UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

Retrieval-Augmented Generative AI Chatbot

Rita Maria Oliveira Rodrigues

Mestrado em Ciência de Dados

Dissertação orientada por:
Professor Doutor António Branco

2024

Agradecimentos

Gostaria de expressar a minha profunda gratidão a todos aqueles que contribuíram para a realização desta dissertação.

Em primeiro lugar, gostaria de agradecer aos meus pais e ao meu irmão pelo apoio incondicional, incentivo e amor ao longo de todo este percurso. Sem o vosso suporte constante e carinho, todo o meu percurso académico não teria sido possível.

Agradeço ao meu orientador, Professor Doutor António Branco, cuja orientação foi fundamental para o desenvolvimento deste trabalho. Agradeço também a todos os membros do Grupo NLX.

E ainda um reconhecimento especial à Fundação para a Ciência e Tecnologia (FCT) pelo financiamento proporcionado através do projeto Accelerat.ai.

Abstract

This work consists on the development and evaluation of a chatbot that integrates retrieval-augmented generation (RAG) to tackle the issue of hallucination in large language models (LLMs). It begins with an introduction that outlines the evolution of chatbots from simple rule-based systems to advanced models using transformers.

Then a detailed history of chatbots, their various categories, and their advantages and disadvantages is provided. It discusses the hallucination problem and introduces the RAG approach, which combines retrieval-based and generative techniques to improve the accuracy and reliability of chatbot responses. The related work section reviews existing literature on methods to mitigate hallucination in LLMs and examines techniques that tackle each stage within the RAG process.

Next, a description of the datasets used is given, including the MS MARCO question-answering and passage retrieval datasets, and the ”Guia Técnico do Alojamento Local.” The preprocessing steps and dataset characteristics are thoroughly explained. The methods chapter outlines the six-phase methodology: data preprocessing, embedding model, vector database, conversational chain, response generation, and interface and deployment. Each phase is elaborated to illustrate the process of constructing the RAG chatbot.

The results of the chatbot's performance are presented using various metrics for retrieval and generation. It presents findings from experiments conducted with the local accommodation dataset and the MS MARCO dataset, demonstrating the chatbot's enhanced performance due to the RAG approach. Finally, the conclusion summarizes the thesis' contributions. It also suggests avenues for future research.

Keywords: Chatbot, Retrieval-Augmented Generation (RAG), Langchain, Hallucination, Large Language Models (LLMs)

Resumo

O principal objetivo desta dissertação é o desenvolvimento de um chatbot que integra a geração aumentada por recuperação (RAG) para diminuir o problema das alucinações nos grandes modelos de linguagem (LLMs).

Ao longo dos anos, temos assistido a uma evolução significativa nos chatbots, que passaram de sistemas simples baseados em regras a modelos avançados que utilizam transformers. Esta evolução foi impulsionada pelos progressos significativos em grandes modelos de linguagem (LLMs), como, mais recentemente, o GPT-4, que tem a capacidade de gerar respostas aparentemente coerentes em conversas.

O problema surge quando o utilizador faz perguntas sobre dados mais recentes ou sobre tópicos que o modelo não foi treinado, ou até mesmo dados que o modelo nunca viu. Nestes casos, o modelo começa a produzir alucinações, ou seja, a geração de conteúdo que não está presente ou implicito nos dados usados para treinar o modelo. Portanto, os modelos geram respostas que, apesar de parecerem plausíveis, são factualmente incorretas ou completamente inventadas. Este problema é particularmente preocupante em domínios onde não podem haver erros, como saúde e lei.

Para abordar este problema, esta dissertação propõe a integração de técnicas de recuperação aumentada por geração, que combinam abordagens baseadas em recuperação de informação com geração de linguagem natural. O objetivo é melhorar a precisão das respostas fornecidas pelos chatbots, garantindo que estas são baseadas em informações factualmente corretas retiradas de fontes fidedignas. A hipótese é a de que a abordagem RAG pode efetivamente mitigar o problema das alucinações, melhorando a fiabilidade e a utilidade dos chatbots em vários domínios.

Os primeiros chatbots, como a ELIZA, desenvolvida na década de 60, funcionavam com base em regras predefinidas. Estes sistemas eram bastante limitados, pois apenas podiam responder a um conjunto restrito de perguntas, o que os tornava incapazes de lidar com linguagem natural de forma flexível e dinâmica.

Com o avanço da tecnologia e das redes neuronais, os chatbots começaram a utilizar modelos baseados em aprendizagem profunda. Estes modelos podem analisar grandes volumes de dados e aprender padrões complexos de linguagem, melhorando significativamente a interação com os utilizadores.

A introdução de grandes modelos de linguagem, como o GPT-3 e GPT-4, revolucionou o campo dos chatbots. Estes modelos, baseados na arquitetura de transformers, são capazes de gerar respostas aparentemente coerentes. No entanto, apesar das suas capacidades avançadas, estes modelos ainda sofrem de problemas de alucinações, especialmente quando confrontados com perguntas fora do seu domínio de treino.

Depois da história da evolução dos chatbots, é feita uma discussão sobre as vantagens e desvantagens dos mesmos. Entre as vantagens, destaca-se a capacidade de fornecer assistência 24/7, melhorar

a eficiência do atendimento ao cliente e reduzir custos operacionais. Contudo, também existem desvantagens, como a dificuldade em lidar com pedidos complexos e a possibilidade de gerar respostas imprecisas ou irrelevantes. A falta de compreensão profunda da linguagem e a dependência de dados de treino limitados são fatores que contribuem para estas desvantagens.

É feita também uma revisão da literatura sobre técnicas para mitigar alucinações em LLMs e sobre técnicas existentes em cada uma das fases da geração aumentada por recuperação: recuperação, aumento dos dados e geração de respostas.

A fase de recuperação envolve a procura de documentos relevantes a partir de uma grande coleção de dados. Esta fase utiliza algoritmos avançados de recuperação de informação para garantir que as informações mais pertinentes são identificadas.

Na fase de aumento dos dados, as informações recuperadas são processadas e estruturadas para serem utilizadas na geração de respostas. Esta fase pode incluir a seleção de passagens específicas e a combinação de informações de múltiplas fontes.

A fase final envolve a geração de respostas utilizando as informações recuperadas e aumentadas. O modelo de geração de linguagem natural é responsável por produzir respostas coerentes e factualmente corretas.

De seguida, os dados utilizados neste trabalho são apresentados. As principais fontes de dados são o MS MARCO e o "Guia Técnico do Alojamento Local".

O 'MS MARCO Question Answering' é utilizado para tarefas de perguntas e respostas, e consiste em pares de perguntas feitas por utilizadores do Bing e respostas cuja geração foi baseada em passagens retiradas da web.

O principal objetivo do 'MS MARCO Passage Retrieval' é recuperar as passagens relevantes a partir de uma grande coleção de documentos, ou seja, encontrar o conteúdo mais pertinente que responde às perguntas dos utilizadores.

O 'Guia Técnico do Alojamento Local' é um documento retirado da web, em português de Portugal, que contém informações detalhadas sobre regulamentações e práticas recomendadas para o alojamento local em Portugal. Com base neste documento, construí um dataset de perguntas e respostas. É feita uma descrição detalhada sobre este processo, assim como uma análise dos datasets e o seu pré-processamento.

A metodologia utilizada para construir o chatbot é detalhada. O processo começa com o pré-processamento de dados, onde os dados são limpos e estruturados para serem utilizados pelo modelo de embeddings. Este modelo transforma os dados textuais em representações vetoriais, que são armazenadas numa base de dados vetorial eficiente. Na fase seguinte, a cadeia conversacional conecta os vetores com a geração de respostas, garantindo que as respostas do chatbot são baseadas em informações recuperadas.

O modelo escolhido foi o Mistral-7B-Instruct-v0.2, um modelo com a licença Apache 2.0, com 7B de parâmetros, o que evita a necessidade de adquirir hardware de custo financeiro elevado. Tem mecanismos de atenção inovadores, como o Grouped-Query Attention e Sliding Window Attention, ambos descritos no capítulo 5.

A interface do chatbot foi desenhada para ser intuitiva e eficiente, garantindo uma interação fluida e agradável para os utilizadores. Este design centrado no utilizador é fundamental para maximizar a usabilidade do sistema, independentemente do nível de habilidade técnica dos utilizadores.

O chatbot apresenta diversas funcionalidades, entre elas permitir aos utilizadores interagirem com o chatbot em tempo real, recebendo respostas rápidas e precisas, armazenar as conversas passadas e recolher feedback dos utilizadores sobre a precisão e relevância das respostas.

Os resultados mostram que a integração de recuperação de informação ajuda a garantir que as respostas sejam baseadas em fontes fiáveis e contextualmente relevantes. A capacidade de fornecer informações fiáveis e baseadas em documentos pode melhorar significativamente a experiência do utilizador e a confiança no sistema, o que é essencial em domínios como a legislação.

Para trabalho futuro, considera-se a expansão dos datasets utilizados e a adaptação do sistema a outros domínios de aplicação, por exemplo, educação e recursos humanos. Além disso, a continuação do desenvolvimento de técnicas avançadas de recuperação e geração de informação pode contribuir para a criação de chatbots ainda mais precisos e confiáveis. A integração de feedback de utilizadores e a melhoria contínua do sistema com base em dados reais também são áreas promissoras para investigação futura.

Em conclusão, o desenvolvimento de um chatbot com geração aumentada por recuperação demonstrou ser uma abordagem eficaz para mitigar o problema das alucinações em grandes modelos de linguagem. Através da combinação de técnicas avançadas de recuperação de informação e geração de linguagem natural, foi possível criar um sistema que fornece respostas precisas e relevantes, melhorando a experiência do utilizador e a confiança no sistema. Continuar a pesquisa e o desenvolvimento nesta área promissora pode trazer avanços significativos não apenas para a chatbots, mas também para a forma como interagimos com sistemas de inteligência artificial em geral.

Palavras-chave: Robô de conversa, Geração Aumentada por Recuperação (RAG), Alucinação, Grandes Modelos de Linguagem (LLMs)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context and Motivation

Chatbots, computer programs designed to engage in conversations, have significantly evolved over time, transitioning from simplistic rule-based systems to sophisticated natural language processing models employing the neural transformers architecture [Caldarini et al., 2022]. These intelligent conversational agents operate seamlessly, replicating the natural flow of human interaction. Offering assistance round the clock through text or voice interactions, chatbots demonstrate remarkable efficiency by attending to multiple users simultaneously, thereby alleviating the burden on human customer support services and subsequently reducing costs [Følstad and Skjuve, 2019]. The integration of chatbots into various facets of daily life has transformed the way individuals interact with technology. From aiding in customer service inquiries to providing personalized recommendations, chatbots have become indispensable tools in enhancing user experience and streamlining processes across diverse industries. Their ability to adapt to different communication styles, coupled with advancements in natural language understanding, has propelled them to the forefront of conversational AI.

One of the driving forces behind the evolution of chatbots is the advent of Large Language Models (LLMs). GPT-4 [OpenAI, 2023], Llama [Touvron et al., 2023], Falcon[Almazrouei et al., 2023], characterized by their vast knowledge base and sophisticated language processing capabilities, have revolutionized the landscape of chatbots.

However, alongside their tremendous potential, LLMs also present significant challenges. Among these is the phenomenon of hallucination, wherein the model generates responses that are not grounded in factual information [Ji et al., 2023]. This issue poses a considerable obstacle, particularly in domains where accuracy and reliability are essential, such as health care, finance, and legal services. As researchers strive to harness the full potential of LLMs in real-world applications, addressing the issue of hallucination emerges as a critical area of focus. By understanding the underlying factors contributing to this phenomenon and developing techniques to mitigate its impact, it is possible to unlock new possibilities for leveraging LLMs in areas where they can truly make a difference.

1.2 Objectives

In this master thesis, my primary objective is to address the challenge of hallucination in Large Language Models (LLMs) and explore practical strategies to mitigate its effects. By conducting a thorough analysis of existing research, empirical studies, and real-world applications, I aim to contribute to ongoing efforts to enhance the reliability and trustworthiness of LLM-powered chatbots. Through practical experimentation and evaluation, I seek to uncover insights that can inform the development of more robust conversational AI systems, ultimately facilitating their adoption across various domains.

To achieve this goal, I propose the development of a chatbot capable of accurately responding to questions based on a provided document. While similar approaches have been explored previously, my work differs in key aspects. Firstly, I prioritize the use of open-source technologies to ensure data privacy and security, contrasting with platforms reliant on proprietary solutions (like the ones offered by OpenAI). Secondly, my chatbot will be equipped with multilingual capabilities, enabling interactions in Portuguese, a language currently underserved in chatbot development.

This chatbot offers a practical and adaptable solution for document interaction, comprising two core components: information retrieval and answer generation. Information retrieval involves the extraction of relevant content from the provided document, while answer generation focuses on crafting accurate responses to user queries based on the retrieved information. This approach aims to ensure that users receive reliable answers in a timely manner.

To evaluate the effectiveness of my proposed methodology, I will conduct a series of experiments using real-world datasets. For the Portuguese language component, I will utilize a comprehensive PDF guide detailing local accommodation options in Portugal. For the English language component, I will use a diverse dataset encompassing various topics.

Through this research, I aim to provide a practical framework for addressing the challenge of hallucination in LLMs, offering one possible solution to more reliable and trustworthy chatbot systems.

1.3 Contributions

This work generated the following contributions:

1. Development of a Portuguese Dataset: Created a dataset in Portuguese, providing resources for further research and development in the field of Natural Language Processing.

2. End-to-End RAG Implementation Chatbot: Implemented an end-to-end Retrieval-Augmented Generation (RAG) chatbot, enabling seamless retrieval of relevant information from a corpus of documents. This implementation empowers the chatbot to generate context-aware responses, ensuring the delivery of accurate and pertinent answers to user queries.

3. Multilingual Functional Chatbot: Designed and implemented a functional chatbot capable of operating in both English and Portuguese languages. The chatbot can proficiently retrieve relevant information from documents and generate context-aware responses in both languages, thereby enhancing its usability and accessibility across diverse linguistic contexts.

1.4 Structure of the document

This introduction outlines the primary objective of the research: to develop a chatbot that accurately responds to queries based on a provided document while minimizing the occurrence of hallucinations in LLMs. The research focuses on using Retrieval-Augmented Generation (RAG) to enhance the chatbot's context-awareness and reliability.

The rest of this report is structured as follows:

Chapter 2 - Background This section delves into the historical evolution of chatbots, tracing their development from early prototypes to modern advanced systems. It begins with a discussion on Alan Turing's proposal of the Turing Test, a benchmark for determining machine intelligence. Then it is highlighted various categories of chatbots such as rule-based systems, conversational agents, and the latest advancements driven by machine learning and artificial intelligence.

Moreover, the section addresses the challenges posed by hallucinations in Large Language Models (LLMs), which pose significant reliability issues, especially in critical domains like healthcare and finance. To counteract this, this section explores the potential of Retrieval-Augmented Generation (RAG) as a promising solution. It also defines the evaluation metrics used in this work.

Chapter 3 - Related Work The related work section reviews various methodologies and technologies focused on mitigating hallucinations. It discusses several advanced prompting techniques, which iteratively guide models to reduce the generation of false information. Another example are knowledge retrieval methods, which identify and validate potential hallucinations before they can affect model outputs.

Furthermore, the section details various methods within each phase of Retrieval Augmented Generation (RAG).

Chapter 4 - Data The data section describes the datasets used for evaluating the chatbot. For the Portuguese language component, a comprehensive PDF guide detailing local accommodation options in Portugal is utilized. For the English language component, a diverse dataset encompassing various topics is used. This section explains the selection of these datasets and their relevance to the research objectives.

Chapter 5 - Methods This section provides a detailed description of the chatbot's development, structured into six phases: data preprocessing, embedding model, vector database, conversational chain, response generation, and interface and deployment. Each phase is meticulously outlined, illustrating how the chatbot retrieves relevant information from documents and generates context-aware responses.

Chapter 6 - Results and Discussion In this section, the experimental results of the chatbot's performance are analyzed. The evaluation metrics are presented. The results highlight the chatbot's capabilities and limitations, providing insights into areas for improvement. The discussion reflects on the effectiveness of the proposed methodology.

Chapter 7 - Conclusion The conclusion summarizes the findings of the research, emphasizing the implementation of a RAG chatbot and its potential utility. It acknowledges the limitations encountered and proposes directions for future work.

The content outlined in this document was created during my internship at NLX-Natural Language and Speech Group [1], a research team dedicated to Natural Language Processing at the Faculty of Sciences, University of Lisbon.

[1] https://nlx.di.fc.ul.pt/, accessed in 04/09/2024

Chapter 2

Background

In this background section, I delve into the historical evolution of chatbots, exploring their development over time. I highlight various categories of chatbots, with a particular emphasis on the evolution of response generation methods. Additionally, I examine the advantages and disadvantages inherent in chatbots. Furthermore, I address the challenges posed by hallucinations in LLMs, a crucial consideration in chatbot research. Additionally, I explore the promise of Retrieval-Augmented Generation as a solution to enhance chatbot functionality. Finally, I discuss the evaluation metrics employed in this study.

2.1 Chatbot history

In 1950, Alan Turing proposed the Turing Test as a measure of intelligence for machines. The test involves a human judge conversing with both a human and a machine via text, without knowing which is which. If the judge cannot distinguish between the two, then the machine passes the test and is considered intelligent [TURING, 1950].

In 1966, Joseph Weizenbaum developed Eliza [Weizenbaum, 1966], a chatbot designed to mimic a psychotherapist. It used a simple pattern-matching algorithm and a set of predefined answers to respond to user input, but it was able to create the illusion of human conversation, making it the first chatbot to capture the public's attention. Parry [Colby et al., 1971], created in 1971, was designed to simulate a patient with schizophrenia and responded based on assumptions and emotional responses activated by the user's input. Although more advanced than Eliza, its capabilities in understanding language were still limited. In 1995, Alice was developed [Wallace, 2009]. It uses a more advanced language understanding system based on Artificial Intelligence Markup Language (AIML) and a knowledge base of 41,000 templates compared to ELIZA that had only 200 keywords and rules [Adamopoulou and Moussiades, 2020]. Alice was considered one of the most advanced chatbots of its time, was able to converse on a wide range of topics, and it won the Loebner Prize, an annual competition for chatbots that can converse like a human in years 2000, 2001, and 2004.

Apple's Siri [1], Microsoft's Cortana [2], Amazon's Alexa [3], Google's Assistant [4] and IBM's Watson [5]

[1]https://www.apple.com/siri/, accessed in 04/09/2024
[2]https://www.microsoft.com/en-us/cortana, accessed in 04/09/2024
[3]https://alexa.amazon.com/, accessed in 04/09/2024
[4]https://assistant.google.com/, accessed in 04/09/2024
[5]https://www.ibm.com/watson, accessed in 04/09/2024

were the first smart personal voice assistants that could cope with voice commands to perform tasks such as sending messages, managing calendars, and controlling home automation devices.

More recently, significant progress in NLP has led to the creation of advanced chatbots. A prominent example is ChatGPT, released by OpenAI in 2022. Unlike earlier chatbots, it uses the transformer architecture [Vaswani et al., 2017], which enables it to comprehend the context of a conversation. It can perform a wide range of tasks, including writing code, books and providing customer support.

The development of European Portuguese chatbots has been limited due to the scarcity of available resources in contrast to widely spoken languages like English. Researchers have recently developed a Portuguese disaster chatbot, DisBot [Boné et al., 2020], a chatbot that informs users about disasters and how to respond to them effectively. Unlike previous systems that focused only on water-related disasters, DisBot is capable of providing information on various types of disasters, including floods, wildfires and earthquakes. DisBot is highly specialized in the area of natural disasters.

2.2 Chatbot categories

Chatbots, with their ability to simulate human-like conversations, have become increasingly prevalent in various industries. They can be categorized based on several criteria, such as the mode of interaction, knowledge domain, application, and the design techniques employed in their development [Hussain et al., 2019].

2.2.1 Mode of interaction

When considering the mode of interaction, chatbots can be classified into different types based on the medium through which users engage with them. These mediums include text-based, voice-based, and embodied interactions. Text-based chatbots enable users to communicate by simply typing their messages through a keyboard. This form of interaction is widely used and familiar to users, allowing for seamless communication with the chatbot. On the other hand, voice-based chatbots utilize microphones, enabling users to engage in conversations by speaking. This mode of interaction is particularly advantageous for older adults and individuals with special needs, as it eliminates the need for typing and accommodates different communication preferences.

In addition to text-based and voice-based chatbots, there are also embodied chatbots. These chatbots possess a physical body, often resembling humans or cartoon animals. The inclusion of a physical form allows embodied chatbots to exhibit facial expressions and emotions, enhancing the user experience and creating a more engaging interaction. By incorporating visual cues and non-verbal communication, embodied chatbots can establish a more natural and intuitive connection with users [Kuhail et al., 2023].

2.2.2 Knowledge domain

The distinction between open domain and closed domain chatbots lies in their scope of knowledge and the extent to which they can provide relevant information. Open domain chatbots excel in engaging users in a wide range of topics, while closed domain chatbots are more focused and specialized in a specific domain [Nimavat and Champaneria, 2017].

2.2.3 Chatbot application

Task-oriented chatbots are specifically designed to perform a particular task or assist users in completing a specific objective. These chatbots typically engage in short conversations within a closed domain, focusing on productivity and providing informative and accurate answers. Their primary function is to guide users through a predefined process or help them accomplish a specific task efficiently.

In contrast, non-task-oriented chatbots are designed to simulate conversations with users, often in open domains, with the aim of providing entertainment and acting as a friendly conversational partner. These chatbots prioritize chit chat and aim to engage users in casual and enjoyable conversations. While they may not have a specific task or objective, their purpose is to entertain and provide reasonable responses that mimic human-like conversation [Hussain et al., 2019], [Xie and Farooq, 2000].

2.2.4 Design techniques (response generation method)

Categorization based on how inputs are processed and responses are generated considers the approach used in handling inputs and creating replies. This classification involves three distinct modes for generating suitable responses: the rule-based approach, retrieval-based approach, and generative model [Hussain et al., 2019].

The **rule-based approach** in chatbots represents a foundational architecture that many early chatbots, including numerous online ones, have been constructed upon. These chatbots determine their responses by following a predefined set of rules, primarily focused on recognizing the structure and form of the input text rather than generating new textual responses. The knowledge utilized by these chatbots is typically manually coded by humans and is structured in a way that aligns with conversational patterns. A larger and more comprehensive rule database enables the chatbot to handle a wider range of user inputs. However, this model is susceptible to errors stemming from spelling and grammatical mistakes in user inputs [Hussain et al., 2019].

The **retrieval-based approach** in chatbots presents a distinct approach from the rule-based, offering enhanced flexibility through the utilization of APIs to query and analyze available resources. In this mode, the chatbot retrieves potential response candidates from an index before employing matching algorithms to select the most suitable response for the ongoing conversation, resulting in more informative and fluent responses compared to rule-based systems. These chatbots leverage response selection algorithms to choose appropriate responses from a repository of pre-existing responses, ensuring that responses are contextually relevant and coherent with the ongoing conversation [Hussain et al., 2019], [Wu et al., 2016], [Qian et al., 2021].

Recent advancements in machine learning, particularly in artificial neural networks, have enabled the development of more intelligent chatbots. Unlike rule-based models, **generative-based chatbots** incorporate learning algorithms, allowing them to adapt and improve over time based on the data they receive. This distinction marks a significant shift in chatbot development approaches towards more dynamic and adaptive systems.

Artificial neural networks (ANNs) are a technological innovation inspired by the study of the brain and nervous system. These networks mimic biological neural networks but simplify the concepts from biological systems. ANNs model the electrical activity of the brain and nervous system by connecting processing elements (also known as neurons or perceptrons) to each other. These processing elements are typically organized in layers or vectors, where the output of one layer becomes the input for the next layer and potentially other layers. Each neuron may be connected to all or a subset of neurons in the subsequent layer, simulating the synaptic connections in the brain. Weighted data signals entering a neuron represent the electrical excitation of a nerve cell and the transfer of information within the network or brain. The input values to a neuron are multiplied by connection weights, which simulate the strengthening of neural pathways in the brain. By adjusting these connection strengths or weights, ANNs emulate the learning process, allowing them to adapt and improve their performance over time [Walczak, 2019].

Natural Language Processing (NLP) encompasses a broad spectrum of techniques and algorithms aimed at interpreting text. Within the realm of Machine Learning, a significant emphasis has been placed on Sequence-to-Sequence learning, often referred to as Seq2Seq which employs a technique where the objective is to convert one sequence into another by acquiring an intermediary representation capable of executing the desired transformation. The fundamental elements consist of an encoder and a decoder network. The encoder converts each element into a hidden vector that encapsulates both the item itself and its surrounding context. Conversely, the decoder operates in reverse, transforming the vector back into an output item, with the previous output serving as the input context.

To delve deeper into the specific techniques used in Seq2Seq learning, I will first explore the vanilla variant of Recurrent Neural Networks (RNNs). Following this, I will introduce more advanced models such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs). Additionally, I will discuss the attention mechanism, a pivotal concept in improving the focus of Seq2Seq models. Finally, I will explore the Transformer architecture.

Recurrent Neural Networks (RNNs) represent a class of artificial neural networks designed to handle sequential data effectively. The architecture is represented in Figure 2.1. Unlike traditional neural networks, RNNs possess the ability to retain memory of previous computations, allowing them to capture temporal dependencies and context within sequential data, such as natural language sentences or time-series data. This capability is achieved through a recurrent structure where the output of each step is fed back as input to the next step, thus enabling the network to consider past information while processing current inputs. This recurrent nature makes RNNs well-suited for tasks where understanding sequential patterns and context is crucial, such as language modeling, speech recognition, and time-series prediction. However, the vanishing gradients problem is a significant challenge that affects the training of RNNs. It occurs when the gradients (derivatives of the loss function with respect to the network's parameters) become very small during backpropagation. This means that the updates to the network's parameters (weights and biases) become very small or negligible, leading to slow or halted learning. The vanishing gradients issue impacts the ability of RNNs to capture long-range dependencies in sequences effectively. For instance, in natural language processing tasks, where understanding context over long sentences or documents is crucial, vanishing gradients can hinder the model's ability to retain relevant information from earlier parts of the sequence [Salehinejad et al., 2018], [Hochreiter, 1998].

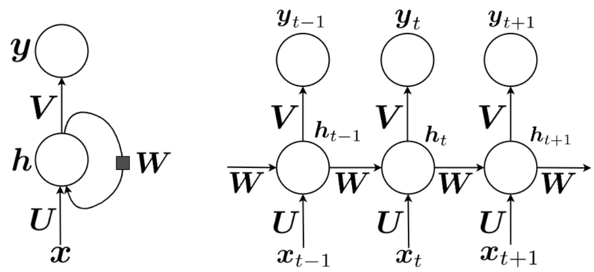


Figure 2.1: RNN architecture, retrieved from [Ghojogh and Ghodsi, 2023]

Long Short-Term Memory (LSTM) networks, represented in Figure 2.2, have emerged as a pivotal advancement in recurrent neural network (RNN) architectures, specifically designed to overcome the long-term dependency problem encountered in traditional RNNs. Unlike standard RNNs, which struggle to retain information over long sequences due to vanishing gradients, LSTMs introduce memory cells and gating mechanisms to effectively store and manage information over extended periods. At the heart of LSTM networks are specialized components known as gates: the input gate, forget gate, and output gate. These gates regulate the flow of information within the network, enabling selective retention, deletion, and utilization of past information. The input gate determines which new information should be incorporated into the memory cell, while the forget gate decides which existing information should be discarded. The output gate controls the dissemination of information from the memory cell to the network's output.

By integrating these gating mechanisms, LSTMs empower networks to learn from experience and maintain contextual understanding over prolonged sequences. Consequently, LSTMs have supplanted traditional RNNs as the standard choice for sequence modeling tasks, offering superior performance in classification, processing, and prediction of time series data, even in scenarios with long intervals between significant events. In the realm of natural language processing and chatbot design, LSTMs excel in preserving contextual information across conversational turns, thus facilitating the generation of contextually relevant responses [Hussain et al., 2019], [Graves and Graves, 2012], [Van Houdt et al., 2020].

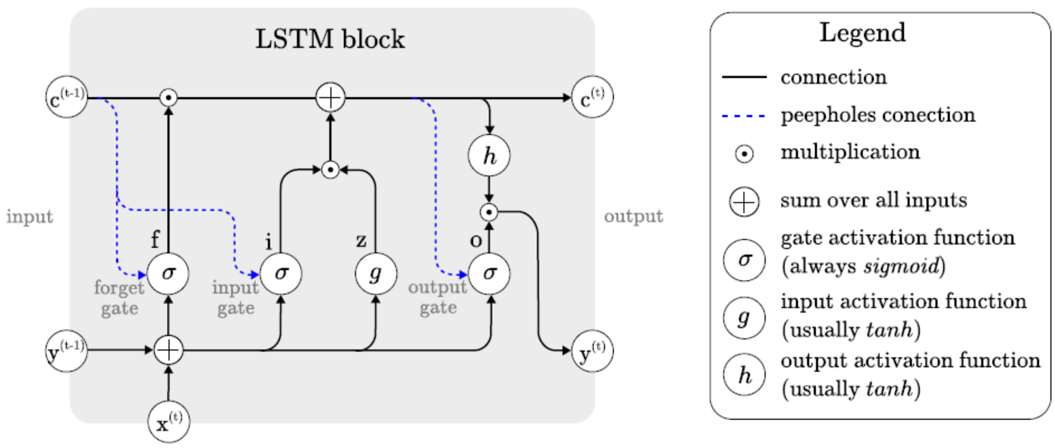


Figure 2.2: LSTM architecture, retrieved from [Van Houdt et al., 2020]

Introduced as a simplified variant of LSTM networks, Gated Recurrent Units (GRUs), illustrated in

Figure 2.3, represent another significant development in recurrent neural network architectures. While sharing similarities with LSTMs, GRUs exhibit distinct characteristics, offering a balance between computational efficiency and memory retention.

GRUs consist of two primary gates: the reset gate and the update gate. The reset gate regulates the degree to which past information influences the current state, while the update gate determines the extent to which new information is incorporated into the current state. Unlike LSTMs, GRUs lack an explicit output gate, resulting in fewer parameters and faster training times. Despite their streamlined architecture, GRUs have demonstrated competitive performance in various sequence modeling tasks, albeit with some limitations compared to LSTMs. While GRUs are generally faster to train due to their reduced complexity, they may struggle with certain tasks that require extensive memory retention capabilities, such as learning complex languages or capturing long-range dependencies in sequences [Chung et al., 2014], [Dey and Salem, 2017].

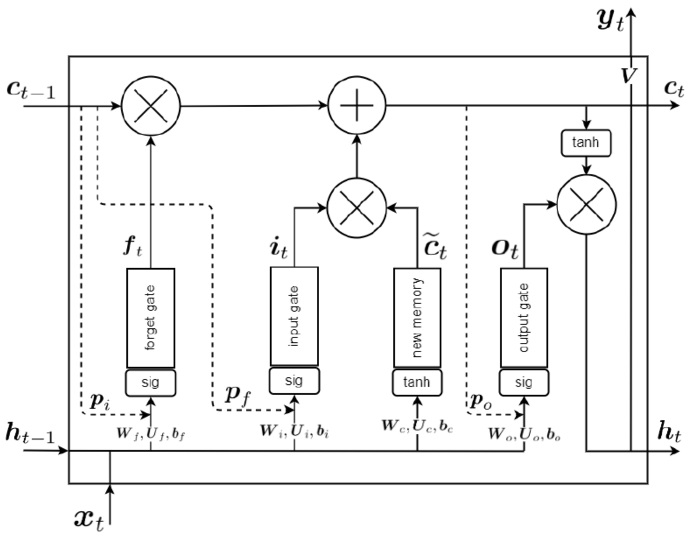


Figure 2.3: GRU architecture, retrieved from [Ghojogh and Ghodsi, 2023]

Despite the improvements in backward error computation achieved by Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures compared to traditional recurrent neural networks (RNNs), they still encounter challenges when processing long sequences, highlighting the limitations of sequential data handling. LSTMs and GRUs update their memory based on short-term relationships, leading to the forgetting of longer-term dependencies over time.

To address this issue, researchers introduced the attention mechanism, which allows models to consider all intermediate states during computation rather than just the last one, improving output predictions by focusing on relevant parts of the input. However, even with attention, the sequential nature of computation remained a bottleneck for vanilla RNNs, LSTMs, and GRUs. This limitation spurred the development of Transformers.

The groundbreaking work of [Vaswani et al., 2017] marked a significant milestone in the field of natural language processing and deep learning. Their introduction of the Transformer architecture revolutionized the way researchers approached sequence modeling by shifting the focus from recurrent structures to an attention-based framework. By emphasizing the power of attention mechanisms and eliminating the need for recurrent connections, the Transformer model improved significantly computational efficiency and performance, laying the foundation for more advanced and effective sequence processing techniques.

The Transformer architecture, represented in Figure 2.4 consists of two main components: the encoder and the decoder segment.

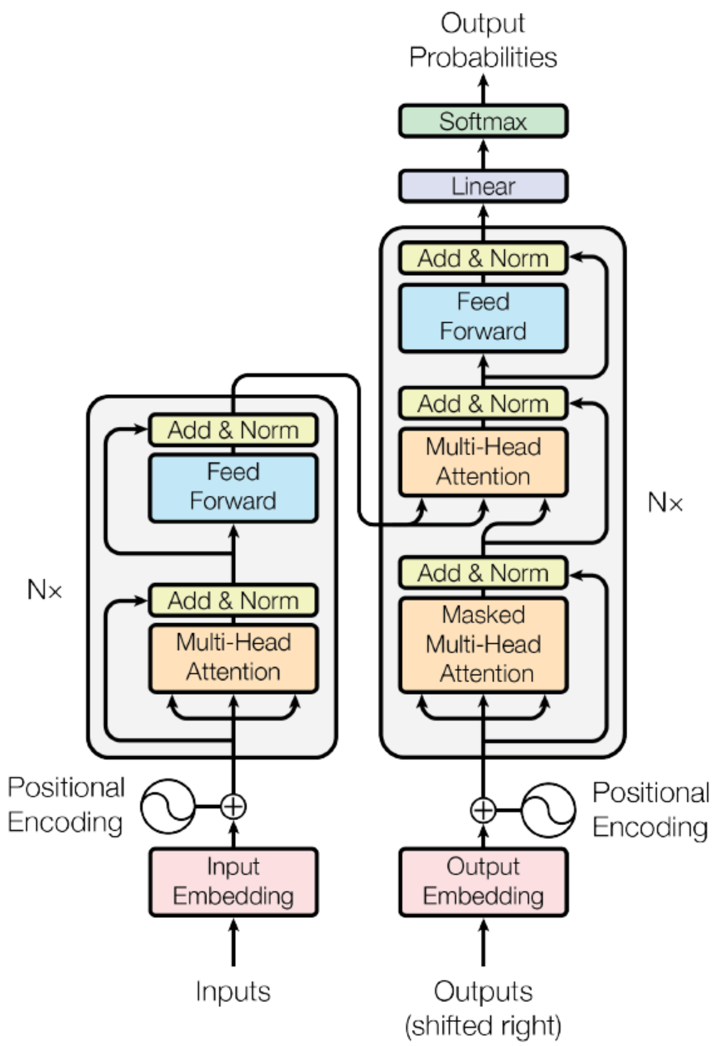


Figure 2.4: Transformer architecture, retrieved from [Vaswani et al., 2017]

The encoder segment is responsible for converting an input sequence into an intermediate representation, which serves as the foundation for subsequent processing in the model. It comprises several crucial elements that capture the semantic and positional information of the input sequence.

Firstly, a learned embedding layer is employed to transform the input tokens into a numerical representation that can be processed by the model. This embedding layer allows the model to capture the semantic meaning of words and their contextual relationships within the input sequence.

In addition to the embedding layer, the encoder segment incorporates positional encoding to provide information about the position of tokens within the input sequence. Since Transformers do not rely on recurrent connections, positional encoding is essential for conveying sequential information to the model. By encoding the position of each token in the input sequence, the model can understand the order and structure of the input data, enabling it to process sequential information effectively.

Furthermore, the encoder segment can be repeated multiple times to enhance the precision of the encoding process. This encoder part typically comprises two main components: a multi-head attention mechanism and a feedforward network. The multi-head attention mechanism allows the model to focus on relevant parts of the input sequence by splitting the inputs into queries, keys, and values. This mechanism facilitates capturing long-range dependencies and identifying important information within the input sequence. Additionally, the feedforward network within the encoder segment generates high-dimensional representations for individual inputs, further enhancing the model's ability to extract meaningful features from the input data.

The use of residual connections, which pass the original input to the output, helps to mitigate the vanishing gradient problem and allows for smoother training of the model by enabling gradients to flow freely during backpropagation.

The decoder part of the Transformer architecture is responsible for generating output tokens based on the intermediate representation obtained from the encoder segment.

Similar to the encoder, the decoder begins with a learned embedding layer that transforms input tokens into a numerical representation suitable for processing by the model. This embedding layer captures the semantic meaning of words and helps the model understand the context of the input tokens.

Just like in the encoder, the positional encoding in the decoder adds information about the position of tokens in the input sequence. This ensures that the model can maintain the sequential order and structure of the input data, despite not using recurrent connections.

The decoder part is repeated N times. Each repetition enhances the decoder's ability to generate accurate output sequences. In the decoder, a masked multi-head attention mechanism is used to generate self-attention for desired outputs. This mechanism ensures that during training, the model can only attend to previous positions and prevent information leakage from future positions. It helps in predicting the next word in a sequence based on the previously generated tokens.

Another crucial aspect of the decoder is another multi-head attention segment. This segment merges information from the encoder's outputs with the self-attention outputs from the decoder. It allows the model to align input tokens with their corresponding output tokens, facilitating tasks like translation and summarization. This attention mechanism enables the model to focus on relevant parts of the input sequence while generating the output sequence.

The decoder also includes a feedforward network, similar to the one in the encoder. This network processes the combined information from the attention mechanisms and generates high-dimensional representations for each output token.

The use of residual connections in the feedforward network helps with gradient flow and improves training stability.

Finally, the predicted outputs from the feedforward network are added back into the decoder segment, allowing the model to predict the next output token based on the encoded input token and previous predictions. This iterative process, along with the residual connections, helps improve the model's ability to generate accurate and contextually relevant output sequences.

This approach has enabled professionals in Natural Language Processing to attain highly impressive outcomes concerning text processing, encompassing tasks such as language generation, summarization, and translation. By addressing challenges related to long-term memory using attention mechanisms and enhancing computational speed through the removal of recurrent segments, this method has reduced the reliance on recurrent neural networks as the primary option for creating language models. However, many LSTM models remain prevalent and actively utilized in current practices.

2.3 Advantages and disadvantages of chatbots

Some vantages of chatbots include:

1. Efficient customer interaction: Chatbots can engage visitors in conversation without the need for them to fill out numerous forms. They use widget interaction, allowing visitors to choose their intent by clicking on one of several options. This can help convert visitors into customers and bring them into the conversation more easily [Meshram et al., 2021].
2. 24/7 availability: Chatbots can maintain a 24/7 response system, answering queries at any time. This is particularly useful in the business world, where customers may have to wait for a long time to get their queries answered [Meshram et al., 2021].
3. Improved efficiency: Chatbots can improve efficiency by taking over tasks that do not require human intervention [aza et al., 2018].
4. Broad reach: Chatbots can reach a wide audience on messenger systems and can automate personalized messages [aza et al., 2018].
5. Facilitated communications: Chatbots are helpful in facilitating customer satisfaction and communication.

And some disadvantages are:

1. Lack of emotions: Chatbots are pre-programmed with conversations and may lack the ability to understand and respond to emotions [Meshram et al., 2021].
2. Referent resolution: Chatbots may struggle with resolving referents, which can lead to misunderstandings in conversation.
3. Lexical ambiguity: Chatbots may have difficulty understanding words with multiple meanings, leading to potential miscommunications.
4. Ellipsis in linguistics: Chatbots may not understand ellipses in language, which can make conversations more difficult.
5. Limitation of answers to closed questions: Chatbots may only be able to provide answers to closed questions, limiting the scope of conversation.
6. Rigidity: Chatbots may be rigid in their responses, which can make conversations feel less natural.
7. Robotic tone: Chatbots may have a robotic tone, which can make conversations feel less personal.

8. Impersonality: Chatbots may lack the ability to create personal connections with users, which can make conversations feel less engaging.

9. Fear of Incorrect Information: There is a fear among some users that chatbots may provide incorrect information, which can lead to mistrust and dissatisfaction [Arsenijevic and Jovic, 2019].

2.4 Hallucinations

The term "hallucination" is used to describe the undesired phenomenon of NLG (Natural Language Generation) models to generate content that is not present or implied in the input data (external source). This can be manifested in various forms, such as generating words or sentences that are not present in the data used to train the model. Hallucinated text appears to be fluent and natural, but it is unfaithful or nonsensical with the input source. There are two main types of hallucinations in NLP: intrinsic and extrinsic [Ji et al., 2023].

Intrinsic hallucination [Maynez et al., 2020] refers to the generated output contradicting the input or training data. For example, if the model generates the sentence "The sky is red" when the input says that the sky is blue, it is considered an intrinsic hallucination. Extrinsic hallucinations [Huang et al., 2021] occur when a model generates text that cannot be verified from the input or training data. It includes information that is neither supported nor contradicted by the data. For example, if a model generates "The sun is yellow" when the input text does not mention anything about the sun, it is considered an extrinsic hallucination.

Intrinsic hallucinations are considered a more severe problem than extrinsic hallucinations, as they can lead to factual errors and contradictions in the generated text. Extrinsic hallucinations, while less severe, can still pose problems as they can generate text that is not relevant or useful in the context.

It is important to note that hallucination in NLG is a significant concern due to its negative impact on performance and the potential safety risks it poses in real-world applications. For example, in medical applications, a hallucinatory summary generated from a patient information form could pose a risk to the patient, potentially leading to life-threatening incidents. Additionally, hallucinations can also result in potential privacy violations [Ji et al., 2023].

2.5 Retrieval-Augmented Generation (RAG)

Lewis et al. [Lewis et al., 2020] introduced the concept of Retrieval-Augmented Generation (RAG): the process of first retrieving relevant information from external sources, and then the LLM uses this information to create an answer. By retrieving data that serves as a reference to organize answers, RAG significantly enhances the accuracy of responses which helps to solve the hallucinations problem. Since the emergence of LLMs, RAG has gained significant interest and is now considered an important technology for improving chatbots and making LLMs more functional. It enables the optimization of a large language model's output by incorporating targeted information without modifying the underlying model itself. RAG consists of three essential steps:

1. Augmented: The documents are broken down into smaller segments and vector embeddings are created for each segment using an embedding model.

2. Retrieval: By comparing the similarity of the user question and the segments, relevant segments are obtained.

3. Generation: Using the relevant sections as a starting point, the system generates an answer to the question.

The benefits of employing RAG include the following:

1. Access to fresh information: RAG enables the generative AI system to tap into information that is more up-to-date compared to the data used during the training of the LLM. This ensures that the responses provided by the system are based on the most recent and relevant data available.

2. Continuously updated knowledge repository: The knowledge repository of RAG can be continually updated without incurring significant costs. This means that new data can be seamlessly incorporated into the system, ensuring that the information used for generating responses remains current and accurate.

3. Contextual data: The knowledge repository of RAG contains data that is more contextual in nature compared to a generalized LLM. This contextual information allows the generative AI system to provide responses that are better aligned with the specific context of the user's query, resulting in more contextually appropriate answers.

4. Source identification: RAG leverages a vector database, which enables the generative AI system to identify the specific source of information cited in its answers. This capability allows for transparency and traceability, as the source of the information can be easily identified and verified.

2.6 Evaluation Metrics

Finally, let us analyze the metrics utilized in this study to evaluate the quality of the obtained results, focusing on two main aspects: retrieval and generation. For the retrieval evaluation, I selected MRR@10, MAP, and R@1000. MRR@10 is the official metric for this task based on the chosen dataset, while the other two are commonly employed in the literature. As for generation, I opted for the widely adopted ROUGE and BLEU metrics, along with cosine scores which are more relevant for assessing RAG performance.

The Reciprocal Rank (RR) information retrieval measure calculates the reciprocal of the rank at which the first relevant document was retrieved. So, if the first relevant document appears at rank 1, RR is 1; if it appears at rank 2, RR is 0.5, and so on. The Mean Reciprocal Rank (MRR) @k is the average of the top k retrieved items across queries. In this case, k=10 [Zhu et al., 2021].

The Average Precision (AP) is determined by finding the weighted average of precision for a specific query within a set of queries. It is computed using the following equation:

$$AP = \frac{\sum_{x=1}^{n} P(k) * R(k)}{number\ of\ the\ relevant\ documents} \tag{2.1}$$

where k represents the rank in the sequence of retrieved documents, n is the number of retrieved documents, P(k) is the precision at cut-off k in the ranked list of documents, and R(k) is an indicator function that equals 1 if the item at rank k is a relevant document, and 0 otherwise. The average is taken over all relevant documents, and in cases where relevant documents are not retrieved, the precision score is considered to be zero.

The Mean Average Precision (MAP) is a score used to assess the quality of the ranked retrieval list of answers. It is calculated as the mean of the AP scores for all encountered relevant documents in each query. The formula for MAP is:

$$MAP = \frac{1}{N}\sum_{x=1}^{n} AP_i \tag{2.2}$$

where N represents the total number of queries, and APi is the average precision score for encountered relevant documents in each query [Farea et al., 2022].

The recall is a metric that calculates the ratio of correctly predicted answers to the total number of correct answers. It assesses the proportion of relevant (correct) documents among all the possible documents. The score obtained for each query is a binary value that indicates whether or not the document is included in the selection [Farea et al., 2022]. Recall at K evaluates the percentage of accurately identified relevant items within the top k recommended items out of the total number of relevant items in the dataset. In this case k=1000.

All of the three previous metrics vary between 0 and 1, and the closer to 1, the better.

ROUGE [Rouge, 2004] stands for Recall-Oriented Understudy for Gisting Evaluation and it focuses on evaluating the quality of summaries. It measures the overlap between the system-generated summary and a set of reference summaries. ROUGE-L uses the longest common subsequence (LCS) between the generated summary and the reference summary, where the matched words are not necessarily consecutive.

BLEU [Papineni et al., 2002] stands for Bilingual Evaluation Understudy and it assesses the quality of machine-generated translations by comparing them to one or more reference translations. BLEU calculates the precision of n-grams (contiguous sequences of words) in the system's translation that match with the reference translations. It then combines these precision scores to calculate the overall BLEU score, which indicates the similarity between the system's translation and the reference translations.

While these last two metrics were originally designed for assessing the quality of summaries and translations, they are widely used in the literature due to the lack of specific metrics tailored for evaluating question and answer systems.

Cosine similarity is a measure used to determine the similarity between two vectors in a multi-dimensional space. In the context of text analysis, each vector represents a document or a query, where the dimensions correspond to the frequency of specific words or terms. The cosine similarity is calculated by finding the cosine of the angle between the two vectors, which ranges from -1 to 1. A cosine similarity of 1 indicates that the two vectors are pointing in exactly the same direction, while a cosine similarity of -1 indicates that the two vectors are pointing in exactly opposite directions. A cosine similarity of 0 indicates that the two vectors are perpendicular to each other. In information retrieval, cosine similarity is commonly used to rank the relevance of documents to a given query, with higher similarity scores indicating greater relevance [Rahutomo et al., 2012].

All of the above formulas were utilized from python libraries that had implementation for these formulas, namely sentence_transformers, sentence_bleu, rouge_scorer. Also, the plots to analyze its results were also used from the following python libraries: matplotlib, seaborn.

Chapter 3

Related Work

Hallucinations represent a significant challenge in the realm of LLMs and diverse strategies have been proposed to alleviate their occurrence. In this section, I delve into an examination of prevalent hallucination mitigation techniques. Subsequently, I conduct a comparative analysis between the chatbot developed within this study and alternative solutions proposed in the literature. Finally, I elucidate the advancements in Retrieval Augmented Generation methods within the current state of the art.

3.1 Hallucination Mitigation Techniques

Hallucination mitigation refers to the process of minimizing or preventing hallucinations in the responses generated by LLMs [Luo et al., 2024].

In addressing hallucinations, it is typically to divide mitigation methods into two main categories, each tackling different sources of the issue: data-related methods and modeling and inference methods. This section provides an overview and summary of these approaches [Ji et al., 2023].

To mitigate hallucination in language models, one effective approach involves building a faithful dataset. This means creating a dataset with accurate, reliable examples that minimize noise and misleading information. There are several methods for constructing such datasets.

One method is to enlist annotators to generate clean and faithful target sentences from scratch based on provided sources. While this approach ensures accuracy, it may result in a lack of diversity in the dataset [Gardent et al., 2017].

Another strategy for mitigating hallucination involves automatically cleaning the data to reduce semantic noise. This approach targets irrelevant or contradictory information within existing parallel corpora and applies filtering or correction mechanisms to refine the dataset. It is particularly effective when the original data exhibits a low to moderate level of noise [Shen et al., 2021].

Another technique to mitigate hallucination involves information augmentation, which enhances the input data with external information. By incorporating additional knowledge such as entity information, relation triples extracted from source documents, pre-executed operation results, synthetic data generated through replacement or perturbation, and retrieved external knowledge [Bi et al., 2019], the model gains a more comprehensive understanding of the source material. This improved alignment between inputs and outputs aids the model in learning task-related features more effectively, thereby reducing semantic divergence from the source. However, integrating augmented information may present challenges due to discrepancies between the original source and the augmented data, requiring careful management to maintain coherence and fidelity in the training process.

LLM-AUGMENTER [Peng et al., 2023], aims to enhance the performance of LLMs by leveraging external knowledge sources and automated feedback mechanisms. This method acknowledges the necessity of addressing the limitations and potential factual errors present in LLM-generated content. By incorporating external knowledge sources and automated feedback loops, LLM-AUGMENTER seeks to improve the accuracy and reliability of LLM outputs. Its goal is to mitigate factual inaccuracies and enhance the overall quality of text generated by large language models.

Hallucination mitigation using prompting techniques involves a process known as ”dehallucinating” [Jha et al., 2023], which aims to reduce the generation of inaccurate or hallucinated information by LLMs. By employing formal methods, the generation process of the LLM is steered through iterative prompts, with the objective of improving the accuracy and reliability of the generated outputs.

Knowledge retrieval [Varshney et al., 2023] is employed as a proactive approach to detect and mitigate hallucinations in text generated by LLMs. Instead of waiting until after sentence creation, this method utilizes logit output values from the model to identify potential hallucinations beforehand. Once identified, these potential hallucinations are validated for accuracy, and any confirmed instances are addressed to prevent their propagation in subsequent outputs.

FreshPrompt [Vu et al., 2023], addresses the challenge of static nature in most large language models (LLMs) by introducing a dynamic prompting method. This method aims to incorporate current and relevant information from search engines into prompts, enabling LLMs to adapt to the evolving world.

The paper by [Si et al., 2022] addresses the challenge of improving the reliability of LLMs, particularly focusing on GPT-3. While GPT-3 demonstrates impressive few-shot prompting capabilities, its reliability remains an underexplored area. The study identifies four crucial facets of reliability: generalizability, social biases, calibration, and factuality. To enhance GPT-3's reliability, the researchers introduce simple and effective prompts tailored to each facet. These prompts aim to guide the model towards generating more reliable and accurate outputs by addressing issues such as bias, calibration errors, and factual inaccuracies. By utilizing these prompting strategies, the research surpasses smaller-scale supervised models on all reliability metrics, offering practical methods for improving GPT-3's performance.

The Chain-of-Verification (CoVe) [Dhuliawala et al., 2023] method, is a technique designed to mitigate hallucinations in responses generated by LLMs. The process involves several steps: the LLM generates an initial response to a given query or prompt; the model then plans a set of verification questions aimed at fact-checking its initial response; these verification questions are answered independently of the initial response, ensuring that the answers are unbiased and not influenced by the original output; based on the answers to the verification questions, the model generates a final response that has been verified for accuracy and reliability.

The Neural Path Hunter (NPH) [Dziri et al., 2021] model addresses the issue of hallucinations in knowledge-grounded dialogue systems by employing a generate-then-refine strategy. After an initial response is generated by a large language model (LLM), NPH utilizes a token-level fact critic to identify potentially hallucinated entities within the response. This critic focuses particularly on instances of entity misuse, which can lead to factual inaccuracies in the dialogue. By training the critic to flag entities of concern with a binary label prediction at each word position, NPH can identify and refine potentially problematic entities in the dialogue. Leveraging the Roberta-Large model for token classification, the NPH critic is trained with manually introduced negative examples, such as replacing correct entities with incorrect ones or swapping the subject and object within dialogues from datasets like OpenDialKG.

Inference-Time Intervention (ITI) [Li et al., 2024], enhances the ”truthfulness” of LLMs by directing model activations during inference along specific directions identified for truthfulness. This technique significantly improves LLaMA models’ performance on benchmarks like TruthfulQA. By interactively shifting activations along truth-correlated directions, ITI reduces the occurrence of hallucinations in generated responses, resulting in a notable performance boost on the TruthfulQA benchmark.

The modeling and inference methods in natural language generation (NLG) encompass various strategies to address the challenge of hallucination [Ji et al., 2023].

In architectural design, both the encoder and decoder components of NLG models are subject to modifications aimed at improving semantic interpretation and reducing hallucination. For instance, researchers have proposed dual encoders comprising sequential document encoders and structured graph encoders to handle additional knowledge effectively. Attention mechanisms, vital in neural networks for focusing on relevant information, are tailored to encourage models to prioritize pertinent details while disregarding irrelevant ones [Wu et al., 2021]. Decoders, responsible for generating final output in natural language, undergo structural enhancements such as multi-branch or uncertainty-aware designs to mitigate hallucination.

Training methodologies play a crucial role in shaping NLG models to address hallucination. Planning and sketching techniques control the content and order of generated text, either as separate steps or integrated into end-to-end models. Reinforcement learning (RL) offers an avenue for optimizing model performance by rewarding actions that reduce hallucination, with reward functions tailored to incentive faithful outputs [Huang et al., 2020]. Multi-task learning enables models to learn from multiple tasks simultaneously, enhancing their understanding of the target task and reducing hallucination. Controllable generation techniques provide flexibility in adjusting the level of hallucination to align with the requirements of diverse real-world applications.

Post-processing methods serve as a valuable tool for rectifying hallucinations in generated text. This approach is particularly advantageous when dealing with noisy datasets where a significant portion of the ground truth references contain hallucinations. Authors like [Chen et al., 2021] and [Cao et al., 2020] have adopted a generate-then-refine strategy, wherein initial outputs from state-of-the-art models are refined to improve faithfulness. Although this correction step may introduce grammatical errors, it enables the utilization of high-performing models known for their fluency.

3.2 Related projects

The most complete work done so far is h2ogpt [Candel et al., 2023], an Apache v2 open-source project that enables query and summarization of documents using local LLMs. It is compatible with Linux, Docker, MAC, and Windows. H2ogpt is a private offline database of various document types, including PDFs, Excel, Word, images, code, text, and MarkDown. It employs persistent databases like Chroma, Weaviate, or in-memory FAISS and embeddings such as instructor-large and all-MiniLM-L6-v2. Supporting models like LLaMa2[Touvron et al., 2023], Mistral[Jiang et al., 2023], WizardLM[Xu et al., 2023], Vicuna[1] and Falcon[Almazrouei et al., 2023], it includes features like AutoGPTQ, and LORA. The platform offers a user-friendly UI and CLI. Via the UI, users can upload and view documents. Supports inference servers such as HF TGI server, vLLM, Gradio, and Azure OpenAI. Additionally, it supports web search integration, agents for search, document Q/A, and evaluates the performance using reward models. The platform also features parallel summarization, state preservation and authentication.

After testing h2ogpt, I found it to be overly complex to the extent that it severely limits the ability to modify its parameters. Additionally, being a very recent project, utilizing cutting-edge technology, I encountered several bugs related to Python libraries dependencies. My goal with the chatbot I present here is to simplify this process significantly. By making it more stable and user-friendly, potential companies will find it easier to adopt and utilize the chatbot without encountering frequent bugs or difficulties in understanding how to use it.

In comparison to my chatbot, there are several projects where it is possible to ask questions about documents, such as private gpt [2] and local gpt [3], ensuring privacy by keeping data within the local environment. In contrast to Privategpt, this chatbot will utilize a vector database specific designed for similarity search. The importance of this is explained in subsection 5.3. LocalGPT exclusively stores the conversation within a session.

While Quivr [4] and Llama Index [5] provide comparable functionalities, they lack the extensive customization and flexibility found in this chatbot. Quivr necessitates Docker and lacks chat history retention, and Llama Index lacks a user interface.

Unlike my chatbot, which guarantees data privacy, platforms such as DocsGPT [6] and VaultAI [7] have a disadvantage in that they heavily rely on OpenAI. This reliance means that every document submitted to these platforms is transmitted through OpenAI, giving rise to concerns regarding data privacy and security. Moreover, there are also some online platforms that provide similar functionality to this chatbot, such as `https://www.chatpdf.com/`, accessed in 04/09/2024 and `https://chatdoc.com/`, accessed in 04/09/2024. These platforms enable users to engage in conversations with PDF documents. However, my chatbot guarantees control over the data.

With the rise in popularity of ChatGPT, LangChain [8] and LLamaIndex have quickly gained recognition. Both offer a comprehensive set of RAG-related APIs, becoming essential technologies in the era of LLMs. At the same time, new forms of technical stacks are continuously emerging. For instance, Flowise AI [9] highlights low-code implementation, enabling users to create various RAG applications through simple drag and drop actions [Gao et al., 2024].

[1] https://lmsys.org/blog/2023-03-30-vicuna/, accessed in 04/09/2024
[2] https://privategpt.dev/, accessed in 04/09/2024
[3] https://github.com/PromtEngineer/localGPT, accessed in 04/09/2024
[4] https://www.quivr.com/, accessed in 04/09/2024
[5] https://www.llamaindex.ai/, accessed in 04/09/2024
[6] https://app.docsgpt.cloud/, accessed in 04/09/2024
[7] https://vault.pash.city/, accessed in 04/09/2024
[8] https://python.langchain.com/, accessed in 04/09/2024
[9] https://flowiseai.com/, accessed in 04/09/2024

3.3 RAG

Extensive research has been consistently conducted at each stage of RAG, which are retrieval, augmentation, and generation. This ongoing research aims to enhance the performance and capabilities of RAG in various aspects [Gao et al., 2024].

3.3.1 Retrieval

In RAG, it is essential to retrieve a collection of relevant documents from the data source, making the retrieval stage a critical step.

PROMPTAGATOR [Dai et al., 2022], a technique proposed to enhance the retrieval process in RAG, focuses on the setting of few-shot dense retrieval where each task is accompanied by a brief description and a few examples. By leveraging LLMs as a few-shot query generator, PROMPTAGATOR creates task-specific retrievers based on the generated data. This approach allows for the creation of end-to-end retrievers solely based on a few examples. PROMPTAGATOR addresses the challenge of supervised fine-tuning, particularly in domains with limited data availability.

HyDE [Gao et al., 2023], enhances the retrieval process in RAG by leveraging LLMs to generate hypothetical documents that capture relevance patterns. These documents are then used to retrieve real documents that are similar in meaning, improving the accuracy of the retrieval process.

The process begins by zero-shot instructing and instruction-following LLM to generate a hypothetical document based on the user's query. This document is designed to be relevant, even if it does not actually exist, and captures the essential patterns of the query. Next, an unsupervised contrastively learned encoder encodes the hypothetical document into an embedding vector. This vector identifies a neighborhood in the corpus embedding space where similar real documents are retrieved based on vector similarity.

The RAG pipeline may not always yield improved outcomes by enhancing the retrieval hit rate, as the retrieved documents might not align with the specific requirements of the LLMs. To address this issue, Augmentation-Adapted Retriever (AAR) [Yu et al., 2023] introduces supervisory signals for a pre-trained retriever using an encoder-decoder architecture LM. The LM's preferred documents are identified through FiD cross-attention scores [Izacard and Grave, 2022], and the retriever undergoes fine-tuning with hard negative sampling and standard cross-entropy loss. The refined retriever can then be directly applied to enhance unseen target LMs, resulting in improved performance in the target task.

3.3.2 Augmented

The naive implementation of RAG has certain limitations that hinder its performance, particularly in terms of contextual richness during inference. To overcome these limitations, advanced augmentation techniques have been introduced that incorporate more contextually rich information during inference. I will now discuss some of these techniques and their impact on RAG performance.

RECITE [Sun et al., 2023] introduces a new paradigm for improving the augmentation process in RAG by generating context through direct sampling of paragraphs from LLMs without requiring external corpus retrieval. This approach, called RECITation-augmented gEneration (RECITE), addresses knowledge-intensive NLP tasks by breaking them down into two sub-tasks: knowledge recitation and task execution. Knowledge recitation serves as an intermediate knowledge retrieval step from the model weights, while task execution generates the final outputs. In contrast to retrieval-augmented language models that retrieve relevant documents before generating outputs RECITE first samples one or more relevant passages and then produces the final answers. By using this two-step paradigm, RECITE decomposes the original knowledge-intensive task, enabling LLMs to generate more accurate factual knowledge without relying on external sources.

Ret-LLM [Modarressi et al., 2023] consists in enhancing language models with a memory module. This module enables the models to extract knowledge from text and store it for future reference. When faced with a task, the language model can query the memory module to retrieve additional information that supports its response. The memory module is versatile and can incorporate information from non textual sources like SQL and no-SQL databases, as well as spreadsheets. It also allows for the aggregation of various pieces of information related to a specific concept, even if they are scattered across large documents or multiple sources.

In the field of RAG, it is common practice to have a single retrieval step followed by generation, which can result in inefficiencies. IRCoT [Trivedi et al., 2023] offers a solution by using a chain-of-thought approach to guide the retrieval process and refine it with the acquired retrieval outcomes. Prompting-based large language models (LLMs) are remarkably effective at generating natural language reasoning steps or chains-of-thoughts (CoT) for multi-step question answering (QA). However, LLMs face challenges when the required knowledge is absent or outdated within their parameters. Although using the question to retrieve relevant text from an external knowledge source helps LLMs, a one-step retrieve-and-read approach is insufficient for multi-step QA. This is because what to retrieve depends on what has already been derived, which may rely on what was previously retrieved. To address this issue, IRCoT proposes an interleaved retrieval approach for multi-step QA that combines retrieval with steps (sentences) in a CoT, guiding the retrieval with CoT and using retrieved results to enhance CoT.

3.3.3 Generation

The generator is a vital element of RAG as it plays a crucial role in transforming retrieved information into coherent and fluent text. In RAG, the generator's input extends beyond typical contextual information to include pertinent text segments obtained through the retriever, allowing it to generate more informative and contextually relevant responses.

Filter-reranker [Ma et al., 2023] is a paradigm that combines the strengths of Large Language Models (LLMs) and Small Language Models (SLMs) to improve the generation process in RAG. The approach consists in using SLMs as filters to identify challenging samples, which are then rearranged by LLMs to prioritize the most relevant items at the top, limiting the total number of documents. This not only enhances retrieval efficiency and responsiveness but also addresses the challenge of context window.

Selfmem [Cheng et al., 2023] is a framework that enhances the generation process in RAG by optimizing the generator's role in producing relevant and natural text from retrieved information to meet the user's query needs. It utilizes a retrieval-augmented generator in an iterative manner to create an unbounded memory pool. It employs a memory selector to choose one output as a memory for the next generation round. It enables a retrieval-augmented generation model to elevate itself using its own output, referred to as self-memory. The key insight behind Selfmem is that the text most closely resembling the data distribution during inference is not the training data, but the model's own output. This approach allows Selfmem to generate more natural and effective text by leveraging the model's own output as a form of self-memory, which improves the generation process in RAG.

Chapter 4

Data

In this section, I will provide a detailed description of the data utilized in this study. The primary objective is to facilitate the generation of responses and operations in both Portuguese and English languages. While sourcing an appropriate question and answer dataset in English proved to be straightforward, the availability of suitable options in Portuguese was limited. Consequently, I chose to develop a custom dataset tailored to the requirements of this research.

4.1 MS MARCO - Question Answering

The Microsoft MAchine Reading COmprehension dataset (MS MARCO) [Nguyen et al., 2016] [1], is a substantial resource tailored for non-commercial research endeavors. It primarily focuses on tasks such as machine reading comprehension, question answering, and passage ranking. The dataset comprises user question queries sampled from Bing's search logs, with the passages extracted from web documents retrieved by the Bing retrieval system.

MS MARCO offers several advantages over other Machine Reading Comprehension (MRC) datasets. Firstly, all questions are sourced from anonymized Bing search queries, ensuring real-world relevance. Secondly, the URLs predominantly contain complete web documents, providing additional contextual information to enhance systems. Thirdly, human-generated answers accompany all questions, and in cases where no answer was found in the passages, judges have noted "No Answer Present." Furthermore, certain questions undergo additional human evaluation to generate well-crafted answers suitable for intelligent agents like Cortana, Siri, and Alexa. Lastly, with over 1 million queries, the dataset is sufficiently large to train even the most sophisticated systems and enables data sampling for specific applications.

The current version of the dataset (v2.1) encompasses 1,010,916 unique real queries obtained by sampling and anonymizing Bing usage logs.

The authors propose three distinct tasks using this dataset, which will be pursued in this study:

1. Determine whether a question can be answered with a set of context passages, extract the relevant information and combine it to form a response.
2. Generate a coherent answer (if possible) based on the context passages that can be understood in relation to the question.

[1] https://microsoft.github.io/MSMARCO-Question-Answering/, accessed in 04/09/2024

3. Rank a collection of retrieved passages based on their relevance to a given question.

Each entry in the dataset contains six parameters:

- query_id: A unique identifier for each query, used for evaluations purposes.
- query: A unique query based on initial Bing usage.
- passages: A set of 10 passages, along with their URLs and an annotation indicating if they were utilized to formulate and answer the query. If a passage is marked as is_selected:1, it implies the judge used that passage to create their answer. If a passage is marked as is_selected:0 it means the judge did not use that passage to generate their response. Two passages may come from the same URL.
- query_type: Queries are categorized into one of {LOCATION,NUMERIC,PERSON, DESCRIPTION,ENTITY} using a trained classifier.
- answer: An array of answers created by human judges, typically containing a single answer but approximately 1% contain more than one answer, with an average of 2 answers if there are multiple answers. These answers were written by real people in their own words, rather than being selected from a span of text. The language used in their answer may resemble or match the language in any of the passages.
- wellFormedAnswers: An array of rewritten answers. Most questions have a single answer but around 1% have more than one answer (with an average of around 5 answers if there are multiple answers). These answers were generated by a new judge who reads the original answer and the query and they would rewrite the answer if it did not (i) include proper grammar to make it a full sentence, (ii) make sense without the context of either the query or the passage, (iii) had a high overlap with exact portions in one of the context passages. This process ensures that well formed answers are true natural language and not just span selection. Well Formed Answers can be a more challenging form of question answering because they contain words that may not be present in either the question or any of the context passages.

For the QA task the target output is 'answer'.

To provide a clearer illustration of a dataset entry, an example is presented in Figure 4.1.

Additionally, Table 4.1 provides a comprehensive overview of the lengths of questions, answers, and passages, offering valuable insights into the distribution of textual elements within the dataset. On average, the queries posed by users are relatively short, possibly indicating concise information needs or straightforward inquiries. The answers provided in response to the queries tend to be longer than the queries themselves. It suggests that the answers may contain more detailed explanations or additional contextual information. While slightly shorter than the average length of all answers, the average length of well-formed answers still indicates a substantial amount of information provided in response to the queries, even after the refinement process. The passages from which the answers are derived are significantly longer than both the queries and the answers themselves. It suggests that the answers are drawn from relatively extensive textual sources, providing ample context for understanding and responding to the queries.

The dataset encompasses a vast corpus of 1,010,916 queries, with corresponding judgments totaling 1,026,758. Within this dataset, 15,777 queries have received multiple judgments. Moreover, a significant subset of the queries, totaling 182,669, has undergone an additional refinement process resulting

MS MARCO example

answers

["It is a compound containing an anionic silicon compound."]

passages

"**is_selected**": [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]

"**passage_text**": ["Non-Silicate Minerals. Minerals can be classified as either silicate - that is, containing silicon and oxygen - or non-silicate - that is, lacking silicon. While most of the earth's crust is composed of silicate minerals, several non-silicate minerals are of great importance.", "Silicates constitute the majority of Earth's crust, as well as the other terrestrial planets, rocky moons, and asteroids. Sand, Portland cement, and thousands of minerals are examples of silicates. Silicate compounds, including the minerals, consist of silicate anions whose charge is balanced by various cations.", "For anion, see Orthosilicate (ion). Silicate minerals are rock-forming minerals made up of silicate groups. They are the largest and most important class of rock-forming minerals and make up approximately 90 percent of the Earth's crust. They are classified based on the structure of their silicate groups, which contain different ratios of silicon and oxygen.", "A silicate is a compound containing an anionic silicon compound. The great majority of the silicates are oxides, but hexafluorosilicate ([SiF6]2−) and other anions are also included. Orthosilicate is the anion SiO4−.", "While most minerals are silicates, many non-silicate minerals are found in the earth's crust and are important as well. This lesson will use examples and describe the three major groups of non-silicate minerals, including carbonates, halides and sulfates.", "The main minerals found in many rocks. Silicates are composed of atoms of silicon, oxygen, and elements such as potassium, sodium, or calcium, under great heat and pressure. Silicates make up about one-quarter of the crust of the Earth.", "Any of a large class of chemical compounds composed of silicon, oxygen, and at least one metal. Most rocks and minerals are silicates. Silicates are also one of the main components of bricks. 2. Any mineral containing the group SiO4 in its crystal lattice. Micas and feldspars are silicate minerals.", "The Silicates. Building Blocks of The Earth's Crust. Silicates are the most widespread of the minerals. They are made up of oxygen and silicon the number one and number two most abundant elements in the earth's crust. By themselves they make up over 90% of the weight of the earth's crust. Most rocks are composed mainly of this class of minerals.", "silicates in Science. Any of a large class of chemical compounds composed of silicon, oxygen, and at least one metal. Most rocks and minerals are silicates. Any mineral containing the group SiO4, either isolated, or joined to other groups in chains, sheets, or three-dimensional groups with metal elements.", "The most abundant elements in the Earth's crust are oxygen (46.6%) and silicon (27.7%). Minerals which combine these two elements are called silicates, and combined they are the most abundant minerals on the Earth. The silicates can be organized in terms of their chemical compositions and their crystal structures (indicated by the existance of cleavage planes). They most often contain members of the Big 8 elements."]

"**url**": ["http://study.com/academy/lesson/non-silicate-minerals-chemical-classifications-examples.html", "https://en.wikipedia.org/wiki/Silicate", "https://en.wikipedia.org/wiki/Silicate_minerals", "https://en.wikipedia.org/wiki/Silicate", "http://study.com/academy/lesson/non-silicate-minerals-chemical-classifications-examples.html", "http://www.dictionary.com/browse/silicates", "http://www.thefreedictionary.com/silicate", "http://www.rocksandminerals4u.com/silicates.html", "http://www.dictionary.com/browse/silicates", "http://hyperphysics.phy-astr.gsu.edu/hbase/Geophys/silicate.html"]

query

what are silicates?

query_id

564,862

query_type

DESCRIPTION

wellFormedAnswers

[]

Figure 4.1: MS MARCO

Table 4.1: Length information

	Average length
queries	6.3733
answers	14.9113
well formed answers	13.7895
passages	56.2534

in Well-Formed Answers. Among these refined queries, 14,460 have elicited more than one judgment, suggesting a nuanced understanding and articulation of the answers provided. These statistics underscore the richness and diversity of the MS MARCO dataset, highlighting its suitability for exploring a wide range of question answering tasks with varying degrees of complexity and linguistic nuances.

4.2 MS MARCO - Passage Retrieval

Initially focused on question-answer tasks, the MS MARCO dataset has since evolved to encompass a broader spectrum of search-related challenges. Leveraging the passages and questions provided in the original question-answer dataset, a new dataset comprising 8.8 million passages was curated [2]. The passage full ranking task consists in retrieving the top 1000 passages sorted by relevance.

Moreover, here are some pertinent statistics concerning this dataset [3]:

- Unique Words: The dataset comprises a substantial vocabulary of 7,555,149 unique words. This indicates a diverse range of language usage within the dataset, potentially reflecting the variety of topics and contexts covered.
- Unique Passages: With 8,841,823 unique passages, the dataset contains a vast array of textual content. This abundance of passages suggests a rich and extensive source of information.
- Average Question Length: The average question length is approximately 6.37 words, with a range spanning from 1 to 75 words. This suggests that questions in the dataset tend to be relatively short.
- Average Passage Length: On average, passages contain around 56.25 words, with a range from 1 to 362 words. This indicates that the passages vary widely in length, with some being quite brief and others more extensive, providing a diverse range of textual contexts.
- Top 1000 Dev and Eval: The Top 1000 development and evaluation sets contain 3,895,239 and 3,831,719 unique passages, respectively.

4.3 Guia Técnico do Alojamento Local

To my knowledge, there currently lacks a dataset containing questions, answers, and passages in European Portuguese. To bridge this gap, I procured a PDF document from the web [4], comprising 45 pages, which will serve as the foundation for a simulation of a real-world use case.

The document titled ”Guia técnico - Alojamento local: regime jurídico” serves as a technical guide outlining the legal framework governing local accommodations in Portugal. It encompasses various

[2]https://github.com/microsoft/MSMARCO-Passage-Ranking/tree/master, accessed in 04/09/2024

[3]https://github.com/microsoft/MSMARCO-Passage-Ranking/blob/master/stats.txt, accessed in 04/09/2024

[4]https://business.turismodeportugal.pt/SiteCollectionDocuments/alojamento-local/guia-alojamento-local-fevereiro-2021-compactado.pdf, accessed in 05/12/2023

topics, including establishment types and regulatory requirements.

To construct this dataset, I systematically extracted each paragraph from the document. These paragraphs then serve as contextual information for ChatGPT, which is prompted with the following instruction:

”I’ll provide a sentence for you to generate questions where the answer lies within the sentence. Create five questions in Portuguese from Portugal with answers of a maximum of 10 words, based on the sentence content. Do you comprehend? The sentence is (...)”

Following this instruction, a paragraph is provided. Notably, although there were a total of 253 paragraphs, titles of sections were treated as paragraphs and excluded from the question-answer generation process. Additionally, for 2 particularly lengthy paragraphs, I requested 20 question-answer pairs instead of the standard 5 to maximize data generation. Each question and its corresponding answer were reviewed to ensure coherence and accuracy. My aim was to attain at least three question-answer pairs per paragraph, with consideration given to additional pertinent pairs when available.

In the following graphs, there are detailed statistics concerning the paragraphs retrieved from the document, offering insights into the dataset composition and characteristics.

The majority of paragraphs contain less than 500 characters 4.2. This suggests that the document’s content is typically concise, conveying information efficiently within relatively short passages. Also, the dataset reveals that the majority of paragraphs consist of less than 100 words 4.3, which indicates that the document primarily comprises brief textual segments, likely organized for easy comprehension and reference. Most words fall within the range of 4 to 8 characters 4.4, which indicates that the document tends to feature moderately-sized words.

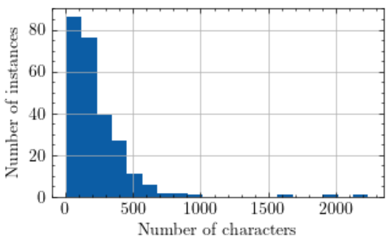


Figure 4.2: Number of characters present in context

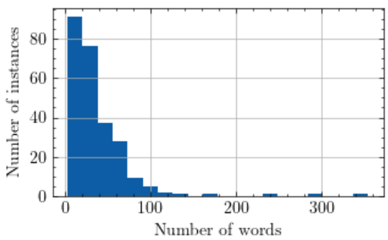


Figure 4.3: Number of words present in context

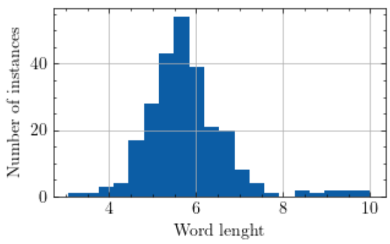


Figure 4.4: Average word length in the sentences of context

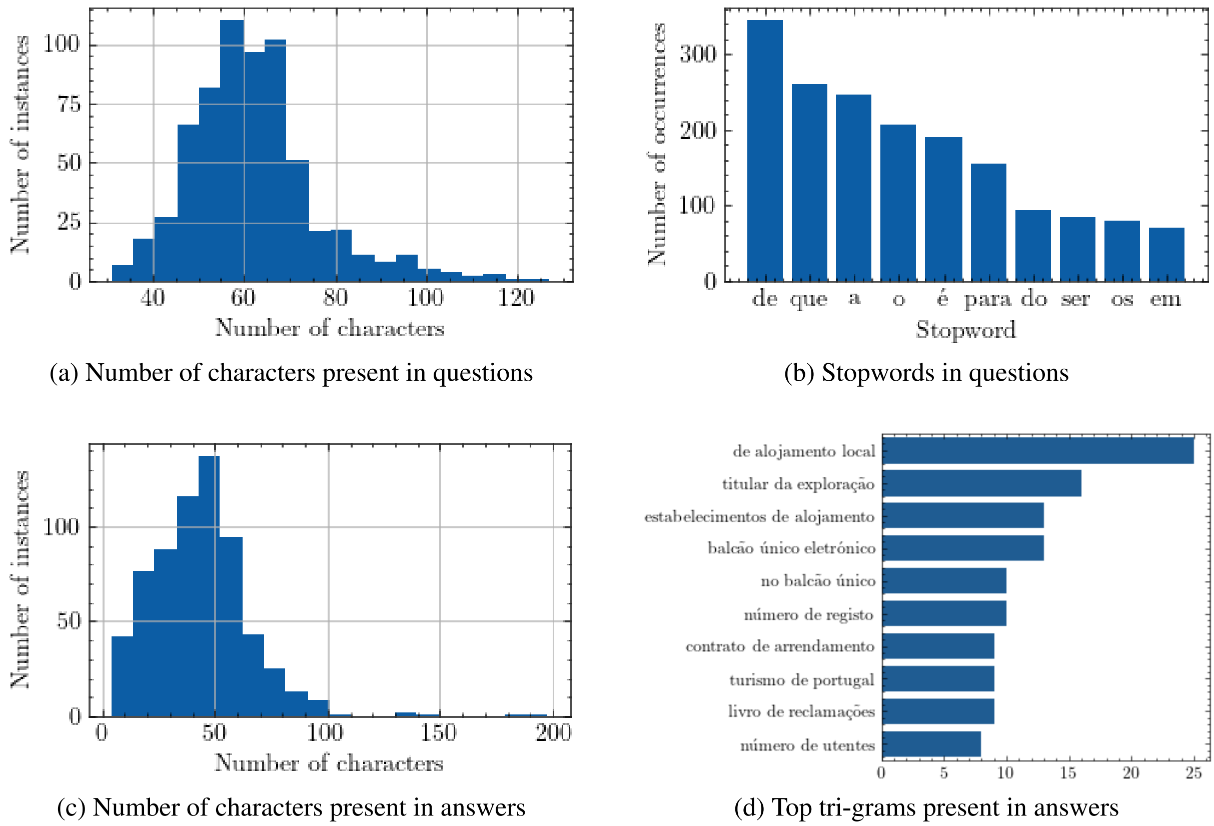
In the end, I acquired 649 entries comprising questions, answers, and contextual information [5]. Detailed statistics regarding the questions and answers can be found in Figure 4.5.

Most questions consist of between 40 and 80 characters which indicates that the questions generated from the dataset are typically short and succinct. Such brevity suggests that the questions are formulated to convey precise information needs or inquiries concisely 4.5a. A significant majority of answers are below 100 characters in length suggesting that the dataset prioritizes concise responses 4.5c.

Stopwords, which are common words with low discrimination value in information retrieval, were analyzed due to their limited informational content and prevalence within the text [Lo et al., 2005]. The most common stopwords in this document are ”de”, ”que”, ”a”, ”o” 4.5b. N-grams, representing contiguous word sequences that recur in a corpus, offer valuable insights into phrase patterns within the

[5] https://huggingface.co/datasets/RitaRodrigues/alojamentolocal

dataset [Cheng et al., 2006]. The top tri-grams shows that the most common answers are about the person responsible for the local accommodation and ”Balcão Único Eletrónico” 4.5d, the website to register local accommodation in Portugal.



(a) Number of characters present in questions

(b) Stopwords in questions

(c) Number of characters present in answers

(d) Top tri-grams present in answers

Figure 4.5: Statistics about questions and answers generated by ChatGPT

The dataset is created solely for evaluation purposes. To utilize the chatbot, the PDF document itself is sufficient.

Chapter 5

Methods

Mitigating hallucinations in LLMs is essential to enhance their reliability and practical applicability. One approach to address this problem, and the one used in this work, is RAG, where relevant information is first retrieved from external sources, and then the LLM uses this information to generate accurate responses.

From now on, I will present an approach to develop a chatbot using the LangChain [1] framework. LangChain aims to simplify the development of applications driven by LLMs. It provides an extensive open-source toolkit accessible in Python and TypeScript, comprising a set of tools that facilitate the creation of applications utilizing LLMs. The framework enables AI developers to seamlessly integrate LLMs like GPT-4 with external computational resources and data sets. Its primary goal is to enable developers to incorporate language processing capabilities into their applications without starting from scratch. LangChain offers a user-friendly approach to interact with LLMs, seamlessly connecting various components and integrating resources such as APIs and databases.

In LangChain, chains are the fundamental concept that organizes the sequence of tasks to be executed. At their core, chains are interconnected components designed to perform specific tasks in a defined order. The most common type of chain within LangChain is the LLMChain, which consists of three main components: a PromptTemplate, a language model (LLM), and an optional output parser. Chains in LangChain are versatile and can be used to carry out various operations on text or other data. A simple chain takes one input prompt and generates an output. Multiple chains can be executed sequentially, where the output of one chain serves as the input for the next. LangChain offers different classes of chains for different scenarios. For instance, the Simple Sequential Chain class is used when there is one input and one output, while the Sequential Chain class is employed when there can be multiple inputs but one output. A common practice in LangChain is to use multiple chains and route inputs to the appropriate chain based on the nature of the input. This is achieved using a router chain, which determines the best suited subchain to process the input before passing it along. Each chain within LangChain can utilize either different or the same LLM, with differentiation primarily based on their prompt. The prompts used in LangChain provide a description of the role of the chain and are combined with user inputs to guide the LLM in producing the desired output [Topsakal and Akinci, 2023].

While chains are structured with predetermined sets of actions, agents employ the analytical capabilities of the LLM in real-time to determine both the actions to take and their sequence. The primary

[1] https://www.langchain.com/, accessed in 04/09/2024

purpose of an agent within LangChain is to leverage a language model in conjunction with a set of actions, employing a reasoning engine to decide on the optimal actions to attain a desired result. Agents play a critical role in handling tasks that range from simple automated responses to complex, context-aware interactions. For instance, an agent integrated with tools like Google Search, Wikipedia, and OpenAI LLM could perform tasks such as searching for results on Google, utilizing retrieved context from Wikipedia to gather detailed information, and expand upon the given context. Agents are particularly useful when an application requires a flexible series of calls to LLMs and other tools based on user input. Depending on the type of agent employed, it can make decisions on the next action using the outputs of previous actions (action agent), or it can devise a full sequence of actions upfront and execute them without further modification (plan-and-execute agent) [Topsakal and Akinci, 2023].

The approach used in this work consists of six phases (see Figure 5.1): (1) Data preprocessing, (2) Embedding model, (3) Vector Database, (4) Conversational Chain, (5) Response generation, and (6) Interface and deployment.

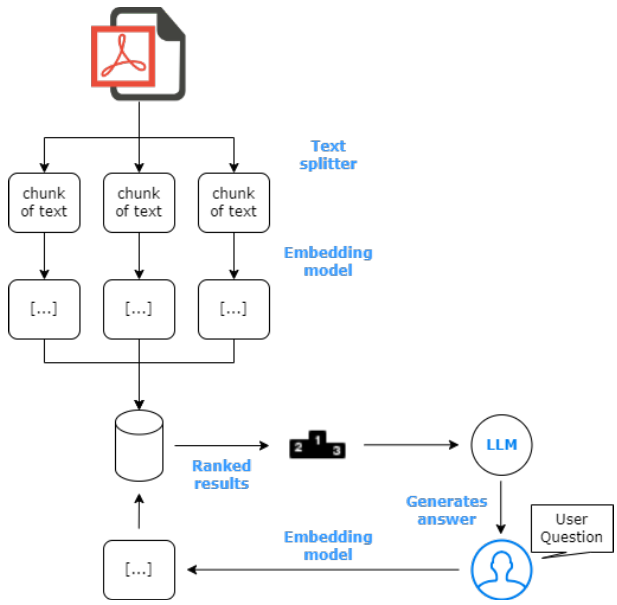


Figure 5.1: Methodology diagram

5.1 Phase 1: Data preprocessing

The first step consists of uploading the document. This chatbot is specifically designed to work with PDFs. After reading the PDF, all of its text is stored as a single string.

Next, the text is divided into chunks of words using a text splitter. The one used here is CharacterTextSplitter from LangChain. This tool starts by breaking down text into smaller units with semantic meaning, such as paragraphs. These smaller fragments are then combined to form larger ones until a specified chunk size, measured by the number of characters, is reached. Once that size is reached, the fragment becomes an independent text unit. Subsequently, a new text fragment is created, with some overlap to preserve context between adjacent fragments.

This text segmentation process is crucial for effective information retrieval, as it efficiently manages large quantities of content while preserving the essential coherence and context of the information. By dividing the text into meaningful chunks, it is possible to get more accurate and contextually relevant responses from LLMs when processing user queries.

5.2 Phase 2: Embedding model

Embeddings are numerical vectorial representations of a piece of information, including text, documents, images, and audio. These embeddings capture the semantic meaning of the content being represented and are projected into a vector space where similar text is positioned close together, facilitating tasks like semantic search, clustering, and retrieval. The most used python library for calculating embeddings for sentences, paragraphs, and images in this domain is sentence-transformers, presented in [Reimers and Gurevych, 2019]. In this paper Sentence-BERT, a variation of BERT (Bidirectional Encoder Representations from Transformers) that is specifically designed for generating sentence embeddings, is presented. It uses siamese and triplet network structures to derive semantically meaningful sentence embeddings. The model is trained to maximize the similarity between semantically similar sentences and minimize the similarity between dissimilar sentences. This allows it to capture the contextual meaning of sentences and produce high-quality sentence embeddings for various natural language processing tasks.

To illustrate what an embedding is, let us consider the example text ”What is a cat?”. An embedding of the sentence could be represented in a vector space, for example, with a list of 384 numbers, like [0.38, ..., 0.91]. Since this list encapsulates the meaning of the text, it is possible to calculate the distance between different embeddings to determine how well the meanings of the two chunks of text match.

For each chunk of text, an embedding is made. When the user poses a query, the same embedding model is utilized to embed the user’s text.

For this chatbot, the paraphrase-multilingual-MiniLM-L12-v2 [2] is utilized, which maps text to a 384 dimension dense vector space. This model is suitable for this chatbot’s purpose, as it can handle both Portuguese and English text. Also, it is in the top 160 on the retrieval section of the Massive Text Embedding Benchmark (MTEB) Leaderboard[3] [Muennighoff et al., 2023].

Alternative options include all-MiniLM-L6-v2 [4], but is specifically designed for English, and text-embedding-ada-002, which relies on OpenAI and may raise concerns about data privacy and incur additional costs.

5.3 Phase 3: Vector Database

In this subsection, I will discuss the vector database component of this methodology, focusing on Faiss [Johnson et al., 2021], a library for efficient similarity search and clustering of dense vectors.

Vector database is a type of database that stores and indexes vector embeddings for fast retrieval and similarity search. Simply storing data as embeddings is not enough; vector databases create indexes on these embeddings to significantly accelerate the search process.

[2] https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2, accessed in 04/09/2024
[3] https://huggingface.co/spaces/mteb/leaderboard, accessed in 04/09/2024
[4] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2, accessed in 04/09/2024

In vector databases, a similarity metric is applied to find the vector that is most similar to a query. The search algorithms typically used in this type of databases are of the category of Approximate Nearest Neighbor (ANN) which optimize the search process by using techniques such as hashing, quantization, or graph-based search. Although they do not return the true k nearest neighbors, they are very efficient.

Faiss, developed primarily at FAIR, the fundamental AI research team of Meta, offers algorithms capable of searching through sets of vectors of various sizes, even those that possibly do not fit in RAM. Additionally, it contains supporting code for evaluation and parameter tuning.

Faiss is implemented in C++ with complete Python wrappers. Some of the most useful algorithms are implemented on the GPU.

By default, Faiss employs the Euclidean distance as the distance metric, which measures the straight line distance between two vectors in a vector space. This metric ranges from 0 to infinity, where 0 represents identical vectors, and larger values indicate increasing dissimilarity. Therefore, a lower score indicates a better match.

The chatbot retrieves the top 4 (k=4) passages from the database.

Other options for vector databases include ChromaDB and Weaviate, but for this chatbot's purpose, Faiss seems to be the most suitable choice.

5.4 Phase 4: Conversational Chain

A typical necessity for retrieval-augmented generation chains is the ability to handle follow-up questions. Follow-up questions may refer to previous conversations (e.g "What is the name of your cat?" followed by "What about its color?), making them unsuitable for direct retriever similarity search.

To support follow-up questions, it is necessary to combine the chat history (either provided or retrieved from memory) and new question into a standalone question prior to retrieval.

ConversationalRetrievalChain from Langchain is a chain that allows conversational interactions based on retrieved documents. It takes in chat history and new questions, and returns an answer to the new question. The algorithm consists of three parts:

1. Create a standalone question by combining the chat history and the new question. This ensures that relevant context is included in the question passed to the retrieval step. If only the new question was passed in, then relevant context may be lacking.

2. Pass the standalone question to the retriever to fetch relevant documents: an embedding is computed for the standalone question using the same embedding model algorithm as the one used to create the database embeddings; calculate distances between this vector and vectors stored in the database and then returns approximate nearest neighbors based on similarity ranking.

3. Pass the retrieved documents to a LLM along with the new question (default behavior) or the original question and chat history. The LLM generates a final response that takes into account the context of the conversation history, producing more accurate and context-aware answers.

Another necessity is keeping the chat history in memory. ConversationBufferMemory from Langchain refers to a feature that enables the storage of messages before processing them. Essentially, it acts as a temporary storage space for messages exchanged during a conversation. This buffer memory captures every interaction within the chat history, retaining a comprehensive record of the dialogue. It does so by extracting these stored messages into a variable. While advantageous in providing the maximum amount of information and maintaining simplicity and intuitiveness, there are drawbacks. These include potential delays in response times and increased computational costs due to the accumulation of more tokens. Additionally, the constraint of memory limits poses challenges, particularly in retaining lengthy conversations, as exceeding token limits can result in the loss of older messages.

5.5 Phase 5: Response generation

The decision to select a particular model was driven by the need for a compact yet powerful solution that could be accommodated within an affordable computer. It is crucial to strike a balance between size and accuracy. Therefore, a 7B model was chosen. Additionally, the model should be able to speak both Portuguese and English. Different models were tested, including Llama[Touvron et al., 2023] and Falcon[Almazrouei et al., 2023], both open source and free for commercial use. However, these models with just 7B parameters face limitations in performing inference in Portuguese. Exploring higher dimensions was considered, but the preference for a 7B model persisted due to its compatibility with affordable hardware components. After trying a few examples, I informally concluded that the model with the most promising results is Mistral-7B-Instruct-v0.2 [5].

Mistral 7B [Jiang et al., 2023] is a model released under the Apache 2.0 license. It utilizes innovative attention mechanisms such as grouped-query attention (GQA) and sliding window attention (SWA).

GQA boosts inference speed and minimizes memory requirement in the decoding process. SWA allows to manage longer sequences more efficiently with reduced computational costs. The incorporation of these attention mechanisms enhances Mistral 7B performance.

An Instruction Tuned LLM is a language model that undergoes an additional round of training on a narrowed dataset specifically designed to fine-tune its performance. This secondary training aims to enable the model to better respond to specific instructions provided in the prompt. In this case, the authors also provide a model fine-tuned to follow instructions, Mistral 7B–Instruct, that surpasses Llama 2 13B – chat model both on human and automated benchmarks.

It is also important to mention that the prompt given to the model by default is the following:

> És um assistente virtual que responde de forma clara e assertiva à pergunta do utilizador, de acordo com o documento dado pelo mesmo. Se o input do utilizador for em português, responde em português de Portugal.

5.6 Phase 6: Interface and deployment

To construct the interface of the application (Figure 5.2), I utilized Streamlit, an open-source python framework that enables the rapid development of web applications for Machine Learning and Data Science purposes.

In order to start using the application, the user just needs to upload PDFs documents on the left bar and click on Process. When the spinning circle disappears, it is possible to start the chat.

[5]https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2, accessed in 04/09/2024

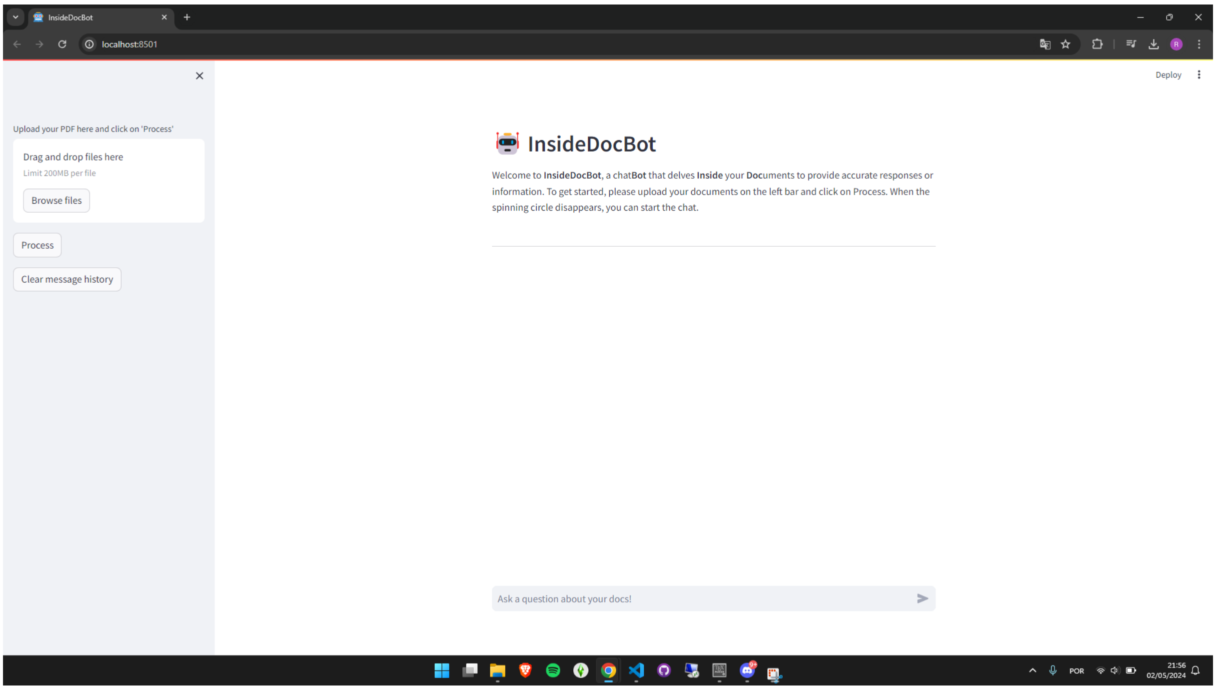


Figure 5.2: Chatbot interface

I accessed the Mistral-7B-Instruct-v0.2 model through the Hugging Face Hub [6], a dynamic platform housing an extensive array of resources including over 350,000 models, 75,000 datasets, and 150,000 demo apps (Spaces). This platform operates as a collaborative space for machine learning enthusiasts, offering open-source tools for exploration, experimentation, and technology development in the field of ML.

To sustain the model's functionality, I have integrated the Serverless Inference API [7] provided by Hugging Face. This API allows seamless testing and evaluation of more than 150,000 publicly available ML models, as well as private models, through straightforward HTTP requests. Leveraging the Hugging Face infrastructure ensures fast inference, enabling efficient utilization of machine learning resources.

LangSmith [8] is an integrated DevOps platform designed to streamline the development, collaboration, testing, deployment, and monitoring processes for LLM applications. With LangSmith, I have incorporated features to enhance user engagement and improve application performance. One notable feature is the ability to maintain a comprehensive chat history of users. Additionally, I have integrated a feedback system within the application, with the option of thumbs-up, thumbs down button and also emoji faces buttons. This feature allows me to gather insights into user experiences and identify areas for improvement.

[6]https://huggingface.co/docs/hub/index, accessed in 04/09/2024
[7]https://huggingface.co/docs/api-inference/index, accessed in 04/09/2024
[8]https://www.langchain.com/langsmith, accessed in 04/09/2024

Chapter 6

Results and Discussion

6.1 Retrieval evaluation

To assess the retrieval component, Pyserini [Lin et al., 2021], a Python toolkit tailored for reproducible information retrieval research, was used. Pyserini offers versatile functionality for both sparse and dense representations, aiming to facilitate effective, reproducible, and user-friendly first-stage retrieval within a multi-stage ranking framework. This toolkit, packaged as a standard Python module, includes essential components such as queries, relevance judgments, pre-built indexes, and evaluation scripts tailored for various widely used information retrieval collections.

Pyserini is engineered to seamlessly facilitate the entire research process aimed at enhancing ranking methodologies with contemporary neural techniques. Its comprehensive support extends across various retrieval paradigms, including sparse retrieval, exemplified by BM25 scoring using bag-of-words representations; dense retrieval, for example nearest-neighbor search on transformer-encoded representations; and hybrid retrieval, seamlessly integrating both approaches.

This comprehensive toolkit empowers researchers to effortlessly replicate experiments across multiple standard information retrieval test collections, facilitating robust and comparative analysis of various methodologies.

The decision to utilize Pyserini was primarily motivated by its compatibility with the chosen components of the study, namely, scripts to convert the MSMARCO passage dataset to a Pyserini-compatible format), FAISS for dense vector indexing, and compatibility with any model available through sentence transformers. This compatibility made it easier to get the results and to compare them across other model. Specifically, the approach of this study was conducted against BM25 [Robertson et al., 2009], the baseline method in this field.

The evaluation process of the retrieval encompassed several steps. Initially, the MSMARCO passage dataset was acquired, followed by the construction of an index utilizing the paraphrase-multilingual-MiniLM-L12-v2 embedding model. During indexing, the document collection, comprising 8,841,823 documents, was too extensive for the RAM of the computer, so only 44,44% of the dataset was processed to create a retrieval structure. Subsequently, retrieval operations were performed on the development subset, containing 6980 queries. In this retrieval phase, the system generated a ranked list of documents based on a given query, leveraging the constructed index. Finally, the obtained results underwent evaluation using the official MS MARCO evaluation script integrated into Pyserini, computing the

Mean Reciprocal Rank (MRR)@10, the designated metric for this dataset.

The resulting evaluation yielded the outcomes in Table 6.1.

Table 6.1: MSMARCO official metric results

	MRR@10
FAISS + paraphrase-multilingual-MiniLM-L12-v2	0.23880076863601185
Lucene + BM25 (baseline)	0.18741227770955546

The results indicate a performance improvement when utilizing FAISS in conjunction with the paraphrase-multilingual-MiniLM-L12-v2 embedding model compared to the baseline system employing Lucene with BM25 scoring. However, it is important to note that the FAISS + paraphrase-multilingual-MiniLM-L12-v2 configuration was evaluated on only 44.44% of the dataset, while the Lucene + BM25 baseline was evaluated on the entire dataset.

For the FAISS + paraphrase-multilingual-MiniLM-L12-v2 configuration, the Mean Reciprocal Rank (MRR)@10 achieved a value of approximately 0.239. This suggests that, on average, the relevant documents were positioned higher in the ranked list compared to the baseline. With a total of 6980 queries ranked, this result demonstrates the potential effectiveness of leveraging advanced techniques such as dense vector indexing and transformer-encoded representations for information retrieval tasks.

In contrast, the Lucene + BM25 baseline configuration yielded a lower MRR@10 value of around 0.187. Although this suggests comparatively less effective retrieval performance and that it may struggle to accurately identify and rank relevant documents within the top positions of the ranked list, it is crucial to recognize that this evaluation was conducted on the full dataset, comprising 8,841,823 documents. The larger dataset size may have introduced more noise and made the retrieval task more challenging, potentially explaining the lower MRR@10.

In addition to the mentioned metric, Pyserini offers the official TREC evaluation tool (Text REtrieval Conference, whose purpose was to support research within the information retrieval community by providing the infrastructure necessary for large-scale evaluation of text retrieval methodologies). This tool enables the computation of various metrics beyond the previously discussed one. In Table 6.2 are the results obtained through this evaluation framework.

Table 6.2: TREC results

	MAP	**Recall@1000**
FAISS + paraphrase-multilingual-MiniLM-L12-v2	0.2439	0.8145
Lucene + BM25 (baseline)	0.1957	0.8573

For the FAISS + paraphrase-multilingual-MiniLM-L12-v2 configuration, the Mean Average Precision (MAP) achieved a value of approximately 0.244, indicating higher average precision across all queries. This suggests that, within the subset of the dataset used (44.44% of the total), the relevant documents were positioned more effectively within the ranked list compared to the baseline. Additionally, the recall at 1000 documents (recall@1000) was approximately 0.814, indicating that a substantial portion of relevant documents were successfully retrieved within the top 1000 positions of the ranked list in this reduced dataset.

In contrast, the Lucene + BM25 baseline configuration, which was evaluated on the entire dataset, yielded a lower MAP value of around 0.196, suggesting comparatively lower precision in retrieving relevant documents across all queries. However, the recall@1000 value was slightly higher at approximately 0.857, indicating that a relatively higher proportion of relevant documents were retrieved within the top 1000.

Given the difference in dataset sizes, the higher MAP observed for the FAISS + paraphrase-multilingual-MiniLM-L12-v2 configuration might be influenced by the smaller subset of data, which could make retrieval tasks easier. Conversely, the full dataset used for the Lucene + BM25 baseline likely introduced more complexity, potentially explaining the lower MAP despite a slightly higher recall@1000.

6.2 Generation evaluation

6.2.1 Local accommodation dataset

To assess the local accommodation dataset, I queried each question to the selected model and subsequently computed the cosine scores between its responses and the ground truth.

I initiated the evaluation of the local accommodation dataset by examining the distribution of cosine similarity scores. The distribution in 6.1 reveals that cosine similarity scores span from a slightly negative value to just above 0.8. Predominantly, the scores are within the range of 0.15 to 0.50. This suggests that while some responses exhibit some alignment with the ground truth, others deviate significantly.

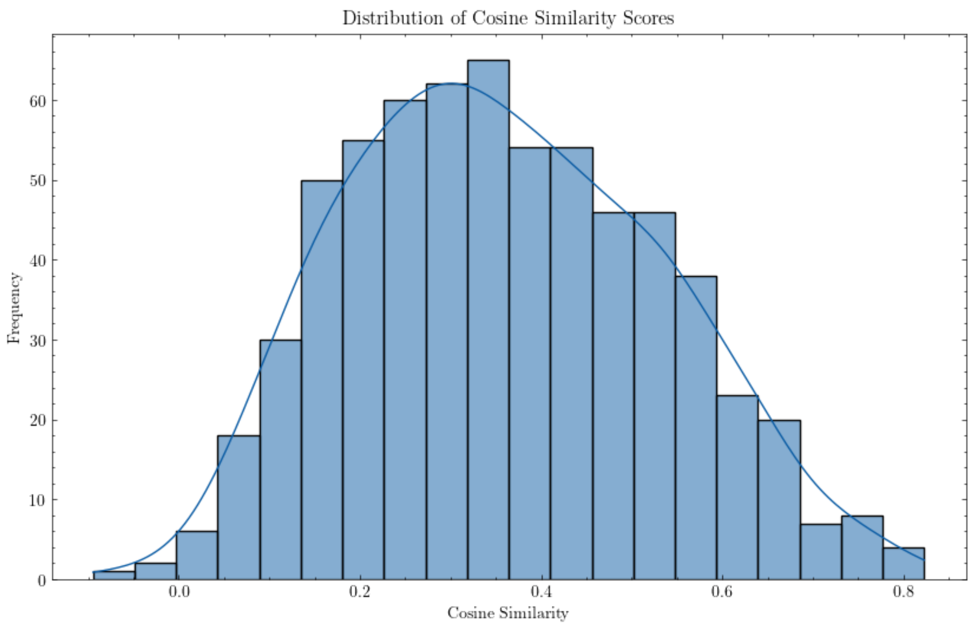


Figure 6.1: Distribution of Cosine Similarity Scores

I performed an additional analysis by calculating summary statistics. The mean score of approximately 0.36 suggests that, on average, the model's responses align moderately with the ground truth. With a median score of around 0.34, it implies that half of the cosine similarity scores fall below this value and half fall above. This value is slightly lower than the mean, suggesting a slightly skewed distribution towards lower similarity scores. And a standard deviation of approximately 0.17 suggests that there is moderate variability between the model's responses and the ground truth.

Let us analyze the top 3 scores of the cosine scores:

- **Score: 0.822868**
  - **Question:** O que a ANEPC recomenda em termos de prevenção?
  - **Answer:** A ANEPC recomenda que medidas de prevenção e instruções de segurança sejam afixadas nos estabelecimentos.
  - **Contexts:** ['10.5 Suspensão de autorizações' '20. Reporte de informação de dormidas', '19. Acesso aos estabelecimentos', 'Sobre as fichas técnicas, a ASAE presta o seguinte esclarecimento:', 'Segundo indicações da Autoridade Nacional de Emergência e Proteção Civil (ANEPC), que é a entidade competente para assegurar o cumprimento do regime de segurança contra incêndios em edifícios (SCIE): O produto extintor a utilizar depende do "tipo de fogo", podendo o Pó Químico ABC (6kg) ou a Água Aditivada (5 Kg) ser uma boa opção. A manta deve ser dimensionada de acordo com o dispositivo de queima sobre o qual se pretenda atuar. A ANEPC recomenda ainda que, em termos de prevenção, é importante que sejam afixadas nos estabelecimentos medidas de prevenção e instruções de segurança.']
  - **Ground truths:** ['Afixar medidas de prevenção e instruções de segurança.']
- **Score: 0.79913**
  - **Question:** O que deve ser disponibilizado ao mesmo tempo que o formato eletrónico?
  - **Answer:** Ao mesmo tempo que o formato eletrónico, o formato físico do livro de reclamações deve ser disponibilizado.
  - **Contexts:** ['16.3 A existência e disponibilização do formato eletrónico não dispensa da necessidade de disponibilização, em simultâneo, do formato físico do livro de reclamações.', '16.2 Livro de reclamações em formato eletrónico', '1) Disponibilizar no sítio da internet, em local bem visível e de forma destacada, o acesso à plataforma digital que disponibiliza o formato eletrónico do livro de reclamações.', 'De acordo com este regime, o titular da exploração do estabelecimento está obrigado a possuir e disponibilizar o livro de reclamações nos formatos físico e eletrónico.', '14.2 Nos ≪estabelecimentos de hospedagem≫, ≪apartamentos≫ e ≪quartos≫ é obrigatória a afixação junto à entrada do estabelecimento de uma placa identificativa.']
  - **Ground truths:** ['O formato físico do livro de reclamações.']
- **Score: 0.791418**
  - **Question:** Quais são os tipos de estabelecimentos que não podem ser explorados como alojamento local?
  - **Answer:** The establishments that cannot be exploited as local lodging are those that fall under any category of tourist establishments, such as hotels, tourist apartments, tourist villages or habitational or rural tourism facilities.

  - **Contexts:** ['Limites à capacidade dos estabelecimentos de alojamento local', 'Assim, para a delimitação do conceito de estabelecimento de alojamento local importa, desde logo, recortá-lo da figura dos empreendimentos turísticos, já que só poderão ser explorados como alojamento local os estabelecimentos que, pelas suas características, não possam ser enquadrados em nenhuma das tipologias de empreendimentos turísticos, como sejam os estabelecimentos hoteleiros, apartamentos turísticos, aldeamentos turísticos ou empreendimentos de turismo de habitação ou de turismo no espaço rural.', '13.1 Os estabelecimentos devem identificar-se como estabelecimentos de alojamento local, não podendo, em caso algum, utilizar a qualificação de empreendimento turístico, ou de qualquer tipologia de empreendimento turístico, nem qualquer sistema de classificação.', '2.3 A exploração de estabelecimento de alojamento local depende de prévio registo destes estabelecimentos através de uma comunicação prévia com prazo realizada no Balcão Único Eletrónico, não sendo permitida a oferta, disponibilização, publicidade e intermediação de estabelecimentos de alojamento local não registados ou com registos desatualizados.', 'Nestas áreas podem ser impostos limites relativamente ao número de estabelecimentos, carecendo a instalação de novos estabelecimentos de alojamento local nesse território de autorização expressa da câmara que, em caso de deferimento, promove o respetivo registo.']
  - **Ground truths:** ['Estabelecimentos hoteleiros, apartamentos turísticos, aldeamentos turísticos, entre outros.']

Each of the top responses directly addresses the questions posed, indicating that the model has understood the queries accurately and provided appropriate answers.

The responses are supported by relevant contexts extracted from the provided data. This suggests that the model is capable of contextual comprehension, drawing information from surrounding text to generate meaningful responses. The ability to incorporate contextual information enhances the relevance and accuracy of the answers. The top responses closely match the ground truths, which are the correct answers or desired responses. This alignment indicates that the model's outputs are consistent with the expected information based on the provided data.

Despite occasional instances where the model provides answers in English, rather than Portuguese, the correctness of the responses remains intact.

These results highlight a significant advancement facilitated by RAG, as the model can now discern the content of documents without the need for extensive computational resources typically associated with training LLMs. This signifies a cost-effective approach to leveraging advanced natural language processing capabilities for tasks such as question-answering.

Let us now observe the bottom 3 results:

- **Score: -0.09466**

  - **Question:** O que deve ser disponibilizado gratuitamente ao utente?
  - **Answer:** Não sei. The context provided does not contain any information about what should be made available for free to the user.

- **Contexts:** ['10.5 Suspensão de autorizações' '19. Acesso aos estabelecimentos', '18.Seguro obrigatório' '15. Período de funcionamento', '20. Reporte de informação de dormidas']
- **Ground truths:** ['O livro de reclamações.']

• **Score: -0.046608**

- **Question:** O que o estabelecimento deve afixar em local bem visível?
- **Answer:** Based on the context, there is no direct information available about what an establishment should display in a visible location. The context discusses registration and modalities of local accommodation establishments, but it does not mention any specific requirements for what should be displayed in a visible location. Therefore, I cannot answer the question definitively based on the context. My suggestion would be to consult the relevant authorities or regulations for specific guidelines on this matter as it can vary depending on the location and legal framework.
- **Contexts:** ['Limites à capacidade dos estabelecimentos de alojamento local', '2.3 A exploração de estabelecimento de alojamento local depende de prévio registo destes estabelecimentos através de uma comunicação prévia com prazo realizada no Balcão Único Eletrónico, não sendo permitida a oferta, disponibilização, publicidade e intermediação de estabelecimentos de alojamento local não registados ou com registos desatualizados.', 'e) Modalidade do estabelecimento de alojamento local;', 'Presume-se existir exploração e intermediação de estabelecimento de alojamento local quando um imóvel ou fração deste: a) Seja publicitado, disponibilizado ou objeto de intermediação, por qualquer forma, entidade ou meio, nomeadamente em agências de viagens e turismo ou sites da Internet, como alojamento para turistas ou como alojamento temporário; ou b) Estando mobilado e equipado, neste sejam oferecidos ao público em geral, além de dormida, serviços complementares ao alojamento, nomeadamente limpeza, por períodos inferiores a 30 dias.', '5.4. Nos estabelecimentos de hospedagem podem existir zonas comuns de acolhimento ou de receção e de estar e/ou lazer, destinadas aos utentes, podendo estas funções coexistir no mesmo espaço.']
- **Ground truths:** ['Letreiro informando sobre o livro de reclamações.']

• **Score: -0.021082**

- **Question:** Que informações devem constar no livro de informações?
- **Answer:** <bot>: The context provided does not specifically state what information must be included in the "livro de informações" or the "book of information" mentioned in the text. Therefore, my answer is: "Não sei" based on the given context. However, it appears from the context that this book might be related to a registration process, and there are certain required communications with prior notice that need to be documented. But, the context does not indicate definitively that this "livro de informações" refers to the same document as the one being asked about in the question. So, my answer is uncertain. Please consult the entire document or additional context if possible for a more definitive answer.

- **Contexts:** ['17. Livro de informações', '4) Fornecer todos os elementos necessários ao correto preenchimento dos campos que, no livro de reclamações, se referem à sua identificação, devendo ainda confirmar que o utente os preencheu corretamente;', 'Sobre as fichas técnicas, a ASAE presta o seguinte esclarecimento:', 'As principais alterações introduzidas pela Lei n.º 62/2018, de 22 de agosto referem-se à forma do procedimento de registo, que passa a ser o da comunicação prévia com prazo; à possibilidade de as câmaras municipais estabelecerem limites à atividade de exploração dos estabelecimentos em determinadas áreas; ao alargamento das situações em que as câmaras podem cancelar os registos; à necessidade de autorização do condomínio para a instalação de ≪hostel≫ e ainda à previsão de novos requisitos ou regras de exploração e funcionamento, como sejam as relativas à capacidade máxima dos alojamentos, a obrigatoriedade de um livro de informações, de afixação de placas identificativas e de celebração de um seguro de responsabilidade civil.', '9.2 Da comunicação prévia com prazo devem obrigatoriamente constar as seguintes informações:']
- **Ground truths:** ['Recolha e seleção de resíduos urbanos, entre outras.']

The contexts provided for these results lack specific information relevant to the questions posed. This limitation hinders the model's ability to generate accurate responses, as the system relies heavily on the context to answer the question.

In response to the questions posed, the model exhibits uncertainty by either providing vague answers or expressing a lack of knowledge ("Não sei"). This limitation was explicitly outlined in the prompt, which stated:

Responde à questão tendo por base apenas o seguinte contexto. Se não conseguires responder à questão com o contexto, responde "Não sei".

Moreover, the uncertainty stems from the insufficiency of relevant information in the provided contexts, making it challenging for the model to generate definitive responses.

To compensate for the lack of specific information in the contexts, the model suggests consulting relevant authorities or regulations for definitive answers. This indicates the model's awareness of its limitations and it attempts to provide guidance on where to find accurate information, which is better than just answering "I do not know".

The generated responses do not align with the ground truths, which indicates the model's inability to generate accurate outputs consistent with the expected information based on the provided data, leading to low cosine scores.

The bottom results highlight the importance of providing comprehensive and relevant contextual information to facilitate accurate responses from the model. For example, the first result shows that in some situations, there really is not the need to include the document sections and subsections titles. Enhancing the quality and specificity of the contexts can help mitigate uncertainty and improve the accuracy of the model's generated answers.

6.2.2 MS MARCO - Question Answering dataset

In order to evaluate the generation component of the system, the MS MARCO Question Answering dataset was used. This process started with acquiring the MS MARCO dataset from Hugging Face. Prior to evaluation, the dataset underwent preprocessing procedures aimed at enhancing its suitability for evaluation. Following this, I computed embeddings and constructed a vector database, loaded the model, established the chain, and executed evaluations. I checked performance using Rouge and BLEU scores.

Let us observe the distribution of the Rouge and BLEU scores in figures 6.2 and 6.3, respectively.

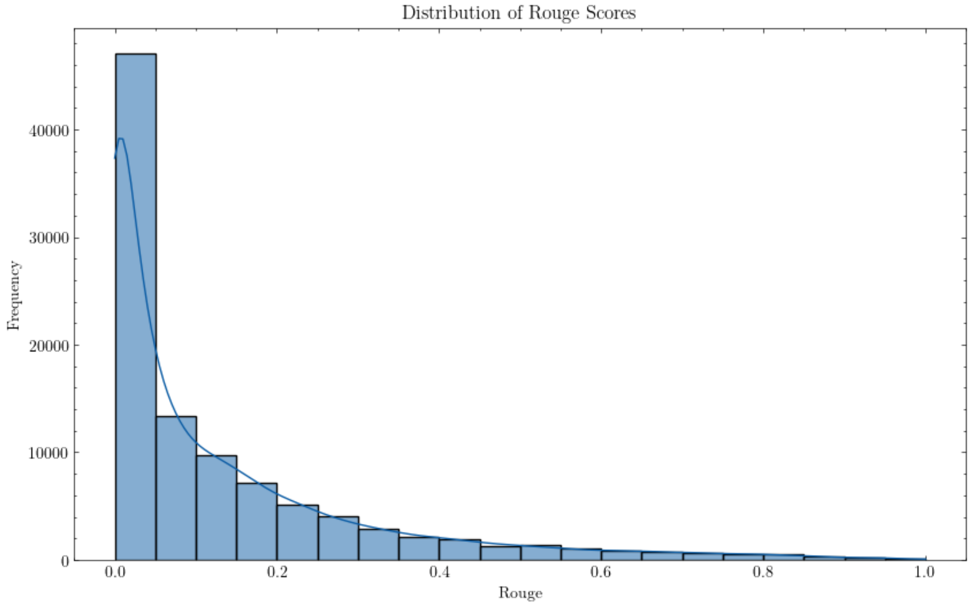


Figure 6.2: Distribution of Rouge Scores

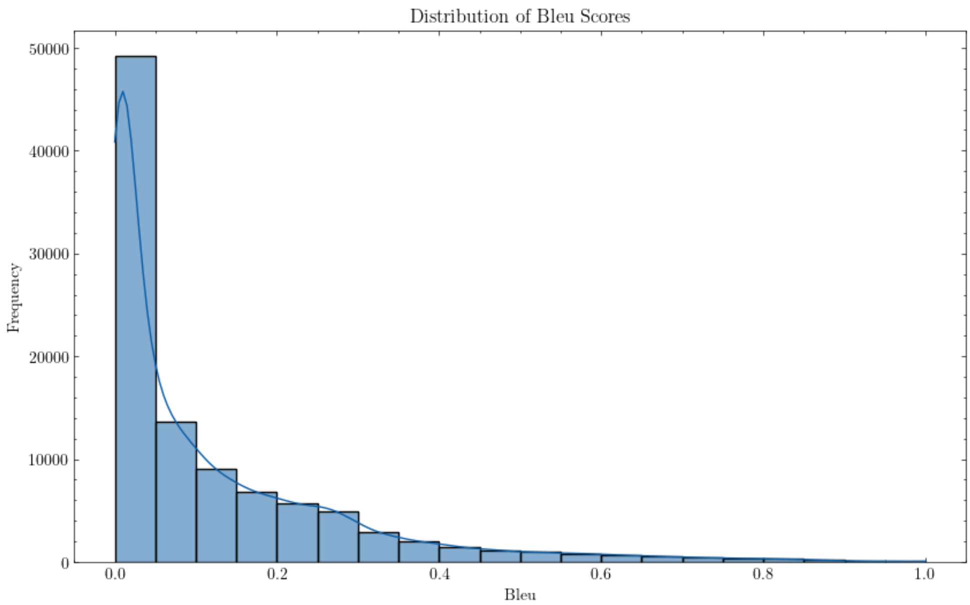


Figure 6.3: Distribution of Bleu Scores

Upon observing the Rouge and BLEU scores, it is evident that the outcomes are considerably underwhelming. Over 4,45% of the Rouge scores and nearly 5% of the BLEU scores are precariously close to zero. This indicates a significant gap between the generated responses and the expected answers within the MS MARCO dataset.

However, it is crucial to recognize that Rouge and BLEU metrics, while commonly used, may not be the most suitable for assessing text generation quality comprehensively. These metrics primarily focus on surface-level linguistic features such as n-gram overlap and translations and may not capture semantic coherence or relevance effectively. In the context of natural language generation, where nuances and context play pivotal roles, relying solely on Rouge and BLEU scores may provide an incomplete picture of model performance.

6.3 Chatbot demonstration

The chat presented in Figures 6.4, 6.5 and 6.6 showcases a typical interaction with the chatbot. In this instance, the selected document is ”Guia técnico - Alojamento local: regime jurídico”. This dialogue with the chatbot exemplifies a well-structured and helpful process for answering questions about local accommodation regulations. The interaction begins with a clear question about whether it is mandatory to provide a complaints book in local accommodation establishments. Citing the document, the solution is found within this excerpt:

> Todos os estabelecimentos de alojamento local devem dispor de livro de reclamações nos termos e condições estabelecidos no Decreto-Lei n.º 156/2005, de 15 de setembro, alterado pelos Decreto-Lei n.º 371/2007, de 6 de novembro, Decreto-lei n.º 118/2009, de 19 de maio, Decreto-Lei n.º 317/2009, de 30 de outubro, Decreto-Lei n.º 242/2012, de 7 de novembro e Decreto-Lei n.º 74/2017, de 21 de junho.

The chatbot promptly responds, confirming that it is indeed mandatory to have a complaints book available, both in physical and electronic formats. This information is correct and aligns with Portuguese law, which requires all local accommodations, to make a complaints book available to their customers.

A key point to note is that the core truth of the chatbot's response is found particularly in ”Source 3” which mentions the legal obligation for all local accommodation establishments to have a complaints book. This shows that the response is well-grounded in legislation. The answer varies slightly from the passage in certain words. This means that the system is able to retrieve the correct passage and then generate an answer to this specific question.

Another important aspect is the additional guidance provided by the chatbot, suggesting that more information can be obtained from the Direção-Geral do Consumidor. This kind of direction is extremely valuable as it guides the user to an official entity that can provide further details and assistance on the matter.

Following the chat exchange, users are provided with sources, a feature integral to the chatbot's reliability and transparency. This functionality has been the subject of extensive study for several years, ensuring that it reliably identifies and presents relevant passages pertinent to the user's query. Additionally, the provision of sources serves as a means for users to independently verify the accuracy of the chatbot's responses.

Moreover, users have the opportunity to offer feedback, enabling ongoing refinement and enhancement of the chatbot's performance.

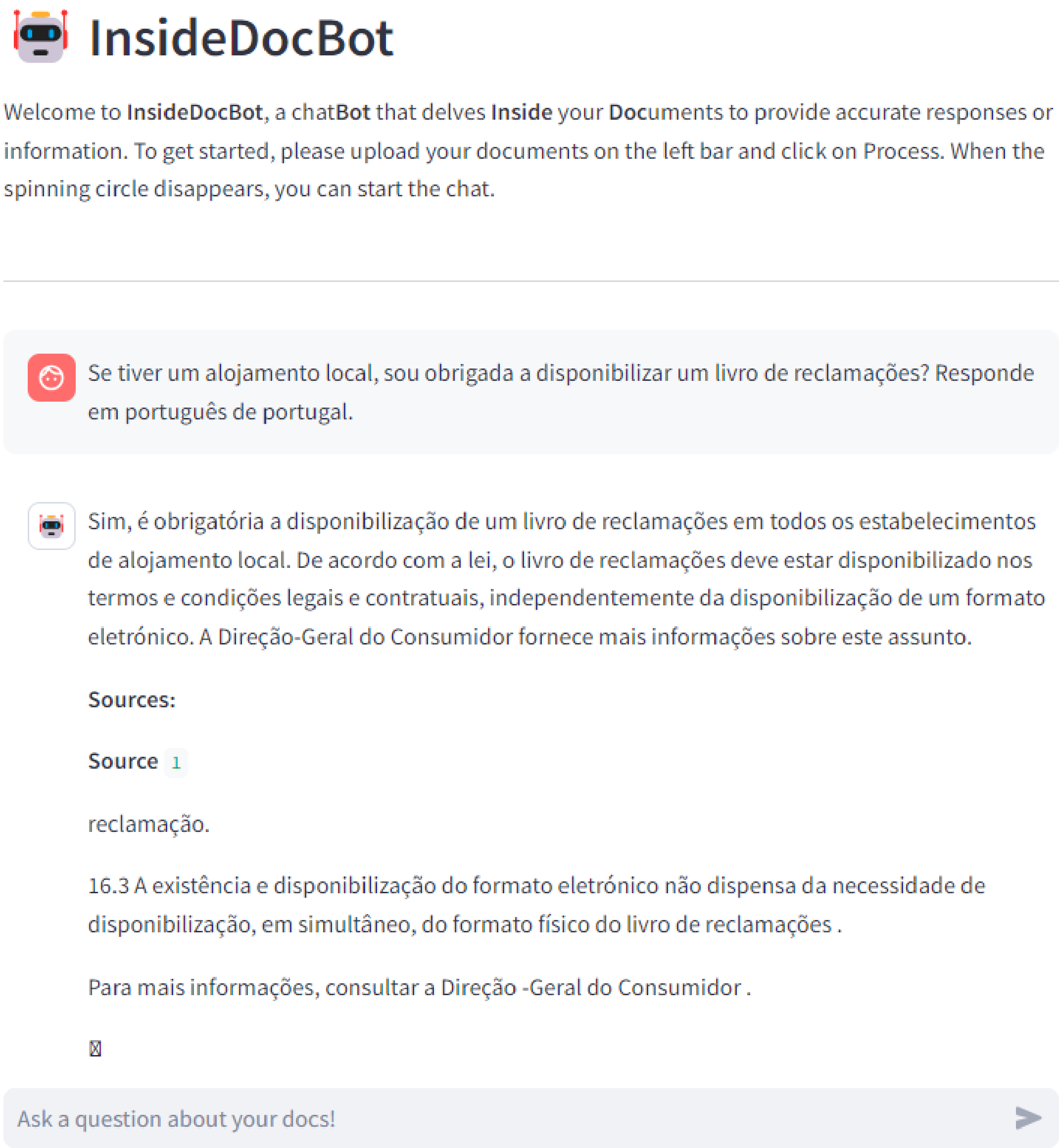


Figure 6.4: Chatbot demonstration (Part 1)

28

26

Em todos os estabelecimentos de alojamento local é obrigatória a existência de um livro de informações a disponibilizar aos hóspedes em português, inglês e , pelo menos, mais duas línguas

Source 2

i m e nt os d e A L re gi st a d

O Turismo de Portugal, I. P., disponibiliza no seu sítio na Internet informação sobre os estabelecimentos de alojamento local registados

Estabelecimentos de Alojamento Local registados: Registo Nacional de Estabelecimentos de Alojamento Local .

Localização georreferenciada dos estabelec imentos de Alojamento Local: SIGTUR .

- info :

Estabelecimentos de alojamento local na Ma deira .

Source 3

funcionamento, sem prejuízo de disposição legal ou contratual.

15.2 Os «estabelecimentos de hospedagem », quando não estejam abertos todos os dias do ano, devem publi citar o seu período de funcionamento.

☒ 15. Período de funcionamento
26

Todos os estabelecimentos de alojamento local devem dispor de livro de reclamações nos termos e

Ask a question about your docs!

Figure 6.5: Chatbot demonstration (Part 2)

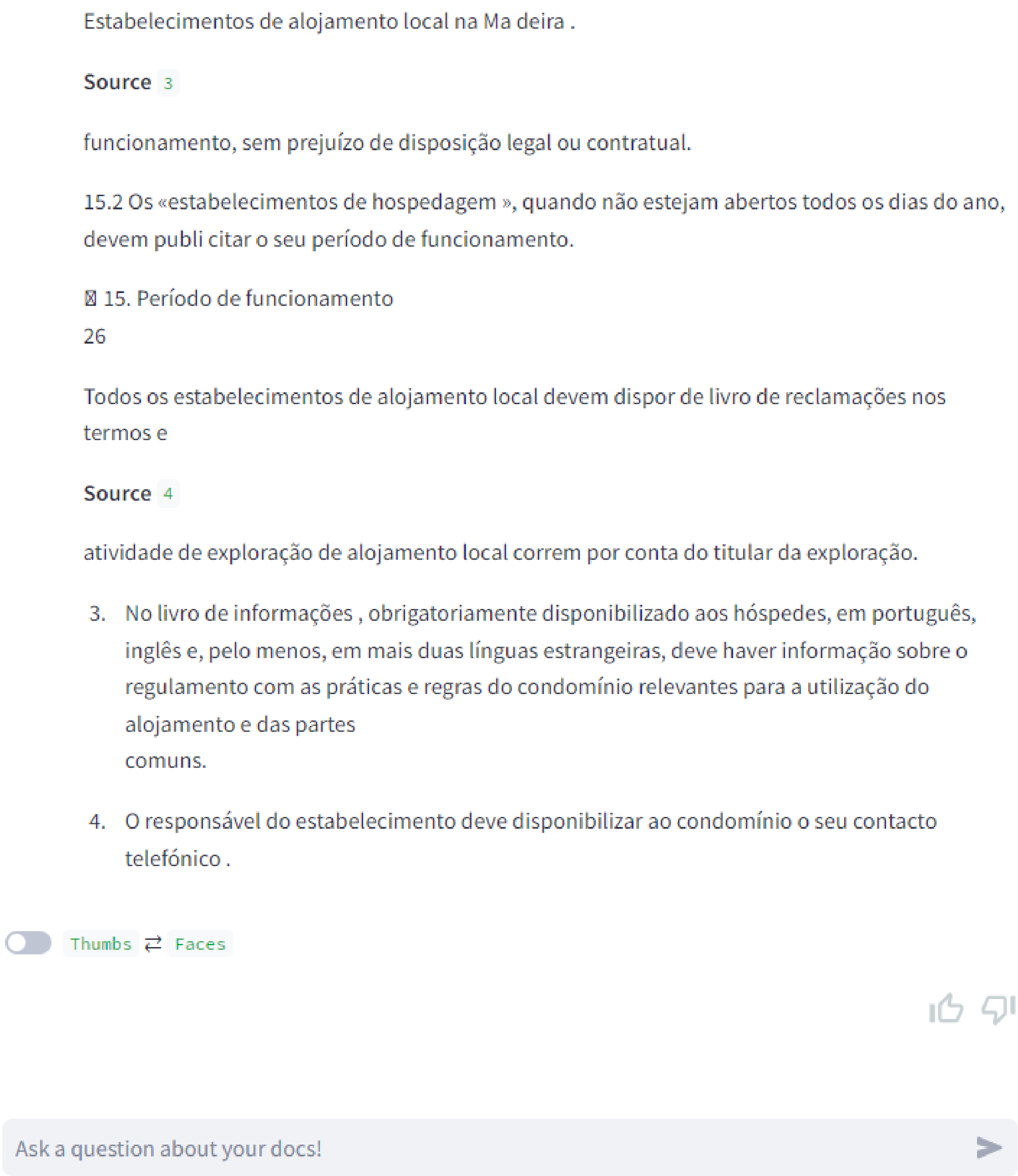


Figure 6.6: Chatbot demonstration (Part 3)

Chapter 7

Conclusion

In this concluding section, I wrap up my master thesis journey exploring conversational AI. It has been a deep dive into tackling the challenge of hallucination in language models and crafting a chatbot that is both reliable and trustworthy. A review of the completed work, resulting contributions, and the upcoming work is presented.

7.1 Final Notes

In this master thesis, I address a challenge within the realm of conversational AI: the persistence of hallucination in LLMs. Hallucination, the generation of content not present or implied in the input data, poses a significant obstacle to the reliability and trustworthiness of LLM-powered chatbots.

My primary objective was clear: to create a chatbot capable of providing accurate responses to user queries based on a provided document, all while circumventing the pitfalls of hallucination. To achieve this, I delved into the realm of Retrieval-Augmented Generation (RAG), a paradigm that joins information retrieval with generation techniques. By adopting this approach, I aimed to imbue the chatbot with a sense of context-awareness and the ability to retrieve pertinent information from external sources, thereby fortifying its capacity to deliver trustworthy and timely responses.

My journey was marked by an exhaustive exploration of existing research, empirical studies, and real-world applications. I meticulously dissected the nuances of hallucination in LLMs, delving into its intrinsic and extrinsic manifestations.

Leveraging the RAG approach, I engineered a chatbot that seamlessly integrates information retrieval and answer generation components. This fusion of capabilities enables the chatbot to navigate complex queries, retrieving relevant content from the provided document to furnish users with accurate and reliable responses. By prioritizing context-awareness, the chatbot's efficacy and trustworthiness augments.

To achieve my goals, I proposed and implemented a comprehensive methodology comprising several distinct phases. I utilized open-source technologies to prioritize data privacy and security, and I equipped the chatbot with multilingual capabilities, catering to users in both English and Portuguese languages. This approach involved information retrieval and answer generation components, ensuring that users receive accurate and pertinent responses.

Through practical experimentation and evaluation using real-world datasets, including a custom developed Portuguese dataset and the MS MARCO Question Answering dataset, I meticulously assessed the effectiveness of my methodology. I conducted retrieval and generation evaluations, employing metrics such as Rouge and BLEU scores to gauge the quality of generated responses. While my results revealed certain limitations and challenges, they also provided valuable insights into areas for improvement.

My chatbot implementation showcased promising capabilities, demonstrating its potential utility in various domains requiring document interaction and question answering.

7.2 Future Work

While I have reached a conclusion, I recognize that there are limitations and ample opportunities for enhancement. Looking ahead, my plan is to persist with the current process and focus on refining it further.

Expanding the chatbot's multilingual capabilities to include additional languages would significantly enhance its usability and accessibility globally. This involves creating more diverse datasets in various languages and optimizing the model to perform effectively across different linguistic contexts. To achieve this, it is essential to collect and integrate datasets from multiple languages. Fine-tuning the model on these multilingual datasets can improve its understanding and generation capabilities in different languages. Additionally, developing robust evaluation metrics tailored to multilingual performance will help in assessing and benchmarking the chatbot's capabilities across various languages.

The chatbot's performance can be further evaluated and optimized for specific domains such as education and human resources. The model needs to understand and process domain-specific terminologies and contexts to improve its reliability in these sensitive areas. In order to do this, domain-specific datasets need to be developed. Collaborating with domain experts to fine-tune the chatbot's responses and ensure accuracy and relevance, while ensuring that the chatbot's responses comply with regulatory standards and ethical guidelines, is essential.

Refining the retrieval component of the RAG is another promising area for future research. Implementing more sophisticated retrieval algorithms or integrating recent advancements in semantic search can enhance the accuracy and relevance of the information retrieved. According to [Setty et al., 2024] improving retrieval quality requires advanced techniques, such as recursive chunking, which uses indicators like punctuation for dynamic chunking. Element-based chunking, considering the document structure like headings and tables, is also effective for financial reports. Query expansion techniques like Hypothetical Document Embeddings (HyDE) improve retrieval by generating a theoretical document in response to a query, enhancing similarity searches. Metadata annotations and indexing enhance retrieval by including key data points missed by standard algorithms. Re-ranking algorithms prioritize relevance over similarity, improving chunk selection. Fine-tuning embedding algorithms with domain-specific knowledge enhances retrieval, requiring datasets with queries, text corpus, and relevant documents.

Implementing a system for real-time learning and adaptation can enhance the chatbot's responsiveness to new information. This involves developing mechanisms for the chatbot to update its knowledge base dynamically as it interacts with users. Integrating real-time data sources such as news feeds, databases, and APIs will keep the chatbot's knowledge base current. Collecting and analyzing feedback from users in real-time can identify areas for improvement and adapt the chatbot's responses accordingly. Developing automated pipelines for data ingestion, processing, and model updating can facilitate real-time learning.

Introducing robust user feedback systems can help in identifying and correcting errors in real-time. Collecting and analyzing user feedback can provide valuable insights into areas where the chatbot's performance can be improved and help in improving its responses. This can be done with Langsmith. Also, conducting regular user surveys to gather detailed feedback on the chatbot's performance, usability, and areas for enhancement are also crucial. Systematically analyzing errors and incorrect responses to understand their root causes will help in developing strategies for mitigating hallucinations.

Future work should also focus on addressing ethical issues in the chatbot's responses. Auditing the training data for biases and implementing algorithms that can detect and correct biased outputs is essential to ensure fair and unbiased interactions. Developing algorithms to detect biases in the chatbot's responses and in the training data, and ensuring that training datasets are representative of diverse populations and viewpoints, will minimize biases. Conducting regular ethical audits to assess the chatbot's performance from an ethical standpoint and ensure compliance with industry standards and guidelines is also necessary.

This research can explore methods to improve the scalability of the chatbot for deployment in large scale applications. This includes optimizing the computational efficiency of the model, reducing latency, and ensuring robust performance under heavy user load. Implementing model compression techniques such as quantization will reduce the computational footprint of the chatbot. Leveraging distributed computing frameworks to scale the chatbot's operations and handle large volumes of user queries efficiently will also be crucial. Establishing performance benchmarks and conducting tests will ensure the chatbot performs reliably under various conditions.

A critical area for future work is the development of robust evaluation metrics specifically designed to detect hallucinations in LLMs. In my initial efforts to address this issue, I explored the use of the RAGAS (Retrieval Augmented Generation Assessment) framework [Es et al., 2023], which aims to provide an evaluation of RAG pipelines by assessing various dimensions such as the fidelity of the generated answers. However, the results from employing RAGAS were inconclusive. Most of the cases resulted in outputs of 'Nan' while in other instances, the system returned the entire dataset with maximum or minimum values. Consequently, I decided not to utilize RAGAS for evaluating hallucinations in my chatbot.

To date, I have not identified any reliable metrics that consistently and accurately evaluate hallucinations in LLM outputs. This gap highlights the need for further research and development in this area. Developing such metrics will be crucial for advancing the reliability and trustworthiness of LLM-based systems, ensuring that they provide accurate and fact-based responses across various applications. Future work will focus on creating and validating new evaluation methodologies that can effectively detect and quantify hallucinations.

By pursuing these future directions, the chatbot can be refined and enhanced to provide more accurate, reliable, and contextually aware responses, ultimately leading to more effective and trustworthy conversational AI systems. Continued research and development in these areas hold the potential for significant advancements, not only for chatbots but also for the broader field of artificial intelligence and human-computer interaction.

On a final note, my internship experience has been positive. It has been a period of significant growth, both professionally and personally. This year has bridged the gap between academia and the real world, highlighting areas for growth like time management while also enhancing my research abilities and providing insights into artificial intelligence. This dissertation not only benefits my academic journey but also enriches my personal life.

Bibliography

[Adamopoulou and Moussiades, 2020] Adamopoulou, E. and Moussiades, L. (2020). Chatbots: History, technology, and applications. *Machine Learning with Applications*, 2:100006.

[Almazrouei et al., 2023] Almazrouei, E., Alobeidli, H., Alshamsi, A., Cappelli, A., Cojocaru, R., Debbah, M., Étienne Goffinet, Hesslow, D., Launay, J., Malartic, Q., Mazzotta, D., Noune, B., Pannier, B., and Penedo, G. (2023). The falcon series of open language models.

[Arsenijevic and Jovic, 2019] Arsenijevic, U. and Jovic, M. (2019). Artificial intelligence marketing: Chatbots. In *2019 International Conference on Artificial Intelligence: Applications and Innovations (IC-AIAI)*, pages 19–193.

[aza et al., 2018] aza, muha, zura, and Ahmad, N. A. (2018). Review of chatbots design techniques. *International Journal of Computer Applications*, 181:7–10.

[Bi et al., 2019] Bi, B., Wu, C., Yan, M., Wang, W., Xia, J., and Li, C. (2019). Incorporating external knowledge into machine reading for generative question answering. *arXiv preprint arXiv:1909.02745*.

[Boné et al., 2020] Boné, J., Ferreira, J. C., Ribeiro, R., and Cadete, G. (2020). Disbot: A portuguese disaster support dynamic knowledge chatbot. *Applied Sciences*, 10(24).

[Caldarini et al., 2022] Caldarini, G., Jaf, S., and McGarry, K. (2022). A literature survey of recent advances in chatbots. *Information*, 13(1):41.

[Candel et al., 2023] Candel, A., McKinney, J., Singer, P., Pfeiffer, P., Jeblick, M., Prabhu, P., Gambera, J., Landry, M., Bansal, S., Chesler, R., Lee, C. M., Conde, M. V., Stetsenko, P., Grellier, O., and Ambati, S. (2023). h2ogpt: Democratizing large language models.

[Cao et al., 2020] Cao, M., Dong, Y., Wu, J., and Cheung, J. C. K. (2020). Factual error correction for abstractive summarization models. *arXiv preprint arXiv:2010.08712*.

[Chen et al., 2021] Chen, S., Zhang, F., Sone, K., and Roth, D. (2021). Improving faithfulness in abstractive summarization with contrast candidate generation and selection. *arXiv preprint arXiv:2104.09061*.

[Cheng et al., 2006] Cheng, W., Greaves, C., and Warren, M. (2006). From n-gram to skipgram to concgram. *International journal of corpus linguistics*, 11(4):411–433.

[Cheng et al., 2023] Cheng, X., Luo, D., Chen, X., Liu, L., Zhao, D., and Yan, R. (2023). Lift yourself up: Retrieval-augmented text generation with self memory.

[Chung et al., 2014] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling.

[Colby et al., 1971] Colby, K. M., Weber, S., and Hilf, F. D. (1971). Artificial paranoia. *Artificial Intelligence*, 2(1):1–25.

[Dai et al., 2022] Dai, Z., Zhao, V. Y., Ma, J., Luan, Y., Ni, J., Lu, J., Bakalov, A., Guu, K., Hall, K. B., and Chang, M.-W. (2022). Promptagator: Few-shot dense retrieval from 8 examples.

[Dey and Salem, 2017] Dey, R. and Salem, F. M. (2017). Gate-variants of gated recurrent unit (gru) neural networks. In *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1597–1600.

[Dhuliawala et al., 2023] Dhuliawala, S., Komeili, M., Xu, J., Raileanu, R., Li, X., Celikyilmaz, A., and Weston, J. (2023). Chain-of-verification reduces hallucination in large language models. *arXiv preprint arXiv:2309.11495*.

[Dziri et al., 2021] Dziri, N., Madotto, A., Zaïane, O., and Bose, A. J. (2021). Neural path hunter: Reducing hallucination in dialogue systems via path grounding. *arXiv preprint arXiv:2104.08455*.

[Es et al., 2023] Es, S., James, J., Espinosa-Anke, L., and Schockaert, S. (2023). Ragas: Automated evaluation of retrieval augmented generation. *arXiv preprint arXiv:2309.15217*.

[Farea et al., 2022] Farea, A., Yang, Z., Duong, K., Perera, N., and Emmert-Streib, F. (2022). Evaluation of question answering systems: Complexity of judging a natural language. *arXiv preprint arXiv:2209.12617*.

[Følstad and Skjuve, 2019] Følstad, A. and Skjuve, M. (2019). Chatbots for customer service: User experience and motivation. In *Proceedings of the 1st International Conference on Conversational User Interfaces*, CUI '19, New York, NY, USA. Association for Computing Machinery.

[Gao et al., 2023] Gao, L., Ma, X., Lin, J., and Callan, J. (2023). Precise zero-shot dense retrieval without relevance labels. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada. Association for Computational Linguistics.

[Gao et al., 2024] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Guo, Q., Wang, M., and Wang, H. (2024). Retrieval-augmented generation for large language models: A survey.

[Gardent et al., 2017] Gardent, C., Shimorina, A., Narayan, S., and Perez-Beltrachini, L. (2017). Creating training corpora for nlg micro-planning. In *55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*, pages 179–188. Association for Computational Linguistics (ACL).

[Ghojogh and Ghodsi, 2023] Ghojogh, B. and Ghodsi, A. (2023). Recurrent neural networks and long short-term memory networks: Tutorial and survey.

[Graves and Graves, 2012] Graves, A. and Graves, A. (2012). Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45.

[Hochreiter, 1998] Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116.

[Huang et al., 2020] Huang, L., Wu, L., and Wang, L. (2020). Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward. *arXiv preprint arXiv:2005.01159*.

[Huang et al., 2021] Huang, Y.-C., Feng, X., Feng, X., and Qin, B. (2021). The factual inconsistency problem in abstractive text summarization: A survey. *ArXiv*, abs/2104.14839.

[Hussain et al., 2019] Hussain, S., Ameri Sianaki, O., and Ababneh, N. (2019). A survey on conversational agents/chatbots classification and design techniques. In *Web, Artificial Intelligence and Network Applications: Proceedings of the Workshops of the 33rd International Conference on Advanced Information Networking and Applications (WAINA-2019) 33*, pages 946–956. Springer.

[Izacard and Grave, 2022] Izacard, G. and Grave, E. (2022). Distilling knowledge from reader to retriever for question answering.

[Jha et al., 2023] Jha, S., Jha, S. K., Lincoln, P., Bastian, N. D., Velasquez, A., and Neema, S. (2023). Dehallucinating large language models using formal methods guided iterative prompting. In *2023 IEEE International Conference on Assured Autonomy (ICAA)*, pages 149–152. IEEE.

[Ji et al., 2023] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

[Jiang et al., 2023] Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. (2023). Mistral 7b.

[Johnson et al., 2021] Johnson, J., Douze, M., and Jégou, H. (2021). Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

[Kuhail et al., 2023] Kuhail, M. A., Alturki, N., Alramlawi, S., and Alhejori, K. (2023). Interacting with educational chatbots: A systematic review. *Education and Information Technologies*, 28(1):973–1018.

[Lamport, 1994] Lamport, L. (1994). *LaTeX - A Document Preparation System: User's Guide and Reference Manual, Second Edition*. Pearson / Prentice Hall, New York.

[Lewis et al., 2020] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

[Li et al., 2024] Li, K., Patel, O., Viégas, F., Pfister, H., and Wattenberg, M. (2024). Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36.

[Lin et al., 2021] Lin, J., Ma, X., Lin, S.-C., Yang, J.-H., Pradeep, R., and Nogueira, R. (2021). Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2356–2362.

[Lo et al., 2005] Lo, R. T.-W., He, B., and Ounis, I. (2005). Automatically building a stopword list for an information retrieval system. In *Journal on Digital Information Management: Special Issue on the 5th Dutch-Belgian Information Retrieval Workshop (DIR)*, volume 5, pages 17–24.

[Luo et al., 2024] Luo, J., Li, T., Wu, D., Jenkin, M., Liu, S., and Dudek, G. (2024). Hallucination detection and hallucination mitigation: An investigation. *arXiv preprint arXiv:2401.08358*.

[Ma et al., 2023] Ma, Y., Cao, Y., Hong, Y., and Sun, A. (2023). Large language model is not a good few-shot information extractor, but a good reranker for hard samples!

[Maynez et al., 2020] Maynez, J., Narayan, S., Bohnet, B., and McDonald, R. (2020). On faithfulness and factuality in abstractive summarization. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

[Meshram et al., 2021] Meshram, S., Naik, N., VR, M., More, T., and Kharche, S. (2021). Conversational ai: Chatbots. In *2021 International Conference on Intelligent Technologies (CONIT)*, pages 1–6.

[Modarressi et al., 2023] Modarressi, A., Imani, A., Fayyaz, M., and Schütze, H. (2023). Ret-llm: Towards a general read-write memory for large language models.

[Muennighoff et al., 2023] Muennighoff, N., Tazi, N., Magne, L., and Reimers, N. (2023). Mteb: Massive text embedding benchmark.

[Nguyen et al., 2016] Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., and Deng, L. (2016). Ms marco: A human generated machine reading comprehension dataset. *choice*, 2640:660.

[Nimavat and Champaneria, 2017] Nimavat, K. and Champaneria, T. (2017). Chatbots: An overview types, architecture, tools and future possibilities. *Int. J. Sci. Res. Dev*, 5(7):1019–1024.

[OpenAI, 2023] OpenAI (2023). Gpt-4 technical report.

[Papineni et al., 2002] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). 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.

[Peng et al., 2023] Peng, B., Galley, M., He, P., Cheng, H., Xie, Y., Hu, Y., Huang, Q., Liden, L., Yu, Z., Chen, W., and Gao, J. (2023). Check your facts and try again: Improving large language models with external knowledge and automated feedback.

[Qian et al., 2021] Qian, H., Dou, Z., Zhu, Y., Ma, Y., and Wen, J.-R. (2021). Learning implicit user profile for personalized retrieval-based chatbot. In *proceedings of the 30th ACM international conference on Information & Knowledge Management*, pages 1467–1477.

[Rahutomo et al., 2012] Rahutomo, F., Kitasuka, T., and Aritsugi, M. (2012). Semantic cosine similarity.

[Reimers and Gurevych, 2019] Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

[Robertson et al., 2009] Robertson, S., Zaragoza, H., et al. (2009). The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

[Rouge, 2004] Rouge, L. C. (2004). A package for automatic evaluation of summaries. In *Proceedings of Workshop on Text Summarization of ACL, Spain*, volume 5.

[Salehinejad et al., 2018] Salehinejad, H., Sankar, S., Barfett, J., Colak, E., and Valaee, S. (2018). Recent advances in recurrent neural networks.

[Setty et al., 2024] Setty, S., Jijo, K., Chung, E., and Vidra, N. (2024). Improving retrieval for rag based question answering models on financial documents.

[Shen et al., 2021] Shen, L., Zhan, H., Shen, X., Chen, H., Zhao, X., and Zhu, X. (2021). Identifying untrustworthy samples: Data filtering for open-domain dialogues with bayesian optimization. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 1598–1608.

[Si et al., 2022] Si, C., Gan, Z., Yang, Z., Wang, S., Wang, J., Boyd-Graber, J., and Wang, L. (2022). Prompting gpt-3 to be reliable. *arXiv preprint arXiv:2210.09150*.

[Sun et al., 2023] Sun, Z., Wang, X., Tay, Y., Yang, Y., and Zhou, D. (2023). Recitation-augmented language models.

[Topsakal and Akinci, 2023] Topsakal, O. and Akinci, T. C. (2023). Creating large language model applications utilizing langchain: A primer on developing llm apps fast. In *International Conference on Applied Engineering and Natural Sciences*, volume 1, pages 1050–1056.

[Touvron et al., 2023] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. (2023). Llama: Open and efficient foundation language models.

[Trivedi et al., 2023] Trivedi, H., Balasubramanian, N., Khot, T., and Sabharwal, A. (2023). Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions.

[TURING, 1950] TURING, A. M. (1950). I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX(236):433–460.

[Van Houdt et al., 2020] Van Houdt, G., Mosquera, C., and Nápoles, G. (2020). A review on the long short-term memory model. *Artificial Intelligence Review*, 53(8):5929–5955.

[Varshney et al., 2023] Varshney, N., Yao, W., Zhang, H., Chen, J., and Yu, D. (2023). A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation. *arXiv preprint arXiv:2307.03987*.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[Vu et al., 2023] Vu, T., Iyyer, M., Wang, X., Constant, N., Wei, J., Wei, J., Tar, C., Sung, Y.-H., Zhou, D., Le, Q., et al. (2023). Freshllms: Refreshing large language models with search engine augmentation. *arXiv preprint arXiv:2310.03214*.

[Walczak, 2019] Walczak, S. (2019). Artificial neural networks. In *Advanced methodologies and technologies in artificial intelligence, computer simulation, and human-computer interaction*, pages 40–53. IGI global.

[Wallace, 2009] Wallace, R. S. (2009). *The Anatomy of A.L.I.C.E.*, pages 181–210. Springer Netherlands, Dordrecht.

[Wei et al., 2023] Wei, J., Kim, S., Jung, H., and Kim, Y.-H. (2023). Leveraging large language models to power chatbots for collecting user self-reported data.

[Weizenbaum, 1966] Weizenbaum, J. (1966). Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45.

[Wu et al., 2016] Wu, Y., Wu, W., Zhou, M., and Li, Z. (2016). Sequential match network: A new architecture for multi-turn response selection in retrieval-based chatbots. *CoRR*, abs/1612.01627.

[Wu et al., 2021] Wu, Z., Galley, M., Brockett, C., Zhang, Y., Gao, X., Quirk, C., Koncel-Kedziorski, R., Gao, J., Hajishirzi, H., Ostendorf, M., et al. (2021). A controllable model of grounded response generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14085–14093.

[Xie and Farooq, 2000] Xie, J. and Farooq, M. (2000). Frequency domain stability test of siso systems with memoryless nonlinear feedback. In *Proceedings of IEEE International Conference on Industrial Technology 2000 (IEEE Cat. No.00TH8482)*, volume 2, pages 206–210 vol.1.

[Xu et al., 2023] Xu, C., Sun, Q., Zheng, K., Geng, X., Zhao, P., Feng, J., Tao, C., and Jiang, D. (2023). Wizardlm: Empowering large language models to follow complex instructions.

[Yu et al., 2023] Yu, Z., Xiong, C., Yu, S., and Liu, Z. (2023). Augmentation-adapted retriever improves generalization of language models as generic plug-in.

[Zhu et al., 2021] Zhu, J., Patra, B., and Yaseen, A. (2021). Recommender system of scholarly papers using public datasets. volume 2021, pages 672–679.