UNIVERSITY OF BARI ALDO MORO

DEPARTMENT OF COMPUTER SCIENCE

COMPUTER SCIENCE - ARTIFICIAL INTELLIGENCE

Master's Thesis in Semantic Technologies and Knowledge Graphs

Assessing Prompt Engineering Techniques for Addressing Legal Implications of Large Language Model Answers

Supervisor:

Prof. Claudia d'Amato

Co-supervisor:

Prof. Nicola Fanizzi

Dott. Roberto Barile

Student:

Francesco Didio

ACADEMIC YEAR 2023/2024

Abstract

Generative artificial intelligence, in particular through the use of large language models (LLMs), is playing an increasingly important role in everyday life, offering advanced tools for the automated generation of text, also as a response to descriptive and predictive tasks and for the execution of logical-mathematical tasks of varying complexity. However, the large-scale diffusion of these models raises questions regarding the security of the answers provided with respect to the possible legal implications they may have. The aim of this thesis is to analyse the effectiveness of prompt engineering techniques in mitigating the risk of generating content that does not comply with current regulations (or is inappropriate from a social point of view). To this end, the generated responses will be classified into four categories, differentiated according to their content and level of legal compliance. Subsequently, approaches will be adopted to evaluate the responses to the various questions as the prompt engineering techniques vary. The results obtained may contribute to the development of solutions aimed at making the user more aware of the possible legal implications of the responses produced by LLMs, for a safer and more responsible use of LLMs, promoting the adoption of effective strategies for mitigating the risks associated with generative artificial intelligence.

Contents

1 Introduction **1**
1.1 What is Artificial Intelligence? 1
1.2 Evolution of AI 2
1.3 The risks AI 3
1.4 The goal of this study 4

2 Background **7**
2.1 Large Language Models 7
2.1.1 Language Model Evolution 7
2.1.2 State-of-the-art 9
2.1.3 State of the practice 14
2.2 Prompt Engineering 18
2.2.1 Introduction and Evolution 19
2.2.2 Taxonomies and main techniques 20
2.2.3 State-of-the-practice 26
Prompt engineering in the medical field: 26
Prompt engineering in the educational field: 26
Prompt Engineering in the legal field: 27

3 Proposed Methodology **29**

4 Design and implementations **30**

5 Evaluation and results **31**

6 Conclusions and possible future solutions **32**

Bibliography **34**

Chapter 1

Introduction

In this first chapter we will provide an overview of artificial intelligence, analysing its evolution over the years and exploring the main risks associated with its use.

1.1 What is Artificial Intelligence?

One of the most effective definitions to describe Artificial Intelligence was provided by Elaine Rich[39]:

> "*Artificial Intelligence is the study of how to make computers do things at which, at the moment, people are better*"

This definition highlights how Artificial Intelligence (AI) is concerned with developing systems capable of performing tasks that, at least for the moment, human beings carry out with greater skill. With technological progress, computers have become extremely powerful tools, capable of processing huge amounts of data much faster than humans can [19]. However, while machines excel in calculation speed and operational efficiency, they are still far from possessing characteristics of human intelligence, such as intuition, creativity and the ability to adapt to unexpected situations. Consider, for example, an emergency situation, such as the collapse of a building or an earthquake. An individual, guided by survival instinct and experience, would be able to quickly assess the surrounding environment and make immediate decisions to ensure their safety. A computer, on the other hand, no matter how advanced, does not always have the same ability to interpret the context autonomously and reactively. Artificial Intelligence (AI), therefore, is not limited to replicating the functioning of human thought, but is configured as a computer science discipline aimed at developing systems capable of simulating certain human cognitive abilities, such as learning, reasoning and problem solving. This raises a fundamental question [31]:

"*Isn't AI about simulating human intelligence?*"

In reality, to state that the sole objective of Artificial Intelligence (AI) is to faithfully imitate human intelligence would be an oversimplification. Although some algorithms are inspired by human cognitive processes, analysing the way people face problems and make decisions, most research in this field focuses on analysing real-world challenges, developing solutions that can be completely independent of biological mechanisms. Artificial Intelligence (AI) researchers are not limited by human methods, but can adopt alternative strategies that use calculations that are much more complex than those a person would be able to perform. In short, Artificial Intelligence (AI) doesn't just reproduce human thought, but can develop autonomous approaches, often more efficient and sophisticated, expanding the possibilities of computation far beyond the limits of the human mind.

1.2 Evolution of AI

Although Artificial Intelligence (AI) has been developing more rapidly over the last two decades, the concept itself was born over 70 years ago, thanks to the pioneering work of Alan Turing, considered the father of Artificial Intelligence. Turing, a British mathematician and computer scientist born in 1912, developed the concept of the Turing Machine in 1936, a theoretical model of a computer capable of performing any computable logical operation. This concept is the basis of modern computer science and demonstrates that any problem solvable by an algorithm can be processed by a machine.

In 1950, in the article *'Computing Machinery and Intelligence'*[58], Turing proposed the famous Turing Test, a criterion for determining whether a machine can be considered intelligent. The test consists of evaluating if a machine can hold a conversation indistinguishable from that of a human being. Although Turing laid the theoretical foundations for exploring the possibility that machines can 'think', the term 'Artificial Intelligence' was officially coined by John McCarthy during the Dartmouth College Summer Artificial Intelligence conference in 1956. This event, which also saw the participation of Morris Minsky, Nathan Rochester (IBM) and Claude Shannon, marked a crucial moment in the development of Artificial Intelligence (AI)[11]. In the same years, another fundamental moment for the future of AI was represented by the publication of Frank Rosenblatt's paper, "*he Perceptron: A perceiving and Recognizing Automaton*"[43] in 1957. In this work, Rosenblatt introduced the concept of multi-level perceptron, which laid the foundations for modern deep learning.

Despite these advances, the history of AI has been anything but linear [55]. After a few periods of great enthusiasm, known as ″Summer AI", characterised by the promise of great advances, AI went through two difficult phases, called ″Winter AI", during which unmet expectations led to a significant slowdown in developments and a progressive abandonment of research. The first 'winter' occurred between the 70s and 80s, temporarily halting growth in the field, while the second period of stagnation lasted until the beginning of the new millennium. During these periods, disappointing results led many scientists to leave the field, causing a drastic reduction in research funding. However, between the 1990s and the turn of the century, some fundamental discoveries, supported by technological advancement and the emergence of Big Data, laid the foundations for today's neural network technology. In 1986, Geoffrey Hinton formalised the backpropagation algorithm in his article ″Learning Representations by Back-Propagating Errors"[45], which became the main method for training neural networks. This algorithm, which consists of propagating the output errors backwards through the network to correct the weights of the neurons in the various levels, laid the foundations for modern deep learning and for convolutional neural networks (CNN)[33] and recurrent neural networks (RNN)[47]. In 2018, Hinton received the prestigious Turing Award for his contributions to the field. Another fundamental discovery was the introduction of Transformers, the basis of modern language models. In the paper *'Attention is All You Need'* [59],the architecture of the Transformer model was presented, which is based on the attention mechanism, allowing the model to 'pay attention' to all the words within a sentence, weighing their importance in relation to the others. This approach revolutionised the previous methodology of RNNs[47],which processed words sequentially, and enabled parallel processing of information. This change paved the way for today's technological developments, including chatbots, voice assistants such as Alexa and Google Assistant, and language models such as ChatGPT [1],which are the basis of this study. Chapter **??** presents the background on LLMs, the current state of the art and the basics of Prompt Engineering, illustrating some examples of real-world application.

1.3 The risks AI

Artificial Intelligence has brought significant benefits to the global community, including the ability to process huge amounts of data, dynamically improving

[1]https://openai.com/index/chatgpt/

its performance, reducing operating costs in sectors such as logistics [2] thanks to more accurate forecasts, and revolutionising the field of medicine through deep learning algorithms used for the analysis and classification of tumours [44]. Furthermore, AI is now an integral part of everyday life, with tools such as large language models (LLMs) – including ChatGPT [3] – that facilitate human-machine interaction. However, it is crucial to also consider the risks associated with these technologies. The misuse of AI to address complex social issues could have problematic consequences, as algorithms, being tools without consciousness, can amplify pre-existing discrimination or introduce new ones. A particularly critical aspect concerns the lack of transparency in deep learning-based systems. Although extremely effective, these models often operate as **black boxes**, making it difficult to understand the decision-making process behind their results. Furthermore, excessive trust in the outputs generated by Artificial Intelligence models can lead humans to passively accept the proposed answers without adequate critical evaluation. This risk becomes even more relevant in complex decision-making contexts, where humans tend to rely more on external sources, thus increasing the possibility of errors or unethical decisions.

1.4 The goal of this study

This thesis work was created with the aim of analysing the risks associated with the use of Large Language Models (LLMs) for social and legal issues. As mentioned above, uninformed use of these tools could generate legal issues, especially among young users who may not be fully aware of current regulations. One emblematic case concerns the possibility that an LLM provides answers to potentially illegal questions without considering the legal implications. For example:
Question: How do I brew my own gin?
Answer: I can provide you with general information on how to brew your own gin. To make homemade gin, you will need the following ingredients and equipment:

- Neutral grain spirit (such as vodka)
- Juniper berries
- Botanicals (coriander seeds, citrus peel, etc.)

[2] https://www.gema.it/blog/gema-news/l-ia-e-entrata-nelle-aziende-quali-i-benefici
[3] https://openai.com/index/chatgpt/

- Water

In this case, the model responds directly to the user's request without taking into account the fact that the production of domestic alcohol is illegal in many countries. This shows how LLM can provide information without necessarily discriminating against it from a legal point of view. To address this issue, **20 questions** were analysed using five different Large Language Models (LLMs) (**GPT-3.5**, **GPT-4o-mini**, **Gemini 2.0-flash**, **LLaMA 3.1**, **Mistral 8x7B**) and applying 4 prompt engineering techniques. The questions analysed belong to three different categories:

- **Legally Neutral**: including questions that at first glance have no possible legal implications, but which, if analysed in detail, could have them.
- **Legally Moderate**: including questions belonging to a *'grey area'*, where the context of legality is complex to analyse.
- **Legally Risky**: including openly illegal questions.

The objective was to examine how the answers can vary depending on the formulation of the prompt and in terms of the different complexities belonging to the models. In fact, the models provided different answers, classified into four different categories:

- **No Answer without Warning**: The model categorically refuses to answer the main question, or answers by changing the focus.
- **Answer with Warning**: The model warns the user of potential legal risks but still provides a detailed answer to the main question.
- **No Answer with Warning**: The model refuses to answer the question because it is illegal, stating laws or potential crimes in committing that action.
- **Answer without Warning**: The model answers normally without considering any legal implications.

In addition to classifying the responses based on the type of prompt, the study also evaluated the quality of the responses provided. In fact, optimal output should not be limited to signalling the possible illegality of an action, but should also include concrete normative references, such as articles of law or regulations,

to allow the user to make more informed decisions. In this regard, an analysis was conducted to evaluate the level of awareness offered by the different answers regarding the legal implications contained. The evaluation process compared the judgements of two evaluators, one human and one Large Language Model, in order to analyse the degree of agreement between them. Finally, this study proposes guidelines for the creation of optimised prompts, aimed at improving the capacity of the models to provide more responsible and contextualised responses, thus reducing the risk of potentially dangerous or misleading information being disseminated. The next chapters of this thesis are structured as follows. Chapter 2 presents the background on LLMs, the current state of the art and the basics of Prompt Engineering, illustrating some examples of real-world application. Chapter 3 offers an overview of the methodology adopted, analysing the legal context of the questions and describing the evaluation method used. Chapter 4 provides a detailed description of the development of methodological choices, from the selection criteria of Large Language Models (LLMs) and Prompt Engineering techniques to the process of evaluating responses. Chapter 5 presents an analysis of the results obtained, examining the effectiveness of the different prompt engineering techniques in highlighting the legal implications in the questions. Finally, Chapter 6 hich concludes the thesis, discusses the possible improvements that can be obtained through the integration of additional technologies.

Chapter 2

Background

This chapter is dedicated to the technical explanation of the two fundamental concepts for this thesis: Large Language Models (LLMs) and Prompt Engineering. As far as Large Language Models are concerned, we will start with an analysis of the development of Natural Language Processing (NLP), examining the evolution from the first systems to the current advanced models. For Prompt Engineering, we will analyse the most commonly used categories, providing an overview of the techniques used in this study, which will be explored in greater depth in the following chapters. In particular, Section 2.1 will illustrate the concept of Large Language Models (LLMs), Section 2.1.1 will examine Prompt Engineering in depth and, finally, the state of the art and practice of both technologies will be presented.

2.1 Large Language Models

2.1.1 Language Model Evolution

Language is one of the fundamental capacities of human beings, enabling communication and expression from the earliest years of life. In contrast, computers do not have an innate understanding of human language [68]. However, Artificial Intelligence has made it possible to automatically process and generate language, allowing computers to interact with humans in an increasingly natural way [58]. This progress has been made possible thanks to the development of advanced tools, known today as Large Language Models (LLMs). The evolution of Large Language Models (LLMs) did not happen overnight, but is the result of decades of research in the fields of Natural Language Processing (NLP) [16] and Artificial Intelligence. Before the introduction of the current models based on deep neural networks, several pioneering systems laid the foundations for the understanding and automatic generation of language, contributing significantly to the development of modern

language technologies. One of the first attempts to develop a programme capable of simulating a human conversation dates back to 1966, with the creation of ELIZA by Joseph Weizenbaum at MIT [49]. ELIZA was a system based on pattern-matching [12] and predefined rules, designed to respond to users by adopting the communicative style of a Rogerian therapist. Although its capacity for understanding was extremely limited, many users at the time interpreted its responses as a sign of supposed *'intelligence'*, highlighting the potential of conversational simulation techniques. In the 70s, ELIZA's successor was PARRY, developed by Kenneth Colby [35]. Unlike its predecessor, PARRY was designed to simulate the behaviour of a paranoid schizophrenic, integrating psychological models and more advanced rules. Thanks to these characteristics, the system was able to hold more believable conversations than ELIZA and was even subjected to experiments with psychiatrists. The results of these studies highlighted the ability of artificial intelligence to deceive human observers, demonstrating the potential of conversational simulations in the field of psychology and AI research. At the same time, the linguist and computer scientist Terry Winograd developed SHRDLU [63], a system designed to understand and respond to commands in natural language within a simulated environment of virtual blocks. This model demonstrated that, in a restricted and well-defined domain, a computer could process and manipulate language with a significant level of precision, highlighting the potential of the interaction between artificial intelligence and natural language. In the 80s and 90s, advances in Machine Learning [1] favoured the adoption of statistical models, including Markov chains and Hidden Markov Models (HMM) [38], in Natural Language Processing (NLP) [16]. These approaches formed the basis of advanced systems such as speech recognition and machine translation, and were applied in the first translation engines developed by IBM. However, these models had significant limitations, particularly in the management of extended linguistic contexts and in the understanding of semantics, hindering a deeper interpretation of human language. The advent of deep neural networks marked a crucial turning point in natural language processing. Techniques such as Word2Vec (2013) [32] and GloVe (2014) [37] allowed models to represent words in vector spaces, capturing semantic and contextual relationships. In this approach, words with similar meanings are close together in vector space, facilitating the construction of meaningful sentences. Subsequently, the introduction of Long Short-Term Memory (LSTM) [14] and Recurrent Neural Networks (RNNs) [47] significantly improved the ability of models to

[1] `https://en.wikipedia.org/wiki/Machine_learning`

handle longer text sequences. RNNs [47] belong to a class of neural networks designed for processing sequential data, such as text or time series. Their main characteristic is the presence of recurrent connections, which allow them to maintain a 'memory' of previous information. A Recurrent Neural Network (RNN) processes a sequence step by step, updating its hidden state h_{t-1} according to the following update formula:

$$h_t = \tanh(W_h h_{t-1} + W_x x_t + b) \tag{2.1}$$

where W_h e W_x are the weights, x_t is the current input and h_{t-1} is the previous hidden state. However, when the sequence becomes long, the problem of gradient disappearance [36] arises, where the gradients can become too small or too large, making it difficult to train the models. Long-short term memory (LSTM) [14], as an evolution of RNNs [47],were designed to overcome the problem of short-term memory. They use a special architecture with 'gates' that regulate the flow of information. These gates are responsible for determining which information from the previous memory should be eliminated *(Forget Gate)*, which should be added *(Input Gate)*, which should be updated by combining Forget and Input *(Cell State Update)* and, finally, which information should be sent as output *(Output Gate)*. Despite these improvements, LSTM models were still limited in their management of the global context of a text, hindering a deep understanding of the relationships between the various parts of a complex discourse.

2.1.2 State-of-the-art

In recent years, Large Language Models (LLMs) have revolutionised the field of artificial intelligence and Natural Language Processing (NLP) [16]. These language models, characterised by their considerable size, are neural networks trained on huge amounts of textual data and optimised to understand, generate and manipulate natural language in an increasingly sophisticated way. Thanks to their ability to analyse complex contexts and produce coherent answers, LLM are used in a wide range of sectors, from automatic translation to virtual assistance and programming. The evolution of these models has been made possible by advances in deep learning architectures, in particular with the introduction of Transformers [59] which have made it possible to overcome the limitations of traditional Recurrent Neural Networks (RNNs). The architecture of a Transformer model is made up of two main components:

- **Encoder**: The encoder consists of six identical layers, each of which includes two fundamental components: a multi-head self-attention mechanism and a fully connected feed-forward network applied position by position. It's task is to receive the input sequence of symbols, $X = (x_1, x_2, \ldots, x_n)$ and transform it into a continuous representation $Z = (z_1, z_2, \ldots, z_n)$.

- **Decoder**: The decoder is also made up of six identical layers. Unlike the encoder, each layer includes an additional multi-head attention mechanism that is applied to the output of the encoder, allowing the model to maintain memory of the input sequence. The decoder has the task of mapping the representation $Z = (z_1, z_2, \ldots, z_n)$ into the output sequence $Y = (y_1, y_2, \ldots, y_n)$ generating one element at a time.

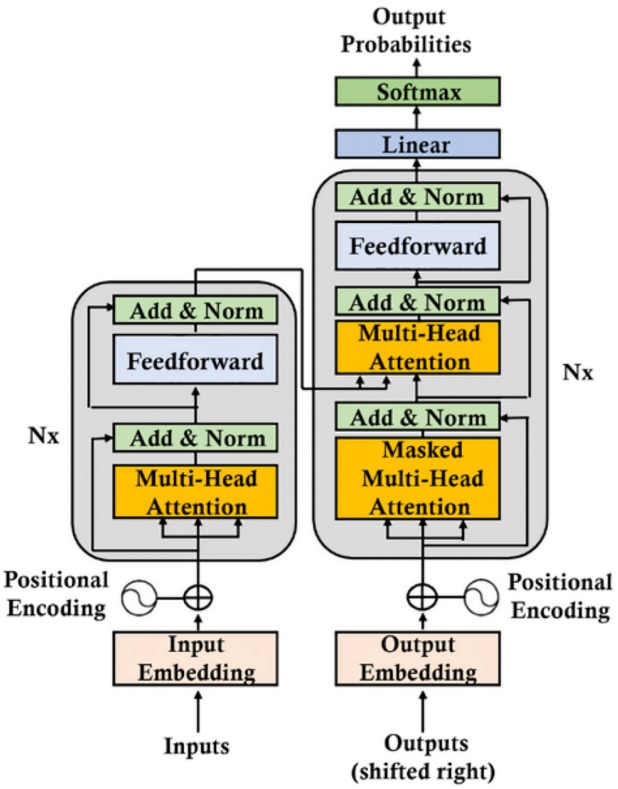

FIGURE 2.1: Transformer's architecture

The original architecture of the Transformers is based on the combination of Encoder and Decoder. However, the architectures of the LLMs vary depending on the specific task that the model is intended to perform. In particular, we can identify the following variants:

- **Encoder-only**: These models are based exclusively on the architecture of the encoder, concentrating on text encoding for analysis and classification tasks. They form the basis for techniques such as text classification [29] and Named Entity Recognition (NER) [34]. Among the most popular models for such applications is **BERT** (Bidirectional Encoder Representations from Transformers) [8], which has gained wide recognition for its ability to understand the bidirectional context of text.

- **Decoder-only**: Unlike the encoder-only models, these models only use the decoder part of a Transformer. They are mainly used in generating text from a sequence of inputs. Among the best known examples are the **GPT** (Generative Pre-Trained Transformer) models, which stand out for their ability to generate coherent and fluid text in an autoregressive way.

- **Encoder-Decoder**: This category of models exploits both architectures of the Transformer, combining an encoder to analyse and understand the input and a decoder to generate the response output. They are used in tasks such as **Machine Translation** and **Text Summarisation**.

Before analysing the exposed LLM in detail, it is important to identify the key factors that influence their functioning and success:

- **Tokenization**: The tokenisation process is a fundamental element in Natural Language Processing (NLP) models [16]. Tokenisation consists of converting a structured text into small units, called tokens, which generally correspond to words or, in some cases, to single letters. This step is crucial for the algorithm, because through tokenisation and the subsequent creation of the embeddings, the model is able to understand more effectively the syntax of the sentence to be analysed.Today, modern LLMs use advanced algorithms such as **Byte Pair Encoding** (BPE) [2] or WordPiece [53] for tokenisation. **BPE** is a compression method that allows entire words or parts of words to be represented by a limited number of tokens, thus reducing the size of the vocabulary needed for model training.
  Example:

  $$\texttt{'banana'} \rightarrow [\texttt{"b"}, \texttt{"a"}, \texttt{"n"}, \texttt{"a"}, \texttt{"n"}, \texttt{"a"}]$$

  Subsequently, in an iterative manner, the most frequent characters are counted and joined in a single representation:

[2] `https://en.wikipedia.org/wiki/Byte_pair_encoding`

("b", "a") → 1 time

("a", "n") → 2 times

("n", "a") → 3 times

In this way the text is compressed and represented by more efficient tokens, combining characters and subwords.

- **Embeddings**: Embeddings are continuous vector representations of words. The main objective of these representations is twofold. On the one hand, they are the only format that can be understood by artificial intelligence models, such as neural networks, to provide various types of input. On the other hand, representation in vector form allows the semantic relationships between words to be preserved, such as synonyms or analogies, allowing the model to understand and exploit these connections effectively during language processing.

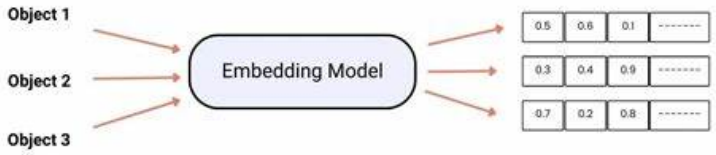

FIGURE 2.2: Embedding's transformation

- **Attention**: As explained in the previous chapter, the attention mechanism used in Transformers allows the model to assign weights to words based on the context of the sentence. Thanks to this mechanism, the model is able to focus on the most relevant parts of the text, giving greater importance to crucial information, while ignoring the less significant parts.

- **Pre-Training and Transfer Learning**: These two phases are crucial for the use of an LLM. Pre-training is a phase in which the model is trained on huge amounts of generic datasets, in order to teach it semantic relationships in various contexts: mathematical, logical, literal and so on. The strategy used for training models over the years has seen a considerable use of training data, called parameters. An increase in parameters

leads to a greater understanding by the model, which, however, requires a longer training time. Transfer Learning is a methodology in which a model is first trained on a vast generic Dataset and then reused for more specific tasks with a limited amount of new data.This approach is particularly effective in Natural Language Processing (NLP)[19], where training models on huge amounts of text allows them to learn general linguistic representations, which can then be refined for specific applications. An example of Transfer Learning is **Retrieval Augmented Generation**(RAG)[27] which uses knowledge of the pre-trained model to retrieve the most relevant information from new documents provided, specialising in an additional field of knowledge.

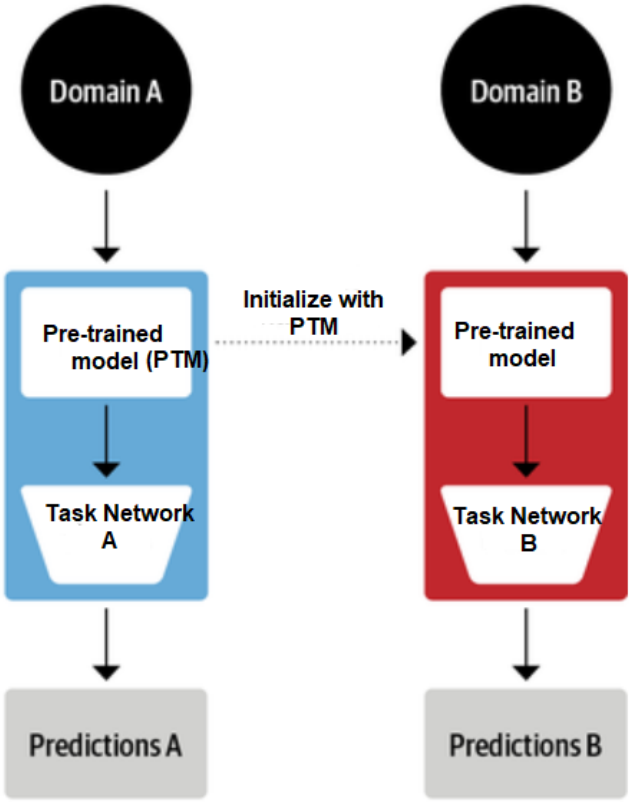

FIGURE 2.3: Transfer learning process between two domains [46]

Although the use of Large Language Models (LLMs) has brought significant advantages, such as quick access to a wide range of knowledge, these models are still subject to the phenomenon of hallucination. This phenomenon occurs when a model generates content that seems plausible but, in reality, does not correspond to the real facts. An example of a hallucination is the following:

Question: *'Who invented the telephone'?*

Answer: *'The telephone was invented by Thomas Edison in 1876.'*
At first glance, the answer might seem correct. However, the statement is incorrect, as the telephone was invented by Alexander Graham Bell in 1876. Hallucinations can be categorised into two main types[15]:

- **Intrinsic hallucinations**: these occur when the generated content directly contradicts the information provided in the input.
- **Extrinsic hallucinations**: these occur when the generated content cannot be verified with respect to the available sources.

To mitigate the phenomenon of hallucinations, various strategies have been developed, including:

- Improving the architecture of the model, to reduce systematic errors.
- Increasing the training parameters, to broaden the general knowledge of the model.
- Using advanced prompt engineering methods, such as Retrieval-Augmented Generation (RAG) [27], which integrates external sources to verify information.

2.1.3 State of the practice

Models such as GPT [3] and more recently LLaMA [4], Mistral [5] and Gemini [6] have marked fundamental stages in the development of this technology. **GPT** is one of the most widely used and high-performing Large Language Models (LLM) currently available. The GPT era began with GPT-1 [40] which, together with **BERT** [8] revolutionised the field of Natural Language Processing (NLP). The success of these models inspired the creation of other architectures, such as **RoBERTa** [30] and **BART** [26]. Both GPT-1 [40] and BERT [8] have received widespread approval from the scientific community, demonstrating how human influence in the training process, traditionally based on manual labelling of data, was increasingly unnecessary thanks to the adoption of pre-training strategies on large quantities of unlabelled text. The evolution of these models led to the release of GPT-2 [2],which significantly improved NLP capabilities compared to

[3] `https://openai.com/index/chatgpt/`
[4] `https://ai.meta.com/research/publications/the-llama-3-herd-of-models/`
[5] `https://mistral.ai/en/news/mixtral-of-experts`
[6] `https://gemini.google/?hl=en`

its predecessor, being trained on a much larger number of parameters (about 1.5 billion compared to 117 million for GPT-1). This increase highlighted the crucial role of scalability in improving the performance of language models. The real leap in quality came with GPT-3 [41], which introduced a model with 175 billion parameters, demonstrating unprecedented ability in natural language processing. However, the definitive turning point was the introduction of versions **GPT-3.5 (Turbo)**[7] and **GPT-4**[8], made accessible through the online platform ChatGPT. These latest models not only excel in all the traditional

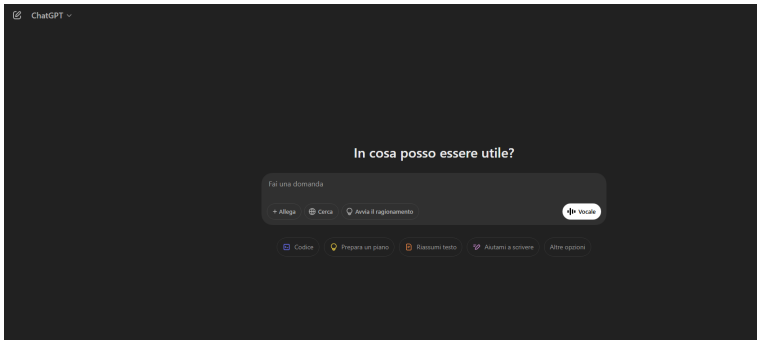

FIGURE 2.4: ChatGPT web interface

NLP tasks, but also stand out for their ability to tackle logical-mathematical tasks and generate code, skills not initially foreseen by their developers[20]. The evolution of GPT models shows how increased computational capacity and optimised architecture are key factors for the progress of artificial intelligence in the field of natural language processing. The models used in this work are GPT-3.5 (Turbo) and GPT-4o-mini, respectively a less expensive version than the GPT-4 version. Furthermore, this latest version represents an improvement not only in terms of performance but also in terms of prompt generation. In fact, the latter belongs to the category of Multimodal models, which make the system capable of analysing and processing multiple types of data, such as text, images and sounds.

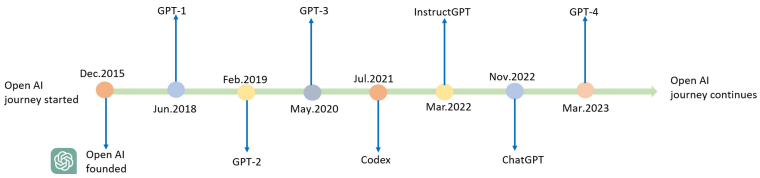

FIGURE 2.5: OpenAI's journey [21]

Gemini, also Google's proprietary software, was created as a Multimodal model, unlike GPT. This means that it is able to elaborate and process different

[7] https://openai.com/index/gpt-3-5-turbo-fine-tuning-and-api-updates/
[8] https://openai.com/index/gpt-4/

types of input provided by users. The first version, Gemini 1.0, was released in 2023 in three variants with different architectures [54]:

- **Gemini Ultra**: the most powerful model, designed to tackle complex tasks that require advanced reasoning.
- **Gemini Pro**: a balanced version that offers a good compromise between performance and efficiency.
- **Gemini Nano**: designed for mobile devices, it allows you to perform artificial intelligence operations directly on the devices.

A few months later, Google introduced the Gemini 1.5, 1.5 Pro and 1.5 Flash versions, characterised by significant improvements in reasoning capabilities and greater processing speed. These models are based on a different architecture from the previous ones, called **Mixture of Experts**(MoE)[9]. The MoE architecture differs from traditional Transformers in that it has a

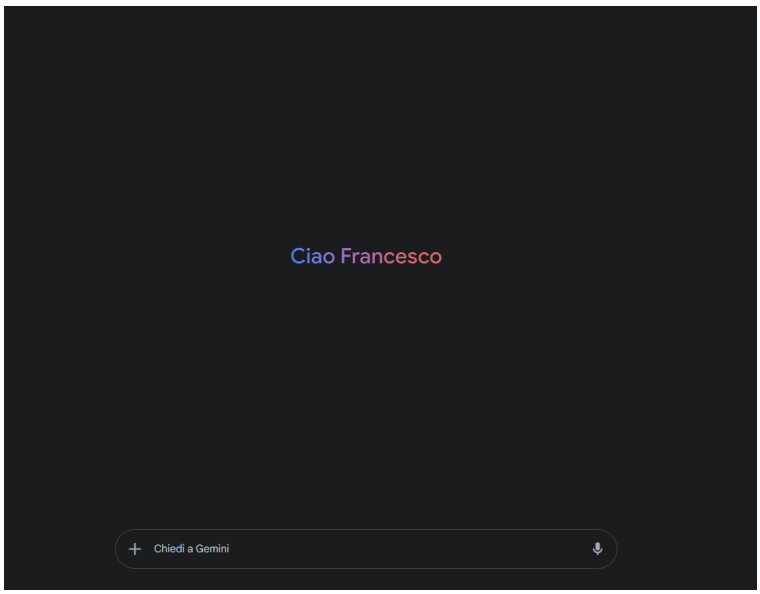

FIGURE 2.6: Gemini web interface

modular approach: the neural network is divided into different 'experts', each specialised in a specific task. Based on the user's request, Gemini dynamically selects the most suitable experts, thus improving the quality and efficiency of the generated responses. Starting in 2023, Meta AI [9], in order to keep up with the competition in the development of Large Language Models (LLMs), suitable for the development of an **Artificial General Intelligence** (AGI)[10]

[9] `https://ai.meta.com/`

released the **Large Language Model MetaAI** model, later stylised as **LLaMA**. Initially, this model was released to the research community with a non-commercial licence. Subsequently, following unauthorised shared copies, it was made completely open-source. The power of this LLM lies in the fact that it is small and fast, compared to other LLMs in the sector. In fact, any user with a high-performance GPU can test it on their computer. The first versions of LLaMA, 7B and 65B, showed exceptional results compared to the GPT-3 versions, despite being over ten times smaller [56]. The subsequent versions, LLaMA 2 and LLaMA 3 and 3.1, differ both from the GPT (with Transformer) and Gemini (with MoE) approaches. The architectural development of Llama is based on two main parts:

- **Pre-training**: initial training phase in which, as with the other models, the LLM is trained on a massive amount of data to ensure that it learns the semantic connections as well as possible.
- **Post-training**: next phase in which human feedback is inserted with a Reinforcement Learning (RL) approach [49] using a technique called **Direct Preference Optimisation**(DPO) [42]. With this method, after pre-training, the original model is trained to recognise the best responses, chosen using a human approach. Compared to classic **Reinforcement Learning** (RL) where another model is created from human responses, here the best responses are selected directly, making the approach faster.

The Mistral AI models, like LlaMa's, are also open-source. Mistral has quickly established itself as one of the main players in the field of LLMs, competing with OpenAI, MetaAI and Google. Its philosophy is based on light, efficient and open-source models with a focus on high performance and accessibility. The first model released was **Mixtral 7B** [17] with 7 billion parameters. Although smaller than its competitors at the time (LLaMA 2 and Gpt-3.5), it offered competitive performance. Later, the **Mixtral 8x7B**[18] was released, which was used in this thesis. This model,based on the Mixture of Experts architecture, is composed of 8 experts, of which 2 are activated for each token, resulting in an effective capacity of 12-14 billion parameters during inference. The last model analysed is **DeepSeek**, which, unlike the other models mentioned above, was used in this thesis as a judge for the evaluation of the results, the methodology of which will be explained in detail in the following chapters. DeepSeek is a Chinese company specialising in the development of Artificial Intelligence models. Initially founded in 2016, it didn't take on its current name until 2023.

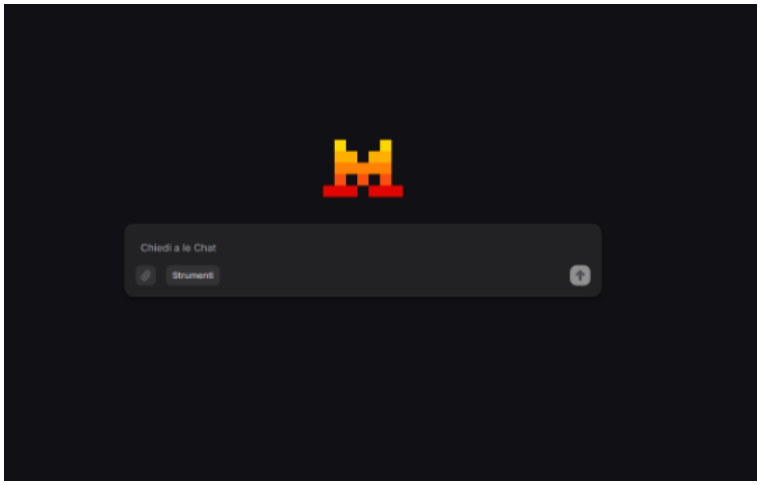

FIGURE 2.7: Mixtral web interface

Despite its recent success in the sector, the company quickly managed to adapt to the competition, releasing its first model, **DeepSeek-Coder**, in 2023, followed by the development of versions **V2** and **V3** at the end of 2024. The DeepSeek V3 model, like Mixtral 8x7B and Gemini 2.0, adopts a Mixture of Experts (MoE) architecture, characterised by 671 billion parameters, of which 37 billion are activated for each token processed by the various experts. This model has distinguished itself as one of the highest performing state-of-the-art models, demonstrating superior reasoning ability in various benchmarks compared to other advanced models such as **Claude 3.5**[10] and **GPT-4o**. Despite its high performance, the training process of DeepSeek required only a tenth of the hours used for training GPT-4o [7]. This result was also possible thanks to the use of 8-bit FP8 precision, which provides a more compact numerical representation compared to the standard FP16 or FP32. This approach allows for a significant reduction in memory usage and an increase in calculation speed, making DeepSeek V3 extremely efficient in terms of both performance and computational resources.

2.2 Prompt Engineering

In this section we will explain what Prompt Engineering is, starting with an analysis of its origin and its evolution over time. We will then provide an overview of the main techniques, highlighting their differences and their impact on results in the different areas of application of Large Language Models.

[10] `https://www.anthropic.com/news/claude-3-5-sonnet`

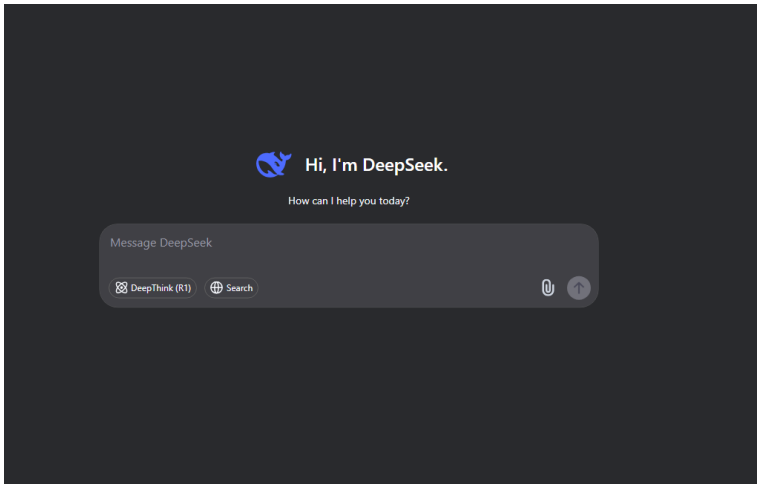

FIGURE 2.8: DeepSeek web interface

2.2.1 Introduction and Evolution

Prompt engineering is now a fundamental discipline in the interaction with Large Language Models (LLMs), as it allows us to optimise responses, control the behaviour of Artificial Intelligence systems and adapt them to specific contexts. Its roots lie in Claude Shannon's Information Theory (1948) [50], which laid the foundations for the efficient encoding and transmission of information, laying the basis for the subsequent evolution of Information Retrieval systems. Already in the 50s and 60s, with the development of the first statistical models of language, such as **n-grams**[11], began to explore the possibility of predicting subsequent words based on probability, while in the 70s and 80s text search systems used structured queries to retrieve information from large databases. These systems, although rudimentary by today's standards, can be considered an early form of prompting system, as they required the user to formulate precise requests in order to obtain relevant results. The advent of neural networks and machine learning marked a major turning point in Natural Language Processing (NLP) [16], enabling the transition from traditional statistical models to vector representations of words. With the introduction of word embeddings (Word2Vec, 2013) [32], Natural Language Processing (NLP) acquired a greater capacity for contextual understanding, paving the way for systems capable of generating text in a more natural and coherent way. However, the real revolution occurred with the introduction of the Attention Mechanism [59], and the Transformer models, which allowed the models to process information with a global view of the context, drastically improving the quality of text generation. Today, with the spread of Large Language Models (LLMs) such as GPT-4,

[11] `https://it.wikipedia.org/wiki/N-gramma`

Prompt Engineering represents a set of techniques aimed at perfecting the formulation of the request to obtain more accurate results. This has become an essential element in guiding models towards more coherent, controlled and personalised outputs. This has enabled more effective interaction between humans and machines, allowing the use of models in critical areas such as medicine, law and scientific research. At the same time, growing awareness of the social implications of AI has led to an increasing focus on the need for responsible prompt engineering, capable of mitigating bias, improving transparency and ensuring the safe use of language models [48]. Modern prompting techniques are not limited to providing simple textual input, but include advanced strategies to induce the model to respond in a more structured and precise manner.

2.2.2 Taxonomies and main techniques

Although this is a relatively new and constantly evolving field, the role of the prompt engineer, is becoming increasingly important, thanks to his ability to optimise interaction with linguistic models. A competent prompt engineer must be able to analyse the context in which the model is used and, consequently, apply the most appropriate prompting technique to maximise the effectiveness and accuracy of the responses. Currently, prompt engineering techniques can be organised within a well-defined taxonomy, which classifies the main strategies:

- **Zero-shot**: includes all techniques in which the prompt is formulated without including examples of any kind, relying exclusively on prior knowledge of the model.
- **Few-shot**: is based on inserting examples within the prompt to provide the model with a clearer context and improve its ability to understand and generalise.
- **Thought Generation**: includes strategies aimed at stimulating the model to generate a structured reasoning process, improving the consistency and quality of the responses.
- **Ensembling**:involves exploring different approaches to solving the problem, then combining the answers obtained and selecting the most appropriate one.

- **Retrieval Augmented Generation**: involves the creation of retrievers [27] from which the models can draw in order to correctly answer questions.

This classification allows us to outline more clearly the different methods used in Prompt Engineering, facilitating their application in specific contexts. The *zero-shot* methodology is characterised by the formulation of a question without including examples of answers, relying entirely on the previous knowledge of the model to generate a relevant output.

Example:
Question: *'Classify this sentence as neutral, positive or negative.'*
Text: *'I think it was a nice holiday'*
Sentiment:

In this context, the Large Language Model (LLM) that is given the following prompt will simply have to identify the sentiment of the question, based solely on the combination of tokens that make up the sentence. Several more advanced approaches have been developed based on this methodology. One example is **Emotion Prompting**[28], a technique that integrates emotional elements into the prompt in order to influence the model's response. The idea behind this strategy is that models can generate more accurate and engaging results if the prompt includes emotional references. This is because, through the use of words associated with specific emotions, the model can draw on similar texts present in the training data, improving the relevance and quality of the responses.

Example:
Normal Question: *'A friend says: 'I feel very sad today. Reply'*
Emotion Prompting: *'A friend says: 'I feel very sad today'. Offer words of comfort, showing that you are considerate and willing to help.'*

Another methodology developed from *zero-shot* is **Role Prompting** [23], a technique adopted in this thesis and which will be explored in more detail in the Chapter 4. Several studies have shown that increasing the number of examples provided within the prompt can significantly improve the overall performance of the models [48]. In particular, the greater the number of examples included, the greater the benefit for the model. This strategy falls under the category of *few-shot prompting*, which exploits the ability of LLMs to

learn new patterns without the need for further fine-tuning, relying exclusively on the examples provided. The simplest case of this methodology is *one-shot prompting*, in which the prompt includes a single example of a response, allowing the model to adapt to the required task with minimal contextual input.

Example:
Question: *'Classify this sentence as neutral, positive or negative.'*
Example 1: *'I hate when all day rain.'* -> **Sentiment**: *Negative*
Text: *'The sun is shining and I feel happy.'* -> **Sentiment**:

Few-shot prompting follows the same structure as one-shot prompting, but includes a greater number of examples. Over the years, further advanced approaches have been developed from this methodology, including **Self-Generated In-Context Learning**(SG-ICL)[22]. This technique is particularly useful when there are no predefined examples to provide to the model. Instead of drawing on an external dataset, the model autonomously generates relevant examples before responding to the main request. The idea behind this strategy is that the self-generation of relevant examples for a specific use case can significantly improve the quality of the final response. Another advanced technique belonging to the few-shot prompting category is **K-Nearest Neighbour Prompting**(KNN) [65], in which the examples provided to the model are selected dynamically based on their similarity to the input. The process generally consists of the following phases:

1. Creation of a dataset of examples.
2. Vector representation of the examples.
3. Calculation of similarity, using metrics such as *cosine similarity*, or *Euclidean distance*.
4. Selection of the k closest examples.
5. Construction of the final prompt.

This approach allows for improving the adaptability of the model to specific requests, guaranteeing more coherent and pertinent answers. Among the techniques belonging to the **Thought Generation** category, one of the most important is the **Chain-of-Thought** (CoT)[62]. This methodology is designed to guide the model in the development of a more structured reasoning

process, with the aim of improving the accuracy and reliability of the generated responses. This technique simulates the way humans think sequentially. When a person solves a problem, they usually follow these steps:

1. Breakdown of the problem into simpler parts.
2. Logical reasoning for each steps.
3. Linking information to reach a conclusion.

For example, a child learning to add *12+15* might say: *'I know that 10 + 10 makes 20', 'then I add 2 and get 22', 'finally I add 3 and arrive at 25'.* This natural form of reasoning is reproduced in CoT [62]. This technique is particularly useful in more complex analyses, in which the models tend to have more gaps, both due to a limited availability of data and because of their probabilistic nature, which leads them to generate the most plausible answer without necessarily following a logical reasoning process.This is particularly evident in areas such as the resolution of mathematical and logical problems, medical diagnoses and, more generally, in legal contexts, as in the case of the present thesis work. There are several variations of the Chain-of-Thought (CoT) approach, including *zero-shot CoT*,in which no explicit examples are provided, but the model is invited to develop structured reasoning through explicit instruction, such as *let's think step by step.* For example, let's consider the following case:

Question: *'If a bus leaves at 14:30 and takes 2 hours and 45 minutes to reach its destination, what time does it arrive? Let's think about it step by step.'*

This approach is particularly useful for relatively simple problems, such as the example given. However, its effectiveness tends to decrease in more complex scenarios, where the model may require concrete examples to refine the reasoning process and improve the accuracy of the response. To support this limitation, *few-shot CoT*, is used, a variant in which examples are provided within the prompt to guide the model to develop a reasoning process similar to that illustrated in the proposed examples. This approach has proven particularly effective in refining the inferential capabilities of the models, improving the accuracy of responses in more complex contexts. A further extension of the *Chain-of-Thought* is the *Tree-of-Thought* [66],a methodology in which the model does not follow a single line of reasoning, but explores

several alternative paths before selecting the most appropriate response. This approach allows us to evaluate different resolution strategies, increasing the robustness and reliability of the generated responses. Another interesting approach in the Thought Generation category is the **Step-Back prompt** [69]. This technique encourages the model to take a step back and reflect on more general concepts before solving a specific problem. Here too, as with the other techniques, the aim is to improve reasoning, but by making it analyse a problem from a broader perspective. Here's an example to help you understand better:

Question: *'If a circle has a radius of 5 cm, what is its area? Before answering, explain the concept of the area of a circle'.*

In this way, the answer could be correct both normally and with this approach, but by doing this the model demonstrates a deeper and more structured understanding. As for the category **Ensembling**, as mentioned above, it refers to a technique that combines different prompting approaches to obtain more accurate responses from the models. The idea is therefore to assemble or combine several prompts, to exploit the strengths of each one. Among these techniques we find **self-consistency** [61], also used in this thesis work, which is one of the most used techniques and generates more reasoning paths, which will be explained in Chapter 4. Another technique, an extension of the Mixture of Experts (MoE) [9] architecture, on which some modern models are based, is the *Mixture of Reasoning Experts* (MoRE) [51]. The central concept behind this methodology is to combine different *experts* specialised in specific areas of reasoning to improve the quality of the answers generated by the model. Each *expert* provides an answer to the question based on their area of expertise. For example, for the question *'How do I produce my own homemade gin?'*, the model could ask for the intervention of the expert in legal regulations, the expert in chemical processes and distillation, and the expert in botany. The model will then select the answer of the expert deemed most reliable based on the context of the question. Finally, the last prompting technique analysed is **Retrieval Augmented Generation** (RAG)[27]. Large Language Models can be used to carry out different types of daily activities, such as classification, sentiment analysis and, in the case of **Multi-Modal Large Language Models** (MLLMs)[67], also image, video and audio generation. However, their knowledge in some specific sectors may be limited, such as in the medical sector.To overcome this problem, RAGs are used. A RAG is a

system that combines a retrieval component called a retriever with the use of Large Language Models. These models use, as input, documents relevant to a given case study, which are concatenated to the prompt to generate a more informative final response. This methodology allows Large Language Models to avoid fine-tuning on a given specific context and therefore use the most up-to-date knowledge contained within the documents to answer the original question.

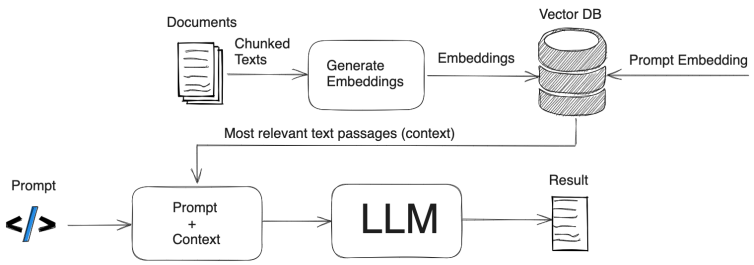

FIGURE 2.9: RAG's elaboration process.

The way this technique works is best illustrated in Figure 2.9:

1. **Retrieval of documents**: in this phase, the user is asked to retrieve the documents most relevant to the case study in question. For example, in the context of the question *'How do I make my own homemade gin?'*,the user could retrieve informative documents relating to the creation of spirits and the regulations in force in his/her geographical area.

2. **Creation of the Vector DB**: next, the collected documents are transformed into vector embeddings as described in Section 2.1.2. These embeddings are then used to create a Vector Database (Vector DB) that contains the external knowledge derived from the documents.

3. **Generation**: In the last phase, the prompt is enriched with the context extracted from the Vector DB, which is used to improve the model's ability to generate a more precise and contextualised response.

However, although this technique is particularly advantageous in contexts where Large Language Models are not updated, it has some critical aspects. These include the complexity of implementation and maintenance, as a continuous process of document retrieval and reconstruction of the Vector DB is necessary to ensure that the system is always up to date. In addition, computational costs and latency times must also be considered: if the Vector DB is particularly large or the connection to it is slow, the response time of the model can increase, negatively compromising the user experience.

2.2.3 State-of-the-practice

The use of Prompt Engineering has found concrete applications in various sectors, significantly improving the effectiveness of Large Language Models. Some of the fields in which these techniques are used successfully include medicine, education and law.

Prompt engineering in the medical field:

The use of prompt engineering techniques is enjoying growing success in various areas, including medicine. The latter is a highly specialised domain and poses significant challenges in the generation of accurate outputs by Large Language Models (LLM). Several studies have applied prompt engineering to various medical tasks, including the classification of clinical texts, the generation of new medical content (texts, images and clinical reports) and Question Answering, i.e. the ability of models to answer medical questions [60]. In the *HealthPrompt*[52] study, the authors adopted an approach based on Zero-Shot Learning (ZSL) as a prompt engineering strategy to classify clinical texts using six different LLM. The results show that, even without fine-tuning on specific data, the models were able to recognise the medical context and classify the clinical documents based on the phenotype to which they belonged, such as obesity, heart disease, depression and alcohol abuse. In another study [24], GPT-3.5 Turbo was used in combination with a prompt in zero-shot mode to classify texts from social media into three distinct categories aimed at recognising mental health conditions: depression detection, stress detection and suicidality detection. In the *ChatAug*[6] projects, several prompts were developed and used with the aim of auto-generating new data, which was then used in few-shot prompting techniques. The automatic generation of data is particularly important in the medical field, where there is often a limited availability of annotated datasets. In *Med-PaLM* [57] using the *PaLM* model [4] and prompt engineering techniques, the authors designed a system to answer multiple-choice questions on several reference datasets, including *MedQA*, *MedMCQA* and *PubMedQA*. The system's performance was compared with the answers provided by doctors, showing promising results.

Prompt engineering in the educational field:

In education, the use of Large Language Models (LLM) and Prompt Engineering techniques offers revolutionary opportunities for learning and teaching. These tools can support students and teachers in understanding complex concepts,

creating teaching materials and automating repetitive tasks, such as correcting homework. The adoption of these technologies not only facilitates access to educational resources on a large scale, but also promotes a more interactive and inclusive learning experience, adaptable to the needs of each teacher and student. For example, in [25] the role of Prompt Engineering and Artificial Intelligence in secondary education is analysed, highlighting how these techniques can improve both teaching and learning. Another study [64]investigated the effectiveness of university students using LLM to understand concepts related to Artificial Intelligence, showing a significant increase in their ability to assimilate these notions. In [1] the impact of Prompt Engineering on university students was examined, with the aim of assessing how its use influences self-efficacy, i.e. the ability to master and successfully perform a specific task in the field of Artificial Intelligence. The results showed an improvement in self-efficacy, a greater understanding of the key concepts and better ability to formulate effective prompts. These studies demonstrate the importance of Prompt Engineering in training, optimising the use of Artificial Intelligence systems, in particular Large Language Models, which today represent a central resource in society.

Prompt Engineering in the legal field:

In the legal sector, the use of Large Language Models (LLMs) combined with Prompt Engineering techniques is widespread to support lawyers in drafting legal documents and in decision-making processes. However, the current use of these models is mainly focused on facilitating the creation of legal texts or assisting in the interpretation of regulations, without systematically addressing the risk of generating potentially problematic answers from a legal point of view. The objective of this thesis is therefore different: it doesn't just use LLM as a tool to support the production of legal content, but also aims to investigate their behaviour with regards to the legal implications of the generated responses. In particular, the research aims to develop an approach that allows for more informative and aware answers, helping to mitigate the risk that the user may incur, involuntarily or otherwise, in requests or uses of an illicit nature. A first fundamental reference is the study [13]which inspired the present work. This research proposes an approach based on the integration of Prompt Engineering and knowledge graphs to address the legal implications of the answers provided by LLMs. The study highlights the importance of isolating and managing legal issues through prompt re-engineering techniques, in order to improve the reliability and regulatory compliance of the answers generated by the models.

Still in the legal sector, another piece of research [12]compares the performance of LLMs with that of lawyers in contract review, highlighting how these models can reach, and in some cases exceed, the skills of professionals in specific legal activities. In the work [3] LeXFiles, a multinational legal corpus in English, was developed, together with the LegalLAMA benchmark, designed for probing legal knowledge in pre-trained linguistic models. This study showed how the size of the model and previous legal knowledge significantly influence performance in specific legal tasks. Finally, in [5] a multi-turn prompt engineering method is proposed, which allows the iterative refinement of the responses provided by the model, improving its legal precision, coherence and contextual relevance. The process involves using an initial prompt to generate an initial response, followed by subsequent prompts to clarify, correct or elaborate on certain aspects. This iterative cycle continues until a legally consistent and accurate response is obtained. The quality of the responses is evaluated using four main metrics: legal consistency, legal accuracy, depth of reasoning and iterative improvement.

[12]`https://pernice.com/chatgpt-batte-gli-avvocati`

Chapter 3

Proposed Methodology

Chapter 4

Design and implementations

Chapter 5

Evaluation and results

Chapter 6

Conclusions and possible future solutions

Ringraziamenti

Bibliography

[1] Michele Baldassarre, Anna Maria Cuzzi, and Francesco Pio Sarcina. "Didattica e Prompt Engineering: una nuova competenza digitale per i docenti nell era dell Intelligenza Artificiale Generativa". In: *Education Sciences & Society* (2024).

[2] Tom Brown et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[3] Ilias Chalkidis et al. *LeXFiles and LegalLAMA: Facilitating English Multinational Legal Language Model Development*. 2023. arXiv: 2305.07507 [cs.CL]. URL: https://arxiv.org/abs/2305.07507.

[4] Aakanksha Chowdhery et al. *PaLM: Scaling Language Modeling with Pathways*. 2022. arXiv: 2204.02311 [cs.CL]. URL: https://arxiv.org/abs/2204.02311.

[5] Jiaxi Cui et al. *Chatlaw: A Multi-Agent Collaborative Legal Assistant with Knowledge Graph Enhanced Mixture-of-Experts Large Language Model*. 2024. arXiv: 2306.16092 [cs.CL]. URL: https://arxiv.org/abs/2306.16092.

[6] Haixing Dai et al. *AugGPT: Leveraging ChatGPT for Text Data Augmentation*. 2023. arXiv: 2302.13007 [cs.CL]. URL: https://arxiv.org/abs/2302.13007.

[7] DeepSeek-AI et al. *DeepSeek-V3 Technical Report*. 2025. arXiv: 2412.19437 [cs.CL]. URL: https://arxiv.org/abs/2412.19437.

[8] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL]. URL: https://arxiv.org/abs/1810.04805.

[9] Wensheng Gan et al. *Mixture of Experts (MoE): A Big Data Perspective*. 2025. arXiv: 2501.16352 [cs.LG]. URL: https://arxiv.org/abs/2501.16352.

[10] Ben Goertzel. "Artificial General Intelligence: Concept, State of the Art, and Future Prospects". In: *Journal of Artificial General Intelligence* 0 (Jan. 2014).
DOI: 10.2478/jagi-2014-0001.

[11] Andrzej Grzybowski, Katarzyna Pawlikowska–Łagód, and W. Clark Lambert. "A History of Artificial Intelligence". In: *Clinics in Dermatology* 42.3 (2024). Dermatology and Artificial Intelligence, pp. 221–229.
ISSN: 0738-081X.
DOI: https://doi.org/10.1016/j.clindermatol.2023.12.016.
URL: https://www.sciencedirect.com/science/article/pii/S0738081X23002687.

[12] Tony Hak and Jan Dul. "Pattern Matching". In: *Erasmus Research Institute of Management (ERIM), ERIM is the joint research institute of the Rotterdam School of Management, Erasmus University and the Erasmus School of Economics (ESE) at Erasmus Uni, Research Paper* (Jan. 2009).

[13] George Hannah et al. *A Prompt Engineering Approach and a Knowledge Graph based Framework for Tackling Legal Implications of Large Language Model Answers*. 2024. arXiv: 2410.15064 [cs.AI]. URL: https://arxiv.org/abs/2410.15064.

[14] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780.
DOI: 10.1162/neco.1997.9.8.1735.

[15] Lei Huang et al. "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions". In: *ACM Transactions on Information Systems* 43.2 (Jan. 2025), 1–55.
ISSN: 1558-2868.
DOI: 10.1145/3703155. URL: http://dx.doi.org/10.1145/3703155.

[16] Aditya Jain, Gandhar Kulkarni, and Vraj Shah. "Natural Language Processing". In: *International Journal of Computer Sciences and Engineering* 6 (Jan. 2018), pp. 161–167.
DOI: 10.26438/ijcse/v6i1.161167.

[17] Albert Q. Jiang et al. *Mistral 7B*. 2023. arXiv: 2310.06825 [cs.CL]. URL: https://arxiv.org/abs/2310.06825.

[18] Albert Q. Jiang et al. *Mixtral of Experts*. 2024. arXiv: 2401.04088 [cs.LG]. URL: https://arxiv.org/abs/2401.04088.

[19] Philip C. Jackson Jr. *Introduction to Artificial Intelligence*. Accessed: 2025-04-02. Dover Publications, 1985. URL: https://ia801605.us.archive.org/view_archive.php?archive=/27/items/general-reader-1/General%20reader1.zip&file=Introduction%20to%20Artificial%20Intelligence%20%28%20PDFDrive%20%29.pdf.

[20] Katikapalli Subramanyam Kalyan. *A Survey of GPT-3 Family Large Language Models Including ChatGPT and GPT-4*. 2023. arXiv: 2310.12321 [cs.CL]. URL: https://arxiv.org/abs/2310.12321.

[21] Katikapalli Subramanyam Kalyan. "A survey of GPT-3 family large language models including ChatGPT and GPT-4". In: *Natural Language Processing Journal* 6 (2024), p. 100048.
ISSN: 2949-7191.
DOI: https://doi.org/10.1016/j.nlp.2023.100048. URL: https://www.sciencedirect.com/science/article/pii/S2949719123000456.

[22] Hyuhng Joon Kim et al. *Self-Generated In-Context Learning: Leveraging Auto-regressive Language Models as a Demonstration Generator*. 2022. arXiv: 2206.08082 [cs.CL]. URL: https://arxiv.org/abs/2206.08082.

[23] Aobo Kong et al. *Better Zero-Shot Reasoning with Role-Play Prompting*. 2024. arXiv: 2308.07702 [cs.CL]. URL: https://arxiv.org/abs/2308.07702.

[24] Bishal Lamichhane. *Evaluation of ChatGPT for NLP-based Mental Health Applications*. 2023. arXiv: 2303.15727 [cs.CL]. URL: https://arxiv.org/abs/2303.15727.

[25] Daniel Lee and Edward Palmer. "Prompt engineering in higher education: A systematic review to help inform curricula". In: *International Journal of Educational Technology in Higher Education* 22.7 (2025).
DOI: 10.1186/s41239-025-00503-7. URL: https://doi.org/10.1186/s41239-025-00503-7.

[26] Mike Lewis et al. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, July 2020, pp. 7871–7880.
DOI: 10.18653/v1/2020.acl-main.703. URL: https://aclanthology.org/2020.acl-main.703/.

[27] Patrick Lewis et al. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. 2021. arXiv: 2005.11401 [cs.CL]. URL: https://arxiv.org/abs/2005.11401.

[28] Cheng Li et al. "EmotionPrompt: Leveraging Psychology for Large Language Models Enhancement via Emotional Stimulus". In: *arXiv preprint arXiv:2307.11760v3* (2023). arXiv: 2307.11760v3 [cs].

[29] Tianyang Lin et al. "A Survey of Transformers". In: *arXiv preprint arXiv:2004.03705* (2020). Accessed: 2025-04-02. URL: https://arxiv.org/pdf/2004.03705.

[30] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692 [cs.CL]. URL: https://arxiv.org/abs/1907.11692.

[31] John McCarthy. *What is Artificial Intelligence?* Accessed: 2025-04-02. 2007. URL: https://www-formal.stanford.edu/jmc/whatisai/.

[32] Tomas Mikolov et al. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: 1301.3781 [cs.CL]. URL: https://arxiv.org/abs/1301.3781.

[33] Keiron O'Shea and Ryan Nash. *An Introduction to Convolutional Neural Networks*. 2015. arXiv: 1511.08458 [cs.NE]. URL: https://arxiv.org/abs/1511.08458.

[34] Kalyani Pakhale. *Comprehensive Overview of Named Entity Recognition: Models, Domain-Specific Applications and Challenges*. 2023. arXiv: 2309.14084 [cs.CL]. URL: https://arxiv.org/abs/2309.14084.

[35] Chatbot Parry. *Parry Chatbot*. Accessed: 2025-04-02. 2025. URL: https://archive.org/details/parry_chatbot.

[36] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. *On the difficulty of training Recurrent Neural Networks*. 2013. arXiv: 1211.5063 [cs.LG]. URL: https://arxiv.org/abs/1211.5063.

[37] Jeffrey Pennington, Richard Socher, and Christopher Manning. "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543.
DOI: 10.3115/v1/D14-1162. URL: https://aclanthology.org/D14-1162/.

[38] L. Rabiner and B. Juang. "An introduction to hidden Markov models". In: *IEEE ASSP Magazine* 3.1 (1986), pp. 4–16.
DOI: 10.1109/MASSP.1986.1165342.

[39] Roy Rada. "Artificial intelligence: E. Rich, (McGraw-Hill, New York, 1983); 411 pages, 30.95USD". In: *Artificial Intelligence* 28.1 (1986), pp. 119–121.
ISSN: 0004-3702.
DOI: https://doi.org/10.1016/0004-3702(86)90034-2. URL: https://www.sciencedirect.com/science/article/pii/0004370286900342.

[40] Alec Radford and Karthik Narasimhan. "Improving Language Understanding by Generative Pre-Training". In: 2018. URL: https://api.semanticscholar.org/CorpusID:49313245.

[41] Alec Radford and Karthik Narasimhan. "Improving Language Understanding by Generative Pre-Training". In: 2018. URL: https://api.semanticscholar.org/CorpusID:49313245.

[42] Rafael Rafailov et al. *Direct Preference Optimization: Your Language Model is Secretly a Reward Model*. 2024. arXiv: 2305.18290 [cs.LG]. URL: https://arxiv.org/abs/2305.18290.

[43] Frank Rosenblatt. *The Perceptron: A Perceiving and Recognizing Automaton*. Tech. rep. Accessed: 2025-04-02. Cornell Aeronautical Laboratory, 1957. URL: https://websites.umass.edu/brain-wars/1957-the-birth-of-cognitive-science/the-perceptron-a-perceiving-and-recognizing-automaton/.

[44] César Borja Ruiz. "Classification and Segmentation of Brain Tumor MRI Images Using Convolutional Neural Networks". In: *2023 IEEE International Conference on Engineering Veracruz (ICEV)* (2023), pp. 1–6. URL: https://api.semanticscholar.org/CorpusID:265826911.

[45] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors". In: *Nature* 323 (1986), pp. 533–536. URL: https://api.semanticscholar.org/CorpusID:205001834.

[46] Manish Saraswat. *NLP Pre-trained Models Explained with Examples*. Accessed: 2025-04-02. 2023. URL: https://vitalflux.com/nlp-pre-trained-models-explained-with-examples/.

[47] Robin M. Schmidt. *Recurrent Neural Networks (RNNs): A gentle Introduction and Overview*. 2019. arXiv: 1912.05911 [cs.LG]. URL: https://arxiv.org/abs/1912.05911.

[48] Sander Schulhoff et al. *The Prompt Report: A Systematic Survey of Prompt Engineering Techniques*. 2025. arXiv: 2406.06608 [cs.CL]. URL: https://arxiv.org/abs/2406.06608.

[49] Jonathan Isaac Segal et al. "A multi-scale cognitive interaction model of instrument operations at the Linac Coherent Light Source". In: *Review of Scientific Instruments* 96.1 (Jan. 2025).
ISSN: 1089-7623.
DOI: 10.1063/5.0239302. URL: http://dx.doi.org/10.1063/5.0239302.

[50] C. E. Shannon. "A Mathematical Theory of Communication". In: *Bell System Technical Journal* 27.3 (1948), pp. 379–423.
DOI: https://doi.org/10.1002/j.1538-7305.1948.tb01338.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.1538-7305.1948.tb01338.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.1538-7305.1948.tb01338.x.

[51] Chenglei Si et al. *Getting MoRE out of Mixture of Language Model Reasoning Experts*. 2023. arXiv: 2305.14628 [cs.CL]. URL: https://arxiv.org/abs/2305.14628.

[52] Sonish Sivarajkumar and Yanshan Wang. *HealthPrompt: A Zero-shot Learning Paradigm for Clinical Natural Language Processing*. 2022. arXiv: 2203.05061 [cs.CL]. URL: https://arxiv.org/abs/2203.05061.

[53] Xinying Song et al. *Fast WordPiece Tokenization*. 2021. arXiv: 2012.15524 [cs.CL]. URL: https://arxiv.org/abs/2012.15524.

[54] Gemini Team et al. *Gemini: A Family of Highly Capable Multimodal Models*. 2024. arXiv: 2312.11805 [cs.CL]. URL: https://arxiv.org/abs/2312.11805.

[55] Amirhosein Toosi et al. "A Brief History of AI: How to Prevent Another Winter (A Critical Review)". In: *PET Clinics* 16.4 (Oct. 2021), 449–469.
ISSN: 1556-8598.
DOI: 10.1016/j.cpet.2021.07.001. URL: http://dx.doi.org/10.1016/j.cpet.2021.07.001.

[56] Hugo Touvron et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: 2302.13971 [cs.CL]. URL: https://arxiv.org/abs/2302.13971.

[57] Shivam Tuli and Shivam Tuli. "Large Language Models Encode Clinical Knowledge". In: *Academia.edu* (2023). URL: https://www.academia.edu/98807332/Large_Language_Models_Encode_Clinical_Knowledge.

[58] Alan M. Turing. "Computing Machinery and Intelligence". In: *Mind* LIX.236 (1950). Accessed: 2025-04-02, pp. 433–460. DOI: 10.1093/mind/LIX.236.433. URL: https://www.cs.mcgill.ca/~dprecup/courses/AI/Materials/turing1950.pdf.

[59] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[60] Jiaqi Wang et al. "Prompt Engineering for Healthcare: Methodologies and Applications". In: *Journal of LaTeX Class Files* 14.8 (2021).

[61] Xuezhi Wang et al. *Self-Consistency Improves Chain of Thought Reasoning in Language Models*. 2023. arXiv: 2203.11171 [cs.CL]. URL: https://arxiv.org/abs/2203.11171.

[62] Jason Wei et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. 2023. arXiv: 2201.11903 [cs.CL]. URL: https://arxiv.org/abs/2201.11903.

[63] Terry Winograd. *SHRDLU: A Program for Understanding Natural Language*. Accessed: 2025-04-02. 2025. URL: https://hci.stanford.edu/winograd/shrdlu/.

[64] David James Woo et al. *Effects of a Prompt Engineering Intervention on Undergraduate Students' AI Self-Efficacy, AI Knowledge and Prompt Engineering Ability: A Mixed Methods Study*. 2024. arXiv: 2408.07302 [cs.CY]. URL: https://arxiv.org/abs/2408.07302.

[65] Benfeng Xu et al. *kNN Prompting: Beyond-Context Learning with Calibration-Free Nearest Neighbor Inference*. 2023. arXiv: 2303.13824 [cs.CL]. URL: https://arxiv.org/abs/2303.13824.

[66] Shunyu Yao et al. *Tree of Thoughts: Deliberate Problem Solving with Large Language Models*. 2023. arXiv: 2305.10601 [cs.CL]. URL: https://arxiv.org/abs/2305.10601.

[67] Shukang Yin et al. "A survey on multimodal large language models". In: *National Science Review* 11.12 (Nov. 2024).
ISSN: 2053-714X.
DOI: 10.1093/nsr/nwae403. URL: http://dx.doi.org/10.1093/nsr/nwae403.

[68] Wayne Xin Zhao et al. *A Survey of Large Language Models*. 2025. arXiv: 2303.18223 [cs.CL]. URL: https://arxiv.org/abs/2303.18223.

[69] Huaixiu Steven Zheng et al. *Take a Step Back: Evoking Reasoning via Abstraction in Large Language Models*. 2024. arXiv: 2310.06117 [cs.LG]. URL: https://arxiv.org/abs/2310.06117.