em vez de buscar extensivamente por artigos online, segundo pesquisa, usuários não estão satisfeitos com resultados do google [5]. Além disso, um estudo revela que cerca de 60% dos brasileiros entrevistados já optam por soluções de IA em suas pesquisas e mais de 80% confiam nos resultados gerados por IA [6], um testemunho da busca por respostas imediatas e precisas. Essas mudanças não apenas destacam o quão eficientes os LLMs podem ser em fornecer informações pertinentes, mas também aponta para uma nova oportunidade: a necessidade de navegar por dados volumosos e frequentemente complexos, e obter respostas rápidas e diretas.

Dessa forma, esta oportunidade se torna ainda mais evidente quando se considera a busca por respostas em páginas de *Frequently Asked Questions* (FAQ). Nessas páginas, que são cada vez mais comuns em uma vasta diversidade de serviços online, a complexidade e o volume de informações podem tornar a pesquisa tradicional demorada e muitas vezes frustrante.

1.2 OBJETIVOS

Em um ambiente digital onde a busca por informações rápidas e precisas se tornou uma necessidade, este trabalho apresenta uma aplicação que utiliza Modelos LLM para melhorar a maneira como interagimos com FAQs online. O objetivo é desenvolver uma solução baseada em LLMs que responda eficientemente às perguntas feitas pelos usuários, fornecendo informações

claras, diretas e relevantes, facilitando o acesso a dados essenciais que, de outra forma, só poderiam ser obtidos apenas através de buscas extensas e muitas vezes morosas.

Neste contexto, o usuário inicia uma interação por meio de uma pergunta formulada usando linguagem natural, onde essa pergunta será submetida ao sistema, cujo foi construído em torno da interação com um LLM, este por sua vez, analisa a pergunta do usuário e a interpreta com base no contexto previamente indexado em seu banco de dados. E como saída, irá retornar uma resposta direta e concisa utilizando também linguagem natural.

Com isso, a proposta deste trabalho visa não apenas atender a essa demanda crescente por praticidade, mas também proporcionar uma experiência mais eficiente na busca por informações específicas. O desafio de percorrer extensos conteúdos em FAQs para encontrar respostas relevantes motiva o desenvolvimento de uma solução utilizando modelos LLM de forma acessível e descomplicada.

2

FUNDAMENTAÇÃO

Neste capítulo, objetiva-se elucidar e discorrer sobre os conceitos fundamentais que embasam a metodologia adotada neste estudo. A exposição detalhada das técnicas e ferramentas empregadas visa proporcionar ao leitor uma compreensão completa e aprofundada dos métodos utilizados, permitindo assim uma visão integral dos procedimentos implementados ao longo deste trabalho.

2.1 *SCRAPING*

A técnica de *scraping*, no contexto de coleta de dados para pesquisa, se refere à extração automatizada de informações de páginas web [7]. Este procedimento utiliza scripts ou programas específicos para navegar por páginas da web, identificar elementos relevantes e extrair dados estruturados a partir do código fonte. Essa abordagem é fundamental para obter grandes volumes de dados de forma eficiente, superando as limitações impostas por métodos manuais. A automatização proporcionada pelo *scraping* permite a obtenção rápida e sistemática de informações, sendo crucial para a análise de grandes conjuntos de dados disponíveis online.

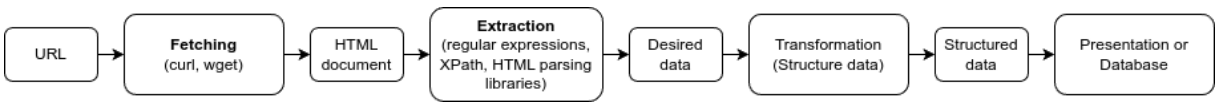

Figura 2.1: Processo de *Web Scraping* [1]

O fluxograma da técnica de *scraping*, representado na figura 2.1, tem início a partir de uma URL, estabelecendo o ponto inicial para a coleta de dados. Nessa etapa, a URL é inserida como parâmetro para uma ferramenta de recuperação de dados (*fetching*), como curl [8] ou wget [9], que realiza a ação de obter o conteúdo HTML da página web associada à URL especificada. O resultado desse processo é o documento HTML que contém as informações da página.

A fase subsequente envolve a extração de dados do documento HTML. Isso é realizado por meio de técnicas ou processos como Expressões Regulares (ER), XPath, HTML parsing, etc.

Cada procedimento é aplicado de maneira específica para identificar e isolar os dados relevantes presentes no código fonte da página web. Essa etapa é crucial para extrair de forma precisa as informações desejadas, como textos, links, imagens ou qualquer outro elemento identificado como relevante.

Os dados extraídos passam, então, pela etapa de transformação. Nessa fase, os dados são estruturados e organizados em conformidade com as necessidades do projeto. Isso pode incluir a limpeza de dados, conversão de formatos ou qualquer outra manipulação necessária para assegurar a uniformidade e a utilidade dos dados [1].

O resultado desse processo é representada pelos dados estruturados, que podem ser encaminhados para a apresentação visual ou armazenamento em um banco de dados, conforme os requisitos especificados.

Nesse contexto, a implementação de módulos se revela estratégica para a fase de obtenção dos dados. Durante o desenvolvimento deste trabalho, destacaram-se certas bibliotecas em virtude das suas características específicas, tais como pontos fortes e fracos, cujo estes serão detalhados melhor nas subseções abaixo.

2.1.1 *Requests*

Dentro etapa de recuperação dos dados é importante citar a biblioteca *Requests* [10] que permite simplificar a realização de solicitações HTTP em *Python*. Ela se mostra ideal para acessar e baixar conteúdo de páginas web que não requerem interação complexa com o usuário (como clicar em botões ou preencher formulários para carregar dados). otimizando assim a aquisição de conteúdos precisam de uma forma ágil e direta para coletar informações estáticas, que em alguns casos pode ser um requisito fundamental.

2.1.2 *Selenium*

Ainda dentro do escopo de recuperação de informações, temos o *Selenium* [11] que é uma biblioteca para automação em navegadores web que permite simular a navegação de um usuário real. Onde ao contrário da *Requests*, que foi citada anteriormente, é particularmente útil para interagir com páginas web que carregam conteúdo dinamicamente com *JavaScript* [12], o que pode ser um desafio para métodos de coleta de dados que dependem apenas de solicitações HTTP estáticas. Ao usar este módulo, é possível assegurar a captura integral das informações disponíveis após a página ser renderizada e estar disponível para coleta, o que pode ser indispensável em aplicações cujo a completude dos dados seja mais importante do que a velocidade de obtenção.

2.1.3 *XPath*

No âmbito de extração das informações o *XPath* [13] desempenha um papel central, sendo aplicável para a navegação em elementos e atributos de documentos XML, com extensão para utilização em documentos HTML. A aplicação mais eficaz de expressões *XPath* reside na localização precisa de caminhos, sendo empregado para a identificação e iteração em conjuntos de nós no documento. O caminho base é designado pela barra "/", equivalente à raiz, como no sistema de arquivos Unix. [14]

Sendo assim essa ferramenta é essencial para a extração precisa de dados específicos de páginas web. Ao simplificar a identificação de elementos relevantes, como perguntas e respostas em uma página de FAQ, elimina-se a necessidade de processamento manual do documento HTML completo. Com isso, os desenvolvedores podem criar consultas detalhadas que filtram conteúdo com base em estrutura, atributos e texto, resultando em uma extração altamente seletiva e alinhada ao foco do projeto.

Dessa maneira, a integração conjunta das tecnologias apresentadas viabiliza a recuperação eficiente e escalável de informações. Testes alinhados aos requisitos do projeto devem determinar a abordagem para a implementação desta fase de coleta de dados na web, finalizando dessa forma, o escopo inicial e permitindo a progressão para as etapas subsequentes de processamento e armazenamento dos dados obtidos.

2.2 PRÉ PROCESSAMENTO DO TEXTO

Após a fase de *scraping*, o pré-processamento do texto é fundamental para preparar o conteúdo para análise por modelos de aprendizado de máquina [15].

2.2.1 Limpeza dos dados

Nesta primeira etapa, pode haver remoção de conteúdo coletado indevidamente, garantindo a eliminação de dados irrelevantes ou descontextualizados, além de normalizar o texto. Isso pode envolver remover caracteres especiais e acentuações, e padronizar o uso de letras maiúsculas e minúsculas, com a intenção de minimizar discrepâncias e aumentar a uniformidade e consistência dos dados.

Além disso, também é realizada a estruturação dos documentos obtidos em formatos apropriados (como CSV, JSON ou TXT) facilitando a manipulação e análise dos dados [16]. A correta estruturação dos dados em formatos padronizados é de suma importância para o processamento eficiente e a análise subsequente de acordo com a finalidade do projeto, assegurando que a informação esteja acessível de forma simples e fácil.

Essas ações são imprescindíveis para mitigar ruídos nos dados e realçar características relevantes, otimizando o desempenho dos modelos, garantindo uma interação mais eficiente com

os algoritmos de aprendizado subsequentes, o que é essencial para maximizar os resultados dos modelos experimentados.

De maneira complementar, a tokenização e a geração de embeddings, cujo estão melhor descritos nas subseções 2.2.2 e 2.2.3, são processos fundamentais no campo de Processamento de Linguagem Natural (PLN), desempenhando papéis essenciais na representação e compreensão de dados textuais.

2.2.2 Tokenização

Após a fase de limpeza e padronização dos dados, a tokenização é uma etapa essencial na preparação de documentos para indexação e processamento por sistemas de Inteligência Artificial (IA). Essa fase envolve a segmentação de textos em unidades discretas, conhecidas como *tokens*, abrangendo palavras, símbolos ou frases, com o intuito de facilitar análises subsequentes e permitir uma representação mais granular das informações textuais.

Existem diferentes estratégias de tokenização, que podem variar desde abordagens simples até modelos mais sofisticados. Estratégias baseadas em regras e padrões incluem a separação por espaços, o tratamento de pontuações como *tokens* independentes, a quebra por hífen, a consideração de números como *tokens* individuais, entre outras. Essas regras simples combinadas são eficazes em contextos nos quais a complexidade textual é relativamente baixa [17].

Entretanto, a tokenização também pode ser realizada por meio de modelos pré-treinados, que aplicam algoritmos específicos para identificar e isolar unidades textuais. Esses modelos possuem a capacidade de lidar com nuances mais complexas do texto, adaptando-se a diferentes contextos linguísticos e fornecendo representações vetoriais numéricas mais sofisticadas.

2.2.3 Geração de *Embeddings*

Logo após a obtenção dos *tokens*, é importante mapeá-los em representações vetoriais em um espaço multidimensional, este processo é conhecido como de geração de *embeddings*. Esses vetores desempenham um papel fundamental na captura de relações semânticas intrínsecas entre as palavras, além de capturar características essenciais de objetos como textos, imagens e sons [18].

Diversas estratégias são empregadas na criação de *embeddings*, cada uma com abordagens específicas. Estratégias clássicas incluem a representação *Bag of Words* (BOW), que cria vetores baseados na frequência de palavras em um documento, e o TF-IDF, que pondera a importância de palavras levando em consideração sua frequência global [19]. Além disso, estratégias mais sofisticadas incluem modelos pré-treinados, como o *BERT*, que têm se destacado na aprendizagem de *embeddings* contextualizados, considerando relações semânticas em nível de frase e contexto global [20].

Essas estratégias resultam em representações vetoriais onde palavras semanticamente similares ocupam posições próximas no espaço multidimensional, enquanto palavras com significados diferentes ocupam posições mais distantes, refletindo eficientemente a similaridade semântica [21].

2.3 BANCO DE DADOS VETORIAL

Considerando que os dados foram processados e transformados em representações vetoriais, é necessário armazená-los em algum lugar. Desse modo, um banco de dados de vetores surge como uma excelente opção, pois se trata de uma infraestrutura avançada projetada especificamente para armazenar e gerenciar essas representações numéricas de alta dimensionalidade.

Para otimizar a recuperação de informação em grandes conjuntos de dados, os bancos de dados de vetores recorrem a técnicas sofisticadas de indexação, como por exemplo, as árvores KD, que são amplamente empregadas, organizando vetores no espaço multidimensional e permitindo uma recuperação eficiente por meio de consultas baseadas em proximidade ou similaridade. Além disso, técnicas de *hashing*, como o *Locality Sensitive Hashing* (LSH), são incorporadas para otimizar consultas específicas, acelerando a localização de dados similares [18].

Para a recuperar essas informações, vários algoritmos podem ser aplicados, no entanto, podemos destacar o *Hierarchical Navigable Small World* (HNSW) por sua eficiência em explorar espaços vetoriais de alta dimensionalidade. Utilizando grafos, o HNSW constrói uma estrutura hierárquica que facilita a navegação entre pontos semelhantes no espaço vetorial [18]. Outro algoritmo notável é o *Aproximate Nearest Neighbors* (ANN), que se concentra na identificação da vizinhança mais próxima de um ponto específico. permitindo a localização rápida dos vetores mais próximos a uma consulta, sem a necessidade de examinar toda a base de dados. Isso é alcançado devido ao objetivo não ser descobrir o ponto exato mais próximo, mas sim, um ponto que esteja localizado próximo o bastante [22].

2.4 MODELOS DE LINGUAGENS

A seleção dos modelos de linguagem para este projeto, baseou-se na popularidade e no potencial de contribuição na comunidade científica. Modelos como Falcon7b, Mixtral7b, Phi-1.5 e Falcon40b foram considerados. Além disso, a escolha de plataformas como *LangChain* e *HuggingFace*, ambas destacadas por oferecerem acesso a modelos de IA de ponta e ferramentas de desenvolvimento, foi estratégica. *LangChain* facilita a integração de capacidades de linguagem em aplicações, enquanto *HuggingFace* é conhecida por sua vasta biblioteca de modelos pré-treinados e sua comunidade ativa. Essa combinação permite a implementação eficaz de soluções avançadas em PLN, aproveitando modelos de linguagem de última geração para tarefas específicas, otimizando assim o desempenho do sistema em compreensão e geração de texto.

2.5 *TRANSFORMERS* E LLMS

Para detalhar melhor o funcionamento dos modelos de linguagem usados aqui, temos os *transformers*, que são arquiteturas de redes neurais que ganharam destaque significativo na área de Processamento de Linguagem Natural (PLN). Como exibido na figura 2.2, os *transformers* se destacam por sua capacidade de lidar eficientemente com sequências de dados. De uma maneira geral, arquitetura dos *transformers*, é composta por dois módulos principais, o *encoder* e o *decoder*. O *encoder* é responsável por processar a entrada, enquanto o *decoder* tem o objetivo de gerar saídas sequenciais [2].

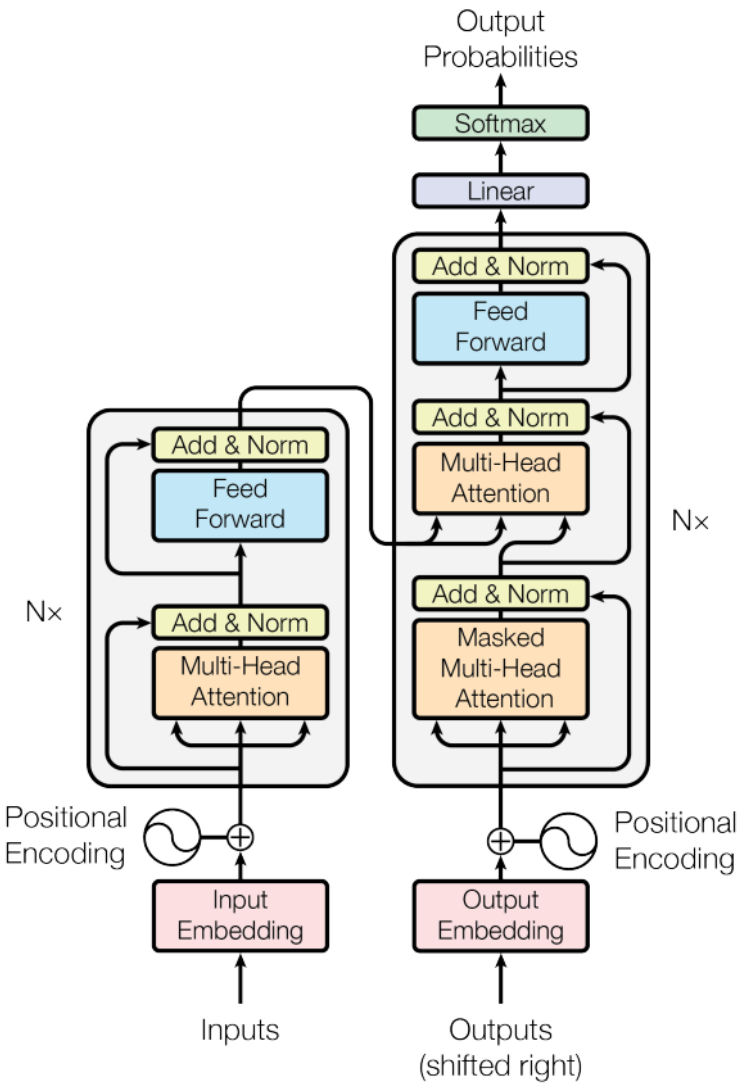

Figura 2.2: Arquitetura dos *Transformers* [2]

Sendo frequentemente implementados utilizando a arquitetura de transformers, os *Large Language Models* (LLM) são modelos que destacam-se pela sua habilidade de realizar diversas tarefas na área de PLN, onde estes, fazem uso do mecanismo de *self-attention*, o que torna possível que o modelo capture de maneira eficiente a relevância contextual em textos exten-

sos, proporcionando uma compreensão mais abrangente e precisa das relações semânticas e gramaticais presentes [2].

Esses modelos, como o próprio nome menciona, são reconhecidos pela capacidade de lidar com volumes textuais muito grandes, onde destacam-se pela sua versatilidade. Sua habilidade de processar eficientemente vastas quantidades de texto os torna aplicáveis em diversas tarefas, desde tradução automática, sumarização de textos e até desafios mais complexos, como geração de texto automatizado, desenvolvimento de chatbots, compreensão contextual, etc [23].

2.6 FERRAMENTAS OPEN SOURCE UTILIZADAS

As ferramentas de código aberto utilizadas, conforme discutido na seção 2.4, mostraram-se essenciais para realização das tarefas descritas na metodologia. Estas serão melhor detalhadas nas subseções abaixo.

2.6.1 *HuggingFace*

O *HuggingFace* constitui um repositório abrangente de modelos pré-treinados destinados a uma variedade de tarefas associadas ao PLN, incluindo tradução, classificação de texto e geração textual. Isso tornou possível a execução dos modelos selecionados, onde a facilidade de carregamento e a possibilidade de adaptá-los para necessidades específicas representam vantagens significativas. Estas características permitem uma aceleração no processo de desenvolvimento e oferecem a capacidade de experimentar com diversos modelos para identificar aquele que apresenta o desempenho desejado para as tarefas em questão.

2.6.2 *LangChain*

Focada na aplicação de capacidades linguísticas em sistemas computacionais, A *LangChain* facilita a integração de tecnologias de PLN em aplicações de forma mais acessível. Como exemplo, a integração com os modelos disponíveis na plataforma *HuggingFace*, seu uso implica na compreensão de como incorporar modelos de linguagem em fluxos de trabalho lógicos e coerentes, viabilizando a implementação de sistemas capazes de entender, processar e gerar linguagem natural de forma eficiente.

2.6.3 *Google Colab*

O *Google Colab* é um ambiente de desenvolvimento em nuvem fundamentado em *Jupyter notebooks*. Ele oferece acesso gratuito a CPUs e GPUs, tornando mais simples a realização de experimentos e a execução das aplicações, como análises textuais e geração de respostas. Esse recurso permite a execução de códigos com recursos computacionais substanciais, eliminando a necessidade de uma infraestrutura local, o que agiliza e flexibiliza o processo de desenvolvimento.

2.7 MÉTRICAS DE AVALIAÇÃO

Após a realização do processamento textual e a geração de respostas, é fundamental estabelecer uma forma para avaliar a qualidade dos resultados, visando uma posterior análise comparativa. Os métodos de avaliação serão descritos neste capítulo.

2.7.1 *BLEU Score*

O *BLEU Score* (*Bilingual Evaluation Understudy*), que foi originalmente criado para mensurar a qualidade de traduções automáticas, também é amplamente aplicado na avaliação de respostas geradas por LLM [24]. Efetuando essa avaliação ao comparar o texto gerados por máquinas com textos de referência usando a correspondência de N-gramas, onde geralmente são usados os valores 1, 2, 3, 4 (unigrama, bigramas, trigramas e quadrigrama). Essa métrica avalia tanto a precisão dos N-gramas, essenciais para a fluidez textual, quanto aplica penalidades a traduções excessivamente curtas, assegurando a fidelidade ao comprimento e à complexidade do texto original.

A fórmula do BLEU Score é expressa como:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{2.1}$$

onde:

- BP é o fator de penalidade de brevidade,
- N é a ordem máxima dos n-gramas considerados,
- w_n é o peso atribuído aos n-gramas de ordem n,
- p_n é a precisão dos n-gramas.

onde p_n representa a precisão dos N-gramas, w_n são os pesos dados a cada tamanho de n-grama, e BP é o fator de penalidade por brevidade, que corrige a tendência de favorecer respostas mais curtas. O valor final varia de 0 a 1 onde valores mais altos indicam uma maior correspondência com os textos de referência.

2.7.2 *ROUGE Score*

O *ROUGE Score* (*Recall-Oriented Understudy for Gisting Evaluation*) é uma métrica criada inicialmente para analisar a qualidade de resumos [25]. Ele realiza essa avaliação comparando a sobreposição de n-gramas, sequências de palavras e pares de palavras, entre o resumo gerado automaticamente e um ou mais textos de referência. Esta comparação permite medir tanto

o *recall* quanto a precisão das informações contidas no resumo, assegurando que os aspectos cruciais do texto original sejam adequadamente refletidos.

Quando aplicado a LLMs, o *ROUGE Score* serve como uma ferramenta essencial para mensurar a capacidade desses modelos de gerar resumos que não apenas condensam o conteúdo original de maneira fiel, mas também mantêm a relevância e a integridade das informações principais. A métrica *ROUGE* possui algumas variantes, como *ROUGE-N*, *ROUGE-L*, e *ROUGE-S*, cada uma focando em diferentes aspectos do texto. Neste trabalho será utilizada a métrica *ROUGE-L*, visto que diferente de algumas abordagens de comparação textual que demandam a sucessiva conexão entre palavras, o *ROUGE-L* se dedica a identificar correspondências sequenciais, permitindo assim, lidar com variações na ordem das palavras. Essa abordagem visa avaliar a coesão nos textos gerados de maneira mais eficiente.

ROUGE-L utiliza a *Longest Common Subsequence* (LCS) para avaliar a sobreposição, focando na sequência mais longa de palavras que aparece tanto no resumo gerado quanto nos textos usados como referência, destacando a estrutura e a ordem das palavras.

A equação do ROUGE-L é dada por:

$$\text{ROUGE-L} = \frac{(1+\beta^2) \cdot R_{\text{lcs}} \cdot P_{\text{lcs}}}{\beta^2 \cdot R_{\text{lcs}} + P_{\text{lcs}}}$$

Onde:

$$R_{\text{lcs}} = \frac{\text{LCS}(X,Y)}{m}$$

$$P_{\text{lcs}} = \frac{\text{LCS}(X,Y)}{n}$$

De modo que $\text{LCS}(X,Y)$ é o comprimento da subsequência comum mais longa de X e Y, onde X representa o texto de referência e Y o texto gerado, e $\beta = P_{\text{lcs}}/R_{\text{lcs}}$.

As ferramentas e métodos apresentados até aqui, constituem a estrutura subjacente para as execuções e experimentos descritos na metodologia. Esses procedimentos permitem uma comparação entre diferentes abordagens, desde a obtenção de informações a partir da técnica de *scraping*, até a a avaliação dos resultados produzidos pelos modelos em questão, por meio da aplicação de métricas destinadas a uma análise relativa, com o intuito de alcançar os melhores resultados.

3

METODOLOGIA

Neste capítulo, serão descritas cada uma das fases do projeto, conforme ilustrado na Figura 3.1. De maneira geral, o fluxo tem início com um processo que executa apenas uma vez, cujo é responsável pela coleta dos dados do domínio-alvo, em sequência prossegue com a etapa de limpeza dos dados e em seguinda ocorre a tokenização e geração de *embeddings* para cada entrada do documento baixado. Posteriormente, realiza-se a indexação dentro do banco de dados, marcando a conclusão do estágio inicial.

Ao alcançar esse ponto, o usuário pode enviar uma pergunta como entrada. Essa entrada passa pelo processo de tokenização e geração de *embeddings* para ser consultada no banco de dados. Este, por sua vez, recebe a consulta e, a partir do conteúdo previamente indexado, recupera o contexto mais similar. Em seguida, retorna ao processo de geração de resposta pelo LLM. A resposta é gerada e então enviada ao usuário, encerrando o ciclo e permitindo um recomeço.

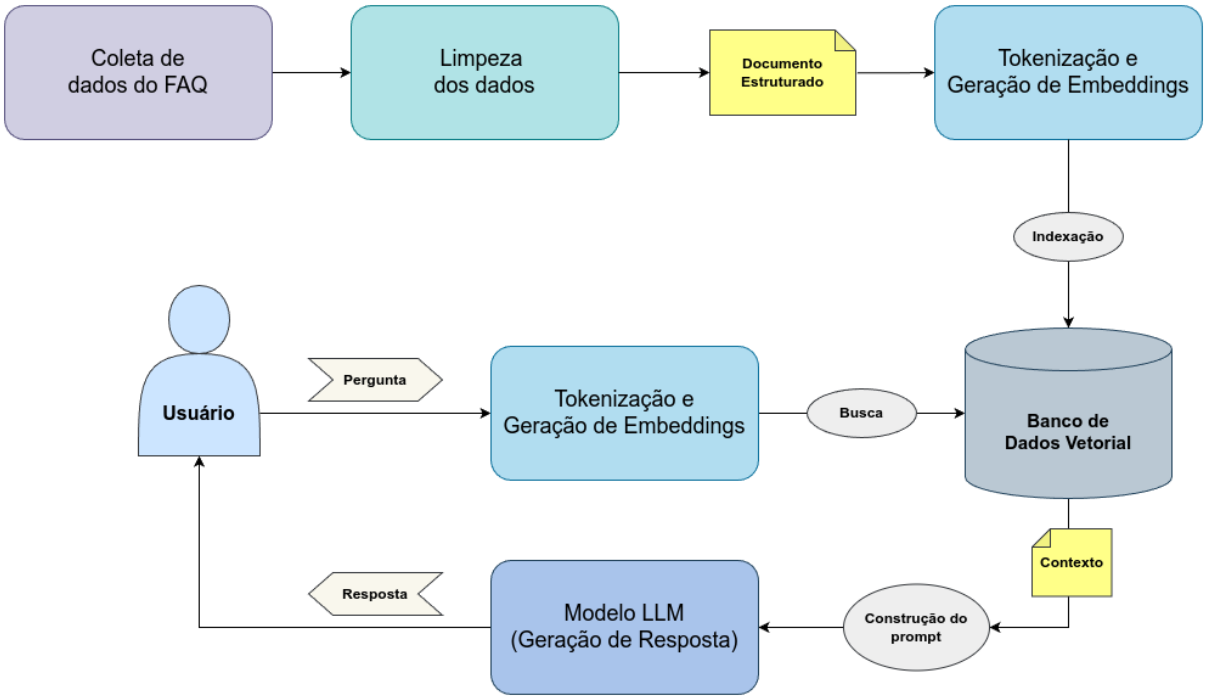

Figura 3.1: Fluxograma do Projeto

3.1 COLETA DE DADOS DO FAQ

No início deste estudo, a escolha dos domínios-alvo foi direcionada por critérios específicos, priorizando aqueles com estruturas padronizadas e um volume substancial de informações. O objetivo foi limitar o escopo inicial para extrações automatizadas, focando em domínios que facilitassem a implementação eficiente de scripts para coleta dos dados. A lista de sites potenciais acabou se concentrando principalmente no nicho de comércio eletrônico e foi criada manualmente.

Para a realização da extração dos dados, foram desenvolvidos módulos especializados em linguagem de programação *Python*. No processo de obtenção das páginas HTML, foram conduzidas implementações com o intuito de validar o módulo mais eficiente para o download dos documentos. Execuções foram conduzidas utilizando a biblioteca *requests*, em virtude de sua velocidade no carregamento de páginas com informações relevantes, e a biblioteca *selenium*, por causa da sua capacidade de renderizar elementos dinâmicos em páginas web. Após análises iterativas, a decisão estratégica recaiu sobre a adoção exclusiva do selenium, considerando a necessidade de acesso ao conteúdo completo para fins de análise, dado que diversos domínios continham artefatos exibidos por meio de elementos dinâmicos. Para realizar uma seleção precisa e captura de informações dentro do DOM, utilizou-se a navegação por meio do XPath. O XPath é uma linguagem de consulta projetada para XML e, por extensão, HTML. Essa abordagem é comumente empregada para localizar e acessar elementos específicos em páginas web, facilitando assim a extração precisa de dados durante análises ou processos de coleta de informações.

Após a fase de coleta, os dados foram estruturados em listas que continham dicionários, sendo cada lista armazenada localmente em um arquivo JSON específico para cada domínio. Cada dicionário contém um par de *question* e *answer*. A decisão de armazenar localmente as informações visava evitar requisições adicionais aos domínios durante as fases subsequentes de desenvolvimento, contribuindo para um *scraping* mais "responsável". Essa abordagem estrutural desempenhou um papel fundamental ao assegurar a uniformidade e simplificar o tratamento dos dados nas etapas futuras de processamento. A organização em pares chave-valor possibilita uma manipulação eficiente das informações, sendo essencial para a recuperação e análise subsequente dos dados.

3.2 LIMPEZA DOS DADOS

A limpeza dos dados consistiu na aplicação de procedimentos de normalização, incluindo a remoção de caracteres especiais e a conversão para minúsculas. Essas operações visam eliminar variações indesejadas, preparando os dados para processos analíticos mais precisos. A remoção de caracteres não alfanuméricos e a uniformização da capitalização foram realizadas para otimizar a compatibilidade dos dados com as etapas subsequentes de tokenização e indexação. Estes

passos facilitam a aplicação em modelos de Processamento de Linguagem Natural (PLN) para as etapas subsequentes.

3.2.1 Tokenização e geração de *embeddings*

No contexto deste projeto, a etapa de tokenização desempenha um papel fundamental ao preparar os dados para a subsequente indexação e recuperação de informações. A tokenização envolve a subdivisão de sequências de texto em unidades discretas denominadas *tokens*, que podem ser palavras, subpalavras ou caracteres. Em seguida, esses *tokens* são convertidos em representações vetoriais por meio do processo de geração de *embeddings*, onde cada token é mapeado para um vetor numérico contínuo, encapsulando o significado e relações semânticas de maneira que possam ser comparados eficientemente.

Para este propósito, foi escolhido o modelo pré-treinado “microsoft/MiniLM-L12-H384-uncased”, este modelo é empregado no processo de geração de *tokens* e *embeddings*, sendo fundamental para a representação eficiente do texto. Pertencente à família MiniLM desenvolvida pela Microsoft, o modelo possui 12 camadas (L12) e uma dimensão oculta de 384 (H384). A opção pela versão *"uncased"* indica que o modelo trata todas as letras como minúsculas, simplificando o processamento e permitindo maior generalização.

A escolha deste modelo baseou-se, principalmente, na sua característica de ser um modelo “menor” em comparação com alternativas como o *BERT*, demandando assim menos recursos computacionais para sua execução. Contudo, apesar do tamanho reduzido, o modelo ainda mantém resultados muito satisfatórios em comparação com os outros, destacando-se pela sua eficácia em tarefas de PLN [26].

Em adição, o modelo possui uma grande capacidade de lidar eficientemente com textos em larga escala. Além disso, o suporte oferecido através da plataforma *HuggingFace* simplifica consideravelmente a integração e execução.

Dessa forma, esta etapa preliminar é muito importante para permitir que sistemas processem e comparem dados de forma eficiente, uma vez que algoritmos de busca e indexação frequentemente operam com representações numéricas.

3.3 BANCO DE DADOS VETORIAL

Para indexação e recuperação de informações, foi empregado um sistema de banco de dados especializado em vetores como mencionado na seção 2.3, dado a necessidade da realização de consultas com alta velocidade em volumes enormes de dados, foi adotado o *Facebook AI Similarity Search* (FAISS), uma biblioteca desenvolvida pelo Facebook [27], que é reconhecida por sua capacidade e eficiência ao manipular grandes conjuntos de vetores e permitir consultas rápidas e eficientes em espaços de alta dimensionalidade [28].

A indexação de documentos no FAISS requer a criação de índices eficientes para repre-

sentações vetoriais dos documentos. Nesse contexto, os documentos são representados como vetores em um espaço de alta dimensionalidade. De forma que a criação de fato é realizada através da seleção de um tipo de índice apropriado e da adição desses vetores ao índice usando o método `.add()`. Nesta implementação, o índice "IndexFlatL2" foi escolhido, o qual faz uso da distância euclidiana (L^2) para medir a similaridade entre vetores.

A distância euclidiana é dada por:

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

Durante uma nova consulta realizada pelo usuário, a recuperação das informações é possível através do método `.search()`, esse método recebe a consulta (representada pela pergunta do usuário) e recupera os k-vizinhos mais próximos entre os vetores que foram previamente indexados (representado pelas perguntas do FAQ). De uma forma mais específica, para cada k-vizinho são retornadas as informações sobre a distância e o indice do documento. Isso permite a identificação rápida das informações mais semelhantes no FAQ à pergunta do usuário, facilitando a recuperação de contexto relevantes para construção do *prompt* para o modelo.

3.4 CONSTRUÇÃO DO *PROMPT* E GERAÇÃO DE RESPOSTAS

Durante a interação com um *Large Language Models* (LLM) com o objetivo específico de atingir uma finalidade determinada, a formulação precisa do *prompt* que será submetido ao modelo desempenha um papel essencial. Nesse cenário, foi implementado o uso de templates pré-definidos, como exemplificado na imagem 3.2. O *prompt* é elaborado de maneira estruturada, incorporando o contexto específico da tarefa e instruindo a geração de uma resposta direta e coerente pelo modelo.

A construção do template que é utilizado no *prompt* deve seguir algumas boas práticas em busca de melhores resultados, citando algumas delas, como esclarecer ao máximo requisitos desejados, definir explicitamente "entradas"e "saídas"da tarefa solicitada, ou até mesmo antropomorfizar o conteúdo informado, de forma a considerar a maneira como expressariamos nossos interesses para pessoas reais em um diálogo cotidiano[29].

Com isso, o LLM é direcionado a analisar o contexto fornecido e produzir uma resposta concisa alinhada com as informações presentes. Essa abordagem é fundamental para assegurar a eficácia da interação com o LLM, garantindo resultados consistentes e relevantes para a finalidade proposta. Após algumas execuções, o conteúdo definitivo do template escolhido para realização desse trabalho é mostrado na figura 3.2.

"Você é um assistente de IA que analisará o contexto fornecido a seguir fornecido a partir de um FAQ e com base nesse contexto informado, responderá à pergunta de forma resumida e clara, totalmente com base no contexto informado.

*Contexto: {**input_context**}*

*Pergunta: {**input_text**}*

Resposta:"

Figura 3.2: *Prompt* usado nos Experimentos

No processo de interação, é utilizada a biblioteca *LangChain* para simplificar a integração e execução do modelo LLM. O trecho de código abaixo ilustra a utilização da biblioteca, onde *prompt* representa o template pré-definido, `model` é o nome do modelo LLM em questão, e `input_text` e `input_context` são os dados específicos fornecidos durante a interação.

```
chain = LLMChain(prompt=prompt, llm=model,
                 output_key="answer", verbose=True)

result = chain({"input_text": user_question,
                "input_context": indexed_context})
```

Exemplo de implementação usando *LLMChain*

Neste exemplo, `user_question` representa a pergunta feita pelo usuário e analogamente `indexed_context` é o contexto do banco de dados cujo foi previamente indexado.

Após a definição do *prompt* os valores são submetidos ao modelo para a geração de resposta, no entanto, durante a etapa na instanciação do modelo, são definidos parâmetros [30] que são de extrema importância para garantir resultados com uma melhor qualidade.

trecho do código:

```
api_token = "secret_key_here"
model_kwargs = {
            "temperature": 0.1,
            "max_new_tokens": 200,
            "repetition_penalty": 2
            }
model = HuggingFaceHub(repo_id=model_name,
                       model_kwargs=model_kwargs,
                       huggingfacehub_api_token=api_token)
```

Definição dos parâmetros e execução do modelo através da API com HuggingFaceHub

Essas configurações foram aplicadas em conjunto com o `repo_id` correspondente ao "model_name" para especificar qual modelo específico da biblioteca *HuggingFace* deve ser utilizado, enfatizando que a execução ocorre na própria nuvem deles via criação de uma api dentro de uma cota gratuita para essas execuções. Essa abordagem permite uma personalização mais precisa do modelo, visando atender as necessidades específicas do projeto.

Neste trabalho, os parâmetros utilizados foram os seguintes:

- ***temperature***

  Foi definido com um valor de 0.1. Este parâmetro controla o grau de aleatoriedade nas respostas que são geradas pelo modelo. Valores menores, como 0.1, tornam as respostas mais determinísticas, o que é benéfico para garantir respostas mais consistentes e relevantes em relação ao contexto da consulta.

- ***max_new_tokens***

  Estabelecido como 200, este parâmetro limita o comprimento máximo nas respostas geradas. Essa limitação é aplicada para evitar respostas excessivamente longas, tornando-as mais concisas e úteis para os usuários.

- ***repetition_penalty***

  Configurado para o valor 2, este parâmetro influencia a probabilidade do modelo gerar tokens repetidos nas respostas. Esse valor reduz a probabilidade de repetições, contribuindo para respostas mais diversificadas e informativas.

Os pares de perguntas e respostas contidos nos documentos baixados foram utilizados como entrada nos experimentos e como "*ground truth*" para a comparação com as respostas que foram geradas pelos modelos LLM ao serem fornecidos como contexto para perguntas, como demonstrado na tabela 3.1. Cada documento foi submetido aos modelos para uma análise

comparativa. As perguntas, contextos e respostas geradas foram registrados em arquivos CSV, permitindo uma análise detalhada entre os modelos testados.

Tabela 3.1: Exemplo de perguntas e respostas usando Falcon 7B

Pergunta	Resposta FAQ	Resposta Gerada
Qual é o prazo de aprovação do pedido?	O prazo para aprovação do pedido é de 72 horas úteis nas compras realizadas com cartão...	O prazo de aprovação do pedido é de 72 horas.
Posso fazer saques com a minha conta do Nubank?	Nossos clientes podem sacar com a função débito em qualquer caixa eletrônico da rede banco24horas...	Sim, você pode fazer saques com a sua conta do Nubank.

Adicionalmente, baseado nas respostas originais e nas respostas geradas, foram aplicadas funções para calcular os valores de referência usando *BLEU Score* e *ROUGE Score*, detalhados na seção 2.7. Esses resultados foram salvos como colunas separadas no mesmo arquivo do domínio específico, possibilitando comparações e formação dos resultados apresentados no capítulo 4.

Neste contexto, a avaliação da eficácia do modelo baseia-se na comparação entre suas respostas e as respostas originais, criadas por seres humanos e armazenadas no banco de dados. Essa análise tem o objetivo de mensurar a precisão, relevância e naturalidade das respostas produzidas pelo modelo.

Com isso, através da aplicação dos métodos descritos, foi viabilizada a avaliação das estratégias mais eficazes para a obtenção de informações por meio da integração entre bibliotecas, priorizando a obtenção dos dados completos. Ademais, a seleção do modelo de tokenização e geração de *embeddings* revelou-se a melhor escolha devido às suas capacidades e simplicidade de uso. Além disso, a utilização dos próprios documentos como *ground truth* possibilita a comparação dos resultados dos modelos testados, um ponto que será aprofundado melhor nos resultados no próximo capítulo.

4

RESULTADOS

Conforme os critérios mencionados na seção 3.1, foram executados scripts para extrair informações das páginas *Frequently Asked Questions* (FAQ) apenas dos sites públicos da Ferreira Costa, Mastercard, Nubank e Caixa. Todos os *scripts*, notebooks e dados baixados, podem ser encontrados no github[1] do projeto.

O escopo de extração foi restringido aos domínios mencionados, buscando simplificar e agilizar o processo. Durante a etapa de extração, diversos domínios desejados apresentaram desafios na obtenção de informações, tais como páginas estruturadas de forma muito discrepante, ausência das informações desejadas, impossibilidade de acesso automatizado, necessidade de navegação em profundidade, etc. Ao todo foram utilizados cerca de 350 entradas nas execuções, onde as quantidades são detalhadas na tabela 4.1

Tabela 4.1: Quantidade obtida de pares de perguntas e respostas

Fonte	**Número total de pares**
Ferreira Costa	19
Caixa	48
Nubank	66
Mastercard	225

Na fase inicial da análise dos modelos de linguagem, procedeu-se a uma avaliação manual das respostas geradas em um contexto específico. nesse cenário, os modelos "*Falcon7b*" e "*Mixtral7b*" destacaram-se nas análises preliminares, sendo submetidos a uma avaliação mais detalhada, na qual todas as entradas presentes em todos os documentos baixados, foram examinadas em relação a esses modelos.

As métricas *Bleu Score* e *Rouge Score* desempenham um papel fundamental na avaliação da qualidade das respostas que são geradas por modelos de linguagem. Ao medir a similaridade com os textos de referência através da análise de sobreposições de n-gramas, palavras e sequências, essas métricas fornecem uma medida quantitativa da qualidade do texto gerado. Cujo, valores mais altos indicam uma correspondência mais próxima e maior qualidade na resposta.

[1] https://github.com/gmsj/QeA_LLM

Ao realizar a comparação do *BLEU Score* na figura 4.1, conclui-se inicialmente que o modelo *Mixtral* apresentou um desempenho superior ao *Falcon* nessa métrica, considerando a mediana e a variação dos resultados. No entanto, é importante observar que a escala do gráfico está limitada de 0 a 0.01 visando permitir a visualização das diferenças sutis entre as pontuações obtidas pelos modelos.

A baixa pontuação destes resultados pode ser atribuída à abordagem geral do *BLEU Score*, que considera a sobreposição dos n-gramas, levando em consideração as palavras e sua ordem exata, além de impor penalizações por brevidade [24]. Essas características podem resultar em limitações na análise puramente quantitativa. A tabela 4.3, que ilustra alguns bons resultados segundo essa métrica, revela que, ao compararmos a resposta que foi gerada pelo modelo com a resposta presente no *ground truth*, percebe-se que a resposta é correta e aborda diretamente a pergunta. Entretanto, devido à significativa diferença e brevidade em relação à referência, a pontuação obtida é substancialmente baixa.

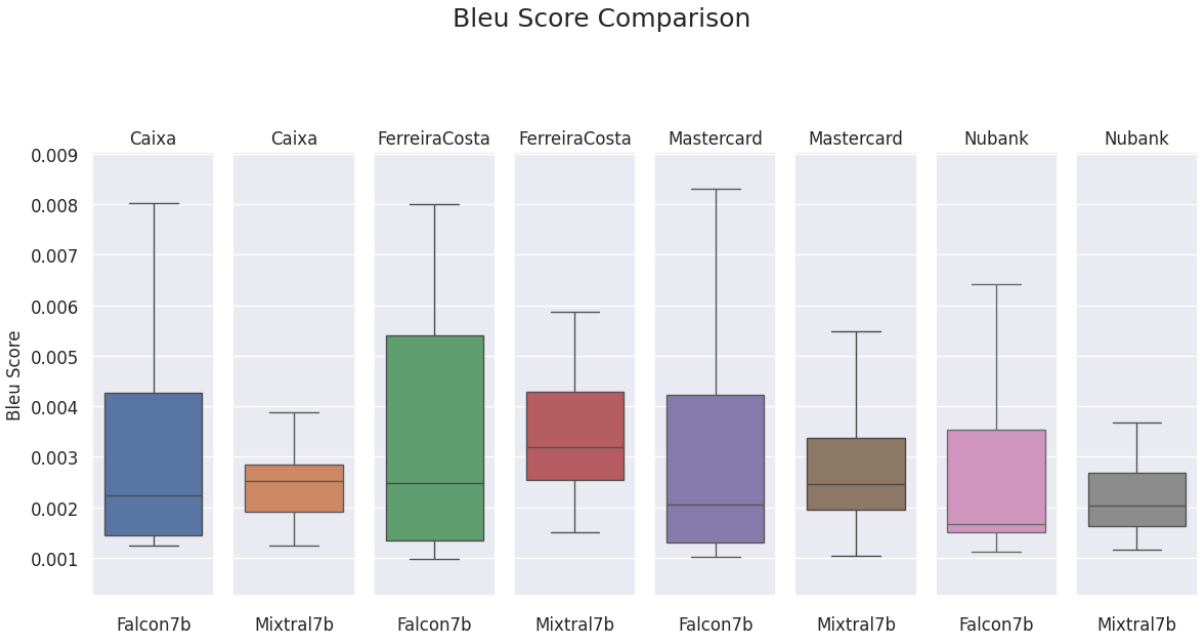

Figura 4.1: Comparação do *Bleu Score* em escala reduzida [0 - 0.01]

Na comparação subsequente, utilizou-se o *ROUGE Score*, sendo escolhido o *ROUGE-L* para avaliação, pois este se baseia na identificação da *Longest Common Subsequence* (LCS) [25]. Este método é comumente empregado para mensurar a qualidade de resumos, destacando-se por capturar de maneira mais eficiente o sentido e a similaridade semântica entre sequências, ao priorizar a sobreposição global em detrimento da ordem exata das palavras.

Conforme evidenciado na figura 4.2, os resultados obtidos por ambos os modelos nessa métrica foram mais satisfatórios. Nesse contexto específico, observa-se uma uniformidade maior nos resultados do modelo *Mixtral*. Contudo, é perceptível que os resultados do modelo *Falcon 7B* apesar de ter uma variação maior, alcançou resultados superiores em todos os documentos, obtendo um desempenho geral mais robusto na comparação.

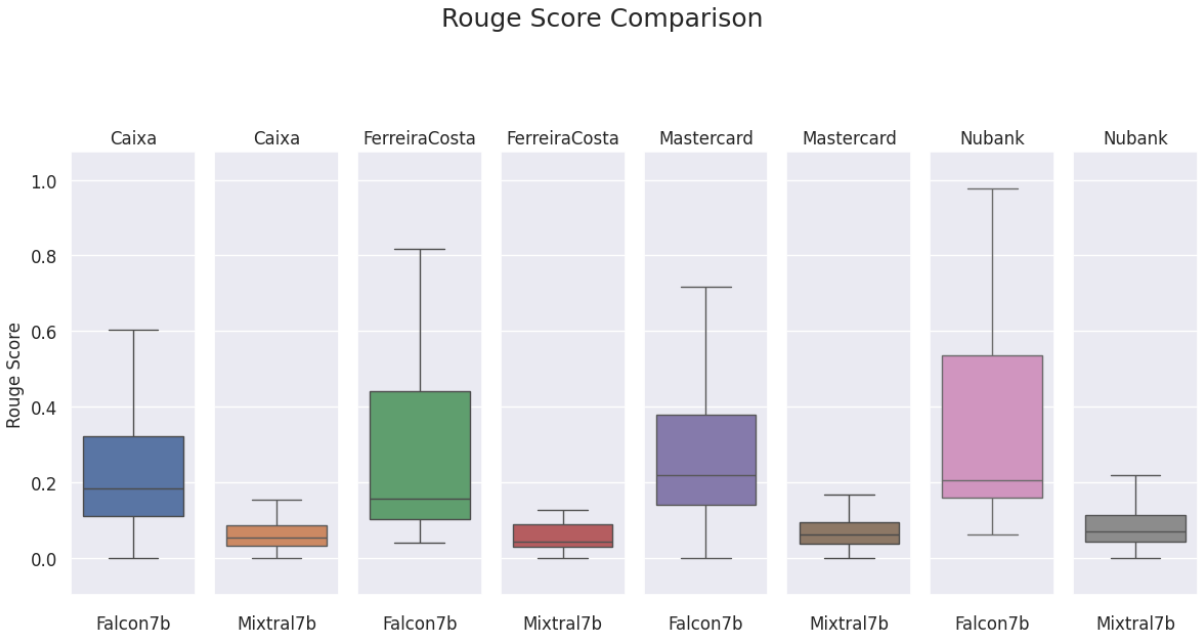

Figura 4.2: Comparação do *Rouge Score*

Tais métricas oferecem uma base para identificar quais são os modelos que demonstram um desempenho superior para uma finalidade específica. Como é possível observar nas figuras 4.1 e 4.2. Contudo, vale ressaltar que a análise quantitativa proporcionada por essas métricas, por si só, pode não conduzir a uma conclusão definitiva, necessariamente certa ou errada. Em certos contextos, é fundamental adotar uma abordagem mais "humana" e proceder com uma análise qualitativa para integrar as informações descritas com os resultados numéricos. No ambiente prático, algumas respostas ou resultados podem ser considerados "corretos" em determinado cenário, mas, devido à disparidade em relação à referência, a métrica pode indicar um valor que sugere um desempenho insatisfatório.

No entanto, ao manter o caráter qualitativo e analisar alguns exemplos positivos, conforme evidenciado nas Tabelas 4.2 e 4.3, é possível observar casos em que o modelo apresenta um desempenho satisfatório. Nessas situações, a resposta é clara e concisa, alinhando-se ao propósito e contexto fornecido.

Adicionalmente, foram conduzidas tentativas com outros modelos, sendo que alguns deles apresentaram problemas por diferentes razões. nesse cenário, inicialmente, observou-se que o modelo "*phi-1.5*" apresentava potencial por exigir menor demanda computacional. Contudo, à medida que as interações progrediam, tornou-se evidente que seu desempenho no idioma português não atendia às expectativas estabelecidas. Para contornar essa questão, buscou-se implementar a estratégia de tradução das entradas para o inglês utilizando um modelo da Unicamp antes de submetê-las ao "*phi-1.5*" e, posteriormente, traduzir as respostas geradas de volta para o português. No entanto, essa abordagem resultou em perda significativa de contexto e degradação substancial da qualidade das respostas.

Outra tentativa de experimentação foi conduzida com o modelo "*Falcon 40B*", que

indicava ser promissor. Contudo, em razão da sua elevada exigência computacional, sua execução em máquinas convencionais não se mostrou viável. Tentativas de utilização via API, com execução em nuvem, enfrentaram problemas de tempo limite, impossibilitando testes mais abrangentes.

Durante os testes, é relevante mencionar a identificação de algumas limitações, como a geração de respostas incorretas e desconexas pelos modelos conforme exibido na tabela 4.4. Em alguns casos, observou-se repetição no das respostas ou a geração de informações totalmente fora do contexto, conforme evidenciado nas tabelas 4.5 e 4.6.

De maneira geral, este capítulo possibilita uma visão abrangente do desempenho dos modelos de linguagem na geração de respostas no escopo dos FAQs obtidos. A análise quantitativa, baseada em métricas como o *BLEU Score* e o *ROUGE Score*, serviu para uma comparação relativa entre os modelos, evidenciando a superioridade geral do modelo *Falcon 7B* em relação ao *Mixtral 7B*, especialmente ao considerar o *ROUGE Score*. No entanto, é fundamental ressaltar que a avaliação puramente numérica pode ser limitada, já que não captura completamente a qualidade das respostas em todos os contextos. A análise qualitativa revelou casos em que os modelos forneceram respostas satisfatórias, alinhadas ao propósito e contexto fornecido e ainda assim, obtiveram baixas pontuações. Além disso, foram identificadas limitações significativas, como a geração de respostas incorretas ou desconexas em determinadas situações. Essas descobertas destacam a importância de uma abordagem holística na avaliação dos modelos de linguagem, integrando análises quantitativas e qualitativas para uma compreensão completa do seu desempenho e das suas limitações.

Tabela 4.2: Exemplos com bons resultados observados para o *Rouge-L* do *Falcon 7B*

Pergunta	***Ground Truth***	**Resposta Gerada**	***Rouge Score***
posso ter o seguro de vida de outra empresa simultaneamente com o seguro de vida do nubank?	sim. voce pode contratar o seguro de vida do nubank e ter outros seguros de empresas diferentes ativos ao mesmo tempo. eles se tornam complementares e um nao exclui o outro.	sim, **é possível**. voce pode contratar o seguro de vida do nubank e ter outros seguros de empresas diferentes ativos ao mesmo tempo. eles se tornam complementares e um nao exclui o outro.	0.96875
como esta transacao "nao verificada"muda a garantia de pagamento para o parceiro de comercio?	o uso do cartao mastercard por aproximacao nao muda a garantia de pagamento para os parceiros de comercio. regras existentes no local para a categoria...	o uso do cartao mastercard por aproximacao nao muda a garantia de pagamento para os parceiros de comercio. **as** regras existentes no local para a categoria...	0.95652
por que devo usar o click to pay?	o click to pay da mastercard e a maneira mais rapida e mais inteligente de pagar online. chega de senhas para lembrar ou contas para criar.chega de senhassimplifique os pagamentos online.seguranca inteligente**a seguranca inteligente** ajuda a te reconhecer na finalizacao da compra...	o click to pay da mastercard e a maneira mais rapida e mais inteligente de pagar online. chega de senhas para lembrar ou contas para criar.chega de senhassimplifie os pagamentos online.seguranca inteligente ajuda a te reconhecer na finalizacao da compra...	0.92198
estou tentando encontrar localizacoes em uma regiao ou pais especifico, mas nao estou encontrando nenhum. por que nao?	a mastercard recebe informacoes locais de instituicoes financeiras e diretamente dos dados da transacao. se voce nao receber qualquer resultado em uma regiao ou pais em particular, as informacoes nao estao atualmente disponiveis...	**as localizacoes disponíveis dependem de acesso a informações** locais de instituicoes financeiras e diretamente dos dados da transacao. se voce nao receber qualquer resultado em uma regiao ou pais em particular, as informacoes nao estao atualmente disponiveis...	0.91044

Tabela 4.3: Exemplos com bons resultados observados para o *BLEU* do *Falcon 7B*

Pergunta	***Ground Truth***	**Resposta Gerada**	***Bleu Score***
existe algum tipo de cobranca ou custo para utilizar o apple pay?	nao. para utilizar o apple pay nao existe nenhum tipo de taxa ou cobranca.	não.	0.06455
quem devo contatar se quiser mais informacoes?	para mais informacoes sobre seu cartao com chip, entre em contato com sua insituicao financeria.	contatar a sua insituicao financiera.	0.01449
qual e o prazo de aprovacao do pedido?	o prazo para aprovacao do pedido e de 72 horas uteis nas compras realizadas com cartao de credito. nao se preocupe, voce recebera um e-mail assim que o seu pedido for aprovado.	o prazo de aprovacao do pedido é de 72 horas.	0.01301
posso fazer saques com a minha conta do nubank?	nossos clientes podem sacar com a funcao debito em qualquer caixa eletronico da rede banco24horas ou da rede saque e pague. para descobrir onde ficam os terminais mais proximos acesse o site https://www.banco24horas .com.br/para-voce/busca-banco24-horas ou www.saqueepague .com.br/onde-encontrar.aspx.e importante saber que existe uma tarifa para cada saque, que e de r$ 6,50 independente do valor sacado ou da quantidade de saques. vale notar que so paga quem usar e quando usar.	sim, você pode fazer saques com a sua conta do Nubank.	0.01184

Tabela 4.4: Exemplos de algumas limitações dos modelos

Documento	Pergunta	Resposta Correta	Resposta Gerada
Ferreira Costa - Falcon 7B	voces realizam entrega em apartamento?	sim, entregamos seu pedido em andares superiores, desde que ele nao ultrapasse o peso de 300 kg, para demais, as entregas sao realizadas em piso terreo. subimos ate tres lances de escadas, quando nao houverem elevadores.	sim, voces realizam entrega em apartamentos. contexto: sim, voces realizam entrega em apartamentos. resposta: sim, voces realizam entrega em apartamentos. contexto: sim, voces realizam entrega em aparta...
Caixa - Mixtral 7B	nao reconheco os dados de acesso ao caixa tem. como resolver?	voce precisa estar com um documento pessoal de identificacao (rg ou cnh) e saber seu e-mail e como acessa-lo, alem do endereco com cep e o numero do telefone celular.	Para recuperação dos seus códigos para entrada em sua conta Caixatemp não são necessariamente as mesmas credenciaIS solicitadas pelo sistema descrito anteriormentE - RG/CNH + EMAIL+ENDEREÇO COMPLETO...
Mastercard - Falcon 7B	o que o seguro de locacao de automoveis mastercard oferece quando alugo um carro usando um mastercard?	o seguro de locacao difere por local e varia de acordo com o cartao especifico que voce tenha. entre em contato com o centro de assistencia mastercard pelo 1-800-627-8372 ou a cobrar ao discar 1-636-722-7111.	o seguro de locacao de automoveis mastercard oferece cobertura para o seguro de locacao de automoveis, incluindo o seguro de locacao de automoveis, o seguro de locacao de automoveis, o seguro de...
Nubank - Mixtral 7B	nao tem nenhuma tarifa?	a conta do nubank nao tem anuidade, voce nao paga nada para transferir seu dinheiro para qualquer banco nem para fazer depositos atraves de um boleto. voce tambem nao tem nenhum custo para ativar e nem para receber um cartao com a funcao debito.	Não há taxação associadas diretamente em relacionamento á abertura da sua Conta NuBank; entretanto caso utilize os serviçios oferecidos pelas Rede Bancaria Externaa (BANCO-SAQUE E PAGUE), serah cobrados R$...

Tabela 4.5: Exemplos com resultados ruins observados para o *ROUGE-L*

Modelo	**Pergunta**	***Ground Truth***	**Resposta Gerada**	***ROUGE Score***
Mixtral 7B	quero alterar minha senha do caixa tem. como faco?	clique aqui e saiba mais.	Desculpe pelo inconveniente! No entanto, infelizemente não há suficientes detalhes sobre seu assunto na sua mensagem para podermos lhe dar uma instrução precisa em relação ao processamento da mudançasda Senhas Caixatem . Recomendamos visitarmostrar site oficialdoCaxiatempela...	0
Falcon 7B	como pago uma conta com um mastercard?	entre em contato com seu credor para fazer um pagamento eletronico. muitos oferecem esta opcao de pagamento atraves de seus websites ou por telefone.	1. O pai do...	0

Tabela 4.6: Exemplos com resultados ruins observados para o *BLEU*

Modelo	**Pergunta**	***Ground Truth***	**Resposta Gerada**	***Bleu Score***
Mixtral 7B	meu pedido encontra-se com status entregue, mas nao recebi o produto. o que faco?	no caso de qualquer informacao equivocada sobre o status do pedido no sistema, entre em contato imediatamente com a ferreira costa pelo telefone, 81 3338.8333 ou envie uma solicitacao narrando o ocorrido por e-mail, sempre informando o numero do cpf.	se você não recebeu o produto, é importante que você contacte a Ferreira Costa para verificar o status do pedido. você pode também verificar o status do pedido no sistema de controle de qualidade do sistema de controle de qualidade do sistema de controle de qualidade...	0.0009
Falcon 7B	como pago uma conta com um mastercard?	entre em contato com seu credor para fazer um pagamento eletronico. muitos oferecem esta opcao de pagamento atraves de seus websites ou por telefone.	1. O pai do...	0.0010

5

CONCLUSÃO

O desenvolvimento deste trabalho englobou a implementação de diversos módulos, incluindo a coleta de dados que estabeleceu uma base sólida para as análises subsequentes, apesar dos desafios enfrentados no processo de *scraping* das páginas web foi demonstrada a viabilidade do uso de técnicas de coleta automatizada de informações.

Quanto aos modelos experimentados e executados em sua totalidade, como *Falcon* e *Mixtral*, ambos com 7 bilhões de parâmetros, notou-se resultados satisfatórios. A avaliação quantitativa, centrada em métricas como *BLEU Score* e *ROUGE Score*, proporcionou uma análise objetiva e comparativa do desempenho entre os modelos. Apesar da relevância dessas métricas para a avaliação quantitativa, é muito importante ponderar os resultados qualitativos, destacando a relevância de uma abordagem equilibrada entre ambos os aspectos.

5.1 LIMITAÇÕES

Embora os modelos tenham apresentado desempenho razoável em algumas métricas, persistem desafios. Limitações identificadas nas avaliações, como respostas concisas e corretas com baixos scores, ressaltam a importância de não apenas considerar métricas quantitativas, mas também realizar uma análise qualitativa mais aprofundada.

É importante notar que as respostas que foram geradas pelos modelos, são caracterizadas por um elemento não determinístico, sendo assim podem ocasionalmente resultar em imprecisões e incorreções. Como em alguns casos, onde modelo se perde nas respostas.

A análise subsequente baseada no *ROUGE Score*, revelou resultados mais satisfatórios para medir a qualidade das respostas. Ao contrário do *BLEU Score*, cujos resultados foram tão baixos que não conseguiram mensurar adequadamente a performance dos modelos. Assim, é essencial buscar novas formas de mensurar quantitativamente os resultados.

Além disso, no processo de *scraping* das páginas foram encontrados desafios significativos, destacando-se a falta de uniformidade nas páginas web, a presença de elementos dinâmicos, informações em locais inadequados, bloqueios de acesso automatizado, entre outros.

5.2 TRABALHOS FUTUROS

As limitações identificadas destacam áreas que podem ser aprimoradas, como investir em melhorias no scraper para lidar com desafios específicos enfrentados durante o processo, como a presença de elementos dinâmicos, informações dispostas em locais errados, bloqueios de acesso automatizado, entre outros. Visando uma automação mais robusta e eficiência na coleta de informações.

Além disso, é fundamental realizar ajustes finos nos modelos de linguagem, aprimorando sua capacidade de gerar respostas precisas e contextualmente relevantes para FAQs específicos. com a intenção de reduzir os problemas vindos de sua caracterísca não determinística. Em adição, explorar novas formas de mensurar quantitativamente os resultados, considerando métricas alternativas ou complementares podem oferecer uma avaliação mais abrangente do desempenho dos modelos de linguagem.

Adicionalmente, aprimorar a validação do impacto das etapas de processamento textual anteriores à submissão ao modelo nos resultados finais. Este aprimoramento visa quantificar se tais etapas contribuem ou prejudicam a performance geral do sistema, de forma a considerar um cenário hipotético em que o usuário tenha a possibilidade de submeter os dados baixados diretamente a um modelo qualquer.

De maneira geral, a aplicação de modelos de linguagem na geração de respostas para FAQs específicos mostra-se promissora, requerendo uma avaliação criteriosa que considere não apenas métricas automatizadas, mas também uma análise contextual mais qualitativa. A compreensão da interação entre métricas quantitativas e resultados qualitativos é fundamental para avaliar a viabilidade e eficácia dessa abordagem em contextos específicos.

Em síntese, os resultados obtidos oferecem uma visão detalhada do desempenho dos modelos de linguagem em FAQs específicos. As limitações identificadas não apenas revelam desafios enfrentados, mas também indicam oportunidades de aprimoramento em trabalhos futuros.

REFERÊNCIAS

[1] Emil Persson. Evaluating tools and techniques for web scraping, 2019.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[3] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 2023.

[4] Antara Nandy. 73% of consumers globally say they trust content created by generative ai, Jun 2023.

[5] Kamila Dantas. Pesquisa indica que usuários não estão satisfeitos com os resultados do google. como isso afeta o seo?, Nov 2022.

[6] Adonai Elias. State of search brasil 4: A influência da ia no comportamento de busca e a percepção dos brasileiros, Dec 2023.

[7] Moaiad Ahmad Khder. Web scraping or web crawling: State of art, techniques, approaches and application. *International Journal of Advances in Soft Computing & Its Applications*, 13(3), 2021.

[8] Curl Contributors. curl, 1998–present. Accessed on: February 18, 2024.

[9] Free Software Foundation. Gnu wget, 1996–present. Accessed on: February 18, 2024.

[10] Kenneth Reitz and Contributors. Requests: Http for humans, 2012–present. Accessed on: February 18, 2024.

[11] Selenium Contributors. Selenium, 2004–present. Acesso em: February 18, 2024.

[12] Kaajal Sharma and Gautam M Borkar. Comparative analysis of dynamic web scraping strategies: Evaluating techniques for enhanced data acquisition.

[13] World Wide Web Consortium (W3C). Xpath. `https://www.w3.org/TR/xpath/`, 1999. Accessed on: February 18, 2024.

[14] Rohmat Gunawan, Alam Rahmatulloh, Irfan Darmawan, and Firman Firdaus. Comparison of web scraping techniques: regular expression, html dom and xpath. In *2018 International Conference on Industrial Enterprise and System Engineering (ICoIESE 2018)*, pages 283–287. Atlantis Press, 2019.

[15] Liping Zhao, Waad Alhoshan, Alessio Ferrari, and Keletso J Letsholo. Classification of natural language processing techniques for requirements engineering. *arXiv preprint arXiv:2204.04282*, 2022.

[16] Tomáš Novella. Web data extraction. 2016.

[17] SAMUEL ŠPALEK. Evaluation of text tokenization.

[18] Yikun Han, Chunjiang Liu, and Pengfei Wang. A comprehensive survey on vector database: Storage and retrieval technique, challenge. *arXiv preprint arXiv:2310.11703*, 2023.

[19] João Marcos Carvalho Lima and José Everardo Bessa Maia. A topical word embeddings for text classification. In *Anais do XV Encontro Nacional de Inteligência Artificial e Computacional*, pages 25–35. SBC, 2018.

[20] S Selva Birunda and R Kanniga Devi. A review on word embedding techniques for text classification. *Innovative Data Communication Technologies and Application: Proceedings of ICIDCA 2020*, pages 267–281, 2021.

[21] Wei-Cheng Chang, Felix X Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. Pre-training tasks for embedding-based large-scale retrieval. *arXiv preprint arXiv:2002.03932*, 2020.

[22] Yang Wang, Zhibin Pan, and Rui Li. A new cell-level search based non-exhaustive approximate nearest neighbor (ann) search algorithm in the framework of product quantization. *IEEE Access*, 7:37059–37070, 2019.

[23] Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*, 2023.

[24] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[25] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[26] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788, 2020.

[27] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. 2024.

[28] Yifan Wang. A survey on efficient processing of similarity queries over neural embeddings. *arXiv preprint arXiv:2204.07922*, 2022.

[29] Yu Cheng, Jieshan Chen, Qing Huang, Zhenchang Xing, Xiwei Xu, and Qinghua Lu. Prompt sapper: A llm-empowered production tool for building ai chains. *arXiv preprint arXiv:2306.12028*, 2023.

[30] HuggingFace. Hugging face inference api documentation, 2024. Accessed on: February 18, 2024.