UNIVERSITAT DE BARCELONA

FUNDAMENTAL PRINCIPLES OF DATA SCIENCE MASTER'S THESIS

Education with Language Models: Analyzing Uncertainty Estimation Techniques

Author:
Dafni TZIAKOURI

Supervisor:
Dr. Jordi VITRIÀ

A thesis submitted in partial fulfillment of the requirements for the degree of MSc in Fundamental Principles of Data Science

in the

Facultat de Matemàtiques i Informàtica

July 6, 2024

UNIVERSITAT DE BARCELONA

Abstract

Facultat de Matemàtiques i Informàtica

MSc

Education with Language Models: Analyzing Uncertainty Estimation Techniques

by Dafni TZIAKOURI

The widespread adoption of Large Language Models (LLMs) underscores the significance of recognizing both their capabilities and constraints. This study aims to delve into understanding the functioning of Large Language Models (LLMs), with a specific focus on GPT models (Sai, 2023), such as GPT-3.5 (Koubaa, 2023) and GPT-4 (OpenAI, 2023). Additionally, it will demonstrate the development of a Chatbot tailored for educational purposes, employing a diverse array of tools. Through systematic examination, this study seeks to determine whether the utilization of LLMs and GenAI can be deemed trustworthy for educational purposes. Moreover, this research will address the challenge of uncertainty estimation, particularly in black-box models, highlighting the need for reliable methods to evaluate model confidence. The investigation will incorporate various experiments designed to evaluate the stability and accuracy of these models. Through comprehensive experimentation, this study seeks to contribute to a deeper understanding of LLMs' behavior, their potential applications in education, and the challenges associated with uncertainty estimation in black-box models. The corresponding notebooks and datasets for this thesis, can be found in the following GitHub repository, `https://github.com/DaphneDjiakouri/MasterThesis`.

Acknowledgements

I am grateful to the Universitat de Barcelona for providing me with the opportunity to work on a project of this nature. Special thanks are extended to my supervisor, Jordi Vitrià, for his exceptional guidance, unwavering support, and endless patience throughout this journey.

Contents

Chapter 1

Introduction

In recent years, Large Language Models (LLMs) have received significant attention both in academic circles and industry, evidenced by the proliferation of open-source versions from various organizations and companies. These LLMs demonstrate proficiency across diverse tasks such as question answering, document summarization, and dialogue systems. Moreover, the integration of Artificial Intelligence (AI) into education marks a transformative era, introducing tools and methodologies to enhance teaching and learning processes. Chatbots and AI-driven systems emerge as key solutions to address educational challenges, particularly in resource-constrained environments, offering scalable and cost-efficient learning avenues.

More specifically, Generative AI technologies, like Large Language Model-based ChatGPT, which, with their sophisticated linguistic capabilities, hold the promise of fostering enriched and interactive real-time learning experiences through natural conversation. These advancements underscore AI's potential to revolutionize education, making it more personalized, accessible, and effective for learners across diverse spectra. Manifestations of this potential include rapid query resolution, tailored explanations based on predefined criteria, and recommendations for supplementary teaching and learning materials.

This article adopts a pragmatic approach, detailing the development of a question-answering system aimed at creating an educational tutor benefiting both students and educators. Operating as a chatbot, this tutor facilitates student inquiries by parsing uploaded PDF or Text files related to selected subjects. The methodology involves elucidating semantic search and querying techniques on PDF or Text files using a combination of OpenAI, LangChain, and an external vector store, Chroma. The PDF/Text file is broken down into smaller documents, and OpenAI embeddings are used to convert them into vectors, which are then stored externally using Chroma. Leveraging the capabilities of Large Language Models (LLMs) such as GPT-3.5 and GPT-4 by OpenAI, the chatbot furnishes detailed explanations and information drawn from the uploaded documents.

This educational tutor serves a dual purpose, aiming to help both students and educators. Students can conveniently select any school subject and upload corresponding materials into the chatbot, facilitating their understanding and study process. By posing targeted questions on areas they find challenging or wish to explore further, students can streamline their home study efforts, gaining clearer insights and enhancing their learning experience with a reliable study companion. This interactive learning approach is tailored to resonate with modern students, fostering greater self-directed learning and yielding improved academic outcomes. For educators, the benefits are equally significant. They gain insight into students' inquiries, allowing for better monitoring of student engagement. Additionally, educators can identify common areas of difficulty through the questions posed, enabling them to adjust their teaching focus accordingly. Moreover, they can request the chatbot to generate sample test questions for specific sections of the material, aiding in the creation of comprehensive assessments. Thus, this educational tool serves as a valuable asset in the daily lives of students and professors alike, enhancing the educational journey by fostering a more engaging, accessible, and insightful learning environment.

Even though Generative AI showcases impressive capabilities, it presents challenges that remain not fully understood. Large language models, for instance, are prone to what's known as "hallucinations," where they generate false information that seems real. Moreover, LLM-generated responses can be incorrect and misleading, potentially spreading misinformation. Additionally, they present important security risks when not managed or monitored properly. They can leak private information, participate in phishing scams, and produce spam. Users with malicious intent can reprogram AI to their ideologies or biases, and contribute to the spread of misinformation. The repercussions can be devastating on a global scale. Another limitation is that many generative AI tools work offline, limiting their access to current information. For instance, popular models like ChatGPT may provide outdated or inaccurate answers due to their training data having specific cutoff dates, which users might not always be aware of.

Hence, it is crucial to develop strong methods to evaluate their performance. However, thoroughly evaluating LLMs is a challenging task. To address this challenge, several open leaderboards like the well-known HuggingFace open LLM leaderboard and Chatbot Arena (Hu et al., 2023) have emerged, providing a comparison of LLM performance. Despite their usefulness, these leaderboards lack consideration for LLM uncertainty. For instance, while the HuggingFace open LLM leaderboard mainly focuses on accuracy as the evaluation metric, it overlooks differences in uncertainty among LLMs. Therefore, integrating uncertainty into the evaluation process is vital for a more comprehensive assessment of LLM effectiveness.

Chapter 2

Background

2.1 How Do Generative AI Models Work?

Generative AI (GenAI) encompasses a range of artificial intelligence models capable of creating diverse content, including text, code, images, video, and music. Within this domain, Large Language Models stand out, trained extensively on textual data to generate coherent text. ChatGPT is a popular example of generative text AI.

The underlying technologies powering GenAI belong to the realm of machine learning (ML), a field leveraging algorithms to iteratively enhance performance based on data. Among the various ML techniques, artificial neural networks (ANNs) have notably propelled AI advancements, exemplified by applications like facial recognition. ANNs draw inspiration from the intricate synaptic connections within the human brain, manifesting in diverse architectures.

Text generative AI relies on a specific class of ANN known as a general-purpose transformer, with a subtype termed a Large Language Model (LLM). Consequently, systems like AI Text GenAI are commonly referred to as LLMs, with the chosen variant being the generative pre-trained transformer (GPT), as evidenced in ChatGPT's architecture. Thus, gaining an understanding of Large Language Models is pivotal before delving into specific models like GPT.

2.1.1 Understanding Large Language Models

A Large Language Model (LLM) is a deep learning algorithm that can perform a variety of natural language processing (NLP) tasks. Large Language Models utilize transformer architectures (refer to section 2.1.2) and are trained using extensive datasets, hence the designation "large." This capacity allows them to comprehend, translate, forecast, or produce text and various other content types. They are also referred to as neural networks (NNs), which are computing systems inspired by the human brain. These neural networks work using a network of nodes that are layered, much like neurons.

To achieve proficiency, Large Language Models must be pre-trained and then fine-tuned to solve text classification, question answering, document summarization, and text generation problems. Moreover, they have large numbers of parameters, which are akin to memories the model collects as it learns from training.

Large Language Models are composed of multiple neural network layers. Recurrent layers, feedforward layers, embedding layers, and attention layers work in tandem to process the input text and generate output content. The following is a brief explanation of the various layers previously discussed:

- **Embedding layers** within Large Language Models construct embeddings from input text, capturing both the semantic and syntactic meaning of the input to facilitate contextual understanding.
- The **feedforward layer (FFN)** is comprised of multiple fully connected layers tasked with transforming input embeddings, thereby facilitating the extraction of higher-level abstractions and understanding user intent within the text.
- **The recurrent layer** interprets the words in the input text in sequence, capturing the relationship between words in a sentence.
- **The attention mechanism** enables a language model to focus on specific parts of the input text that are relevant to the task at hand. This layer allows the model to generate the most accurate outputs.

Large Language Models typically fall into three categories:

1. **Generic or raw** language models predict subsequent words based on the language present in the training data, primarily serving information retrieval purposes. This is illustrated in Figure 2.1.
2. **Instruction-tuned** language models are trained to predict responses aligned with given instructions, facilitating tasks like sentiment analysis or text and code generation.
3. **Dialog-tuned** language models are specialized in conducting dialogues by predicting subsequent responses, commonly utilized in chatbots or conversational AI systems.

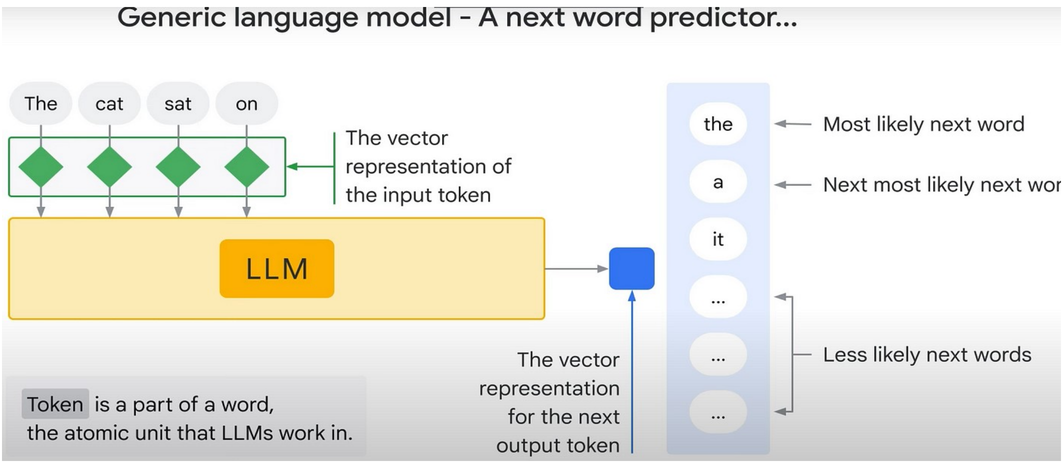

FIGURE 2.1: Example of Generic of Raw LM. Image Source: Google Cloud Tech

How Do Large Language Models Work?

A Large Language Model is based on a transformer (Vaswani et al., 2023) model and works by receiving an input, encoding it, and then decoding it to produce an output prediction. Before a Large Language Model can receive text input and generate an output prediction, it requires training, so that it can fulfill general functions, and fine-tuning, which enables it to perform specific tasks.

Training:

Large Language Models are pre-trained using large textual datasets from sites like Wikipedia, GitHub, or others. These datasets consist of trillions of words, and their quality affects the language model's performance. At this stage, the Large Language Model engages in unsupervised learning, meaning it processes the datasets fed to it without specific instructions. During this process, the LLM's AI algorithm can learn the meaning of words, and the relationships between words. It also learns to distinguish words based on context. For example, it would learn to understand whether "right" means "correct," or the opposite of "left."

Fine-tuning:

In order for a Large Language Model to perform a specific task, such as translation, it must be fine-tuned to that particular activity. Fine-tuning optimizes the performance of specific tasks.

Prompt-tuning:

This process trains a model to perform a specific task through few-shot prompting or zero-shot prompting. A prompt is a natural language request submitted to a language model to receive a response. They can take various forms, such as questions, instructions, contextual information, examples or partial input. Few-shot prompting teaches the model to predict outputs through the use of examples. Alternatively, zero-shot prompting does not use examples to teach the language model how to respond to inputs.

2.1.2 Transformer Architecture

A transformer (Vaswani et al., 2023) model is the most common architecture of a Large Language Model. It consists of an encoder and a decoder. The encoder processes input sequences, and the decoder converts the output of the encoder to the generated sequence at the output of the transformer. A transformer model processes data by tokenizing the input, then simultaneously conducting mathematical equations to discover relationships between tokens. This enables the computer to see the patterns a human would see were it given the same query.

Transformer models work with self-attention mechanisms, which enable the model to learn more quickly than traditional models like long short-term memory models. Self-attention (Shaw, Uszkoreit, and Vaswani, 2018) is what enables the transformer model to consider different parts of the sequence, or the entire context of a sentence, to generate predictions.

2.1.3 Operational Workflow of GPT Models

In this study, the primary focus is on dialog-tuned language models, since models such as GPT-3.5 (Koubaa, 2023) and GPT-4 (OpenAI, 2023) were used to create the Educational Tutor (chatbot). The aim is to provide a comprehensive understanding of these models and their application in the educational framework.

Developed by OpenAI, ChatGPT stems from the GPT-3 (Koubaa, 2023) iteration, marking a significant milestone in AI evolution. OpenAI's successive iterations, including the most recent GPT-4 in March 2023 (OpenAI, 2023), have consistently enhanced performance through advancements in architecture, training methodologies, and optimization strategies. A noteworthy aspect of this progress is the continual expansion of the model's 'parameters', akin to metaphorical knobs fine-tuning its functionality. These parameters encompass the model's 'weights', numerical entities dictating input processing and output generation.

The GPT-3 falls into the category of Reinforcement Learning with Human Feedback (RLHF) where GPT-4 introduced a rule-based reward model (RBRM) approach. It was reported in the GPT-4 Technical Report (OpenAI, 2023): *"Our rule-based reward models (RBRMs) are a set of zero-shot GPT-4 classifiers. These classifiers provide an additional reward signal to the GPT-4 policy model during RLHF fine-tuning, targeting correct behavior, such as refusing to generate harmful content or not refusing innocuous requests."*

The Rule-Based Reward Models (RBRMs) approach improves language models' performance and safety, like GPT-4. It provides additional reward signals during the Reinforcement Learning from Human Feedback (RLHF) fine-tuning process on the generated text to ensure its compliance with generating safe and correct content. Since OpenAI has not released a detailed technical report on GPT-4, the focus will be more on the architecture of GPT-3.

Architecture

The main architecture of this model is similar to the traditional transformer with an encoder, decoder, and attention layers. However, the number of each entity present in it is higher than any other transformer ever created. GPT-3 comes in 8 different sizes: GPT-3 small, medium, large, XL. The transformer architecture is illustrated in Figure 2.2.

The smallest GPT-3 is similar to the BERT (HuggingFace, n.d.(a)) in terms of architecture and has 12 attention layers each with 64-dimensional heads (12x64). The model GPT-3 or GPT-3 175B has 175 billion trainable parameters with 96 attention layers and the dimensions used here are 128 (96x128).

Training

GPT-3 implements unsupervised training and is trained on next word prediction which means it predicts the next words based on the input tokens provided. The parameters present in the models extract the relationship between the input tokens or words. It employs "Semantic Analysis"[1] to understand not just the words and their meanings, but also how usage of words differs depending on other words.

To train models of different sizes, the learning rate is reduced and the batch size is increased. GPT-3 with 125 million parameters has a batch size of 0.5 million and a learning rate of 6×10^{-4}, while the one with 175 billion parameters has a batch size of 3.2 million and a learning rate of 0.6×10^{-4}. However, the core architecture of GPT-3 is not fully revealed by OpenAI.

[1] *Semantic Analysis: This is a main driving mechanism in Natural Language Processing. It refers to a process of extracting the main ideas like context, tone, meaning from the given unstructured data. Semantic Analysis-driven tools in any organization can extract the context in emails, support tickets, customer feedback.*

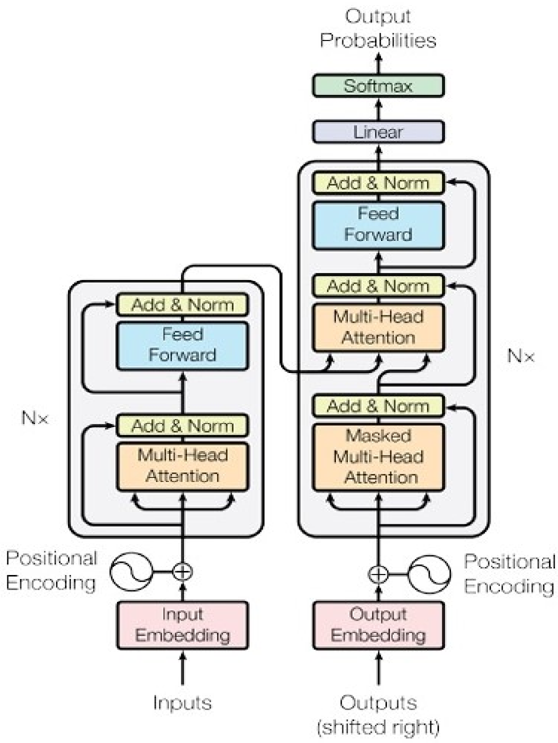

FIGURE 2.2: GPT-3 Architecture. Image source: Sai, 2023

Working

GPT-3 is trained using a reinforcement learning algorithm, so every time it makes a mistake or repeats the same word multiple times, it penalizes itself. OpenAI released the GPT-3 API in early 2021, allowing developers or users to leverage it through the API.

As mentioned earlier, GPT-3 is a few-shot and multitask system. Let's discuss what few-shot and multitask systems are.

A neural network is said to be zero/one/few-shot depending on the sets of classes it identifies after training. If a network is trained on limited classes and, while testing, the model only classifies (e.g., object detection) the data based solely on the training data, then it is zero-shot. If it can identify data from one or more new classes, then the model is called one-shot or few-shot.

GPT-3 has two main features:

1. **Temperature:** This feature describes how creative the model should be while generating text. If the temperature is set to 'low', the model is less creative, and vice versa.

2. **Presets:** Presets are prewritten prompts that inform the model about the user's request. Every time a user provides a preset, the model knows exactly what to generate. For example: chats, summarizing, text to code, language translation.

Another feature that makes the model very powerful and the best so far in the AI industry is its ability to meta-learn a new task. We can think of the user prompt as a new program, as it not only makes the model generate text but also helps in training the model with new data. GPT-3 is trained on a huge amount of data so that it has no choice but to learn higher-level ways of manipulating the language. Every time a user provides a prompt to learn a new task, pretrained weights of the network don't change, but the input text (prompt) is transformed into complex abstractions.

Following the completion of training, generating a text response with GPT involves several sequential stages:

1. The prompt is broken down into smaller units (known as tokens) that are inputted into the GPT.
2. Leveraging statistical patterns, the GPT anticipates probable words or phrases to construct a coherent response:
   (a) By discerning common word and phrase correlations within its extensive precompiled dataset, sourced from diverse online and offline text repositories.
   (b) Utilizing these established patterns, the GPT calculates the likelihood of specific words or phrases occurring within a given context.
   (c) Beginning with a random prediction, the GPT uses these estimated probabilities to forecast the subsequent probable word or phrase in the response.
3. The predicted elements are transformed into readable text.
4. The readable text is filtered through what are known as 'guardrails' to remove any offensive content.
5. Steps 2 to 4 are repeated until a response is finished. The response is considered finished when it reaches a maximum token limit or meets predefined stopping criteria.
6. The response is post-processed to improve readability by applying formatting, punctuation, and other enhancements (such as beginning the response with words that a human might use, such as 'Sure', 'Certainly' or 'I'm sorry').

Now that, the operation and categorization of Large Language Models, including various types of GPT models, were thoroughly examined. It is essential to highlight some of the primary limitations inherent in these models.

2.2 Limitations of GenAI Models

In this section, are pointed out some of the main limitations of GenAI in more depth:

Incapable of Autonomous Learning: Generative models operate within a fixed dataset and lack the ability to acquire new knowledge independently. While they can integrate external inputs for context, they remain static in terms of their fundamental knowledge base. While domain-specific knowledge and industry context can be provided, adapting these models to individual business environments necessitates specialized training.

Hallucinations: LLM hallucinations are the events in which ML models, produce outputs that are coherent and grammatically correct but factually incorrect or nonsensical. "Hallucinations" in this context means the generation of false or misleading information. These hallucinations can occur due to various factors, such as limitations in training data, biases in the model, or the inherent complexity of language.

For example, when asked to the ChatGPT (OpenAI, n.d.) "How many 'm's are in the word 'Elephant'?" and the Chat answer with "There is one 'm' in the word 'Elephant'", that is an hallucination.

Inability to Replicate Human Traits: Essential human attributes such as creativity, emotional intelligence, and proactive learning are beyond the reach of generative models. They are unable to comprehend human thoughts, emotions, or conceive novel ideas. Presently, they fall far short of replacing humans in tasks where these traits are indispensable. Although configuration options exist, like adjusting the 'temperature' parameter in systems such as ChatGPT to enhance creativity, this often results in increased inaccuracies and occasional generation of entirely fabricated information.

Challenges in Source Citation: Generative models lack a structured information storage mechanism conducive to citation. As a consequence, substantiating work derived from tools like ChatGPT poses difficulties. Professionals must invest additional effort in documenting data sources, lest reliance on unverified information becomes problematic. While some resources offer guidance on integrating citation functionality, success is not guaranteed, with instances of fake citations or omission of crucial sources being prevalent.

Bias, Discrimination and Stereotype: Biases in LLM training data perpetuate harmful cultural and gender stereotypes, reinforcing societal prejudices and hindering progress. This issue exacerbates cultural divides and gender disparities, leading to discrimination based on sex, ethnicity, age, or disability due to underrepresented training data. Biased LLM outputs affect marginalized communities, influencing hiring processes and educational opportunities, and raise ethical concerns in decision-making. For instance, when asked for a famous physician, ChatGPT might only name a male, highlighting the lack of diversity in responses.

Absence of Certainty: GenAI's responses lack reassurance and cannot be definitively verified, notwithstanding their apparent confidence. Operating on probability-based word predictions, they yield grammatically coherent yet potentially inaccurate outputs. Professionals are advised against assuming the validity of information provided by these models without scrutiny.

Challenge in Identifying Artificial Content: Advancements in generative technologies blur the boundary between human-generated and AI-generated content, rendering the distinction increasingly elusive. While tools for detecting AI-generated content exist, they are far from foolproof. Consequently, the convergence of artificial and genuine content fosters an environment conducive to the proliferation of misinformation, erroneous insights, and related societal issues.

In summary, generative models have several key limitations. This project will focus on estimating the uncertainty of the answers provided by LLMs and gaining a deeper understanding of the mathematical methods and techniques for this estimation. Enhancing control over LLMs and improving comprehension of their responses are anticipated outcomes. Therefore, the next section will describe the uncertainty estimation from a mathematical perspective.

2.3 The problem of uncertainty in technical terms

2.3.1 Definition

Uncertainty Estimation (UE) is defined as the process of quantifying the degree of confidence in the predictions made by a machine learning model. Research on Uncertainty Estimation (UE) methods for LLMs has mainly focused on theoretical aspects rather than practical applications. Additionally, much of the existing literature assumes that users can access LLMs' inner workings, which is often not the case due to proprietary software or computational limits. For instance, many existing LLMs are black-boxes[2] served via APIs[3], implying that end-users typically do not have full access to how they work. A reliable way to measure uncertainty is crucial for knowing when to trust a model. When a model shows high uncertainty or returns low-confidence predictions, the input should either be rejected or subjected to further evaluation (Yarin Gal, 2016). Therefore, accurately quantifying uncertainty in LLMs is vital, especially for tasks like question-answering (QA).

In this work, this gap is addressed by implementing simple yet effective UE methods for black-box LLMs in text generation tasks. UE in tasks like text generation presents a complex challenge. To quantify the uncertainty of the entire sequence, it necessitates aggregating uncertainties of numerous individual token predictions, along with addressing complex sampling and pruning techniques like beam search. Unlike finite prediction options in classification tasks, text generation involves an infinite or immensely large pool of potential predictions, making probability estimation and scoring complex. Moreover, natural language texts embody nuanced interplays of context, semantics, and grammar, leading to diverse surface forms with similar meanings, a factor that warrants consideration during the UE process.

2.3.2 Methodology and Estimators

Various approaches for estimating uncertainty in black-box LLMs are explored in this segment. Several of these techniques are sourced from prior research that has demonstrated effectiveness in different contexts (Fomicheva et al., 2020). Specifically, focus is placed on Lexical Similarity, Number of Semantic Set, Graph Laplacian Eigenvalue Sum, Degree Matrix, and Eccentricity. These methods can be organized into the following procedural framework:

1. For a given input x, generate K response samples $y_1, \ldots, y_K$.
2. Compute a $K \times K$ similarity matrix S between responses, where $S_{ij} = s(y_i, y_j)$ for some similarity score s (Natural Language Inference score or Jaccard score).
3. Based on the similarity matrix S, compute the final uncertainty score.

Hence, the methodology revolves around examining the similarity matrix and aggregating the data to calculate the uncertainty score.

[2]Black-box models operate without revealing their internal processes, focusing solely on the input-output relationships.

[3]API is the acronym for application programming interface — a software intermediary that allows two applications to talk to each other. APIs are an accessible way to extract and share data within and across organizations.

Measuring Answer Similarities:

Following the work of Lin et al. (2023) (Lin, Trivedi, and Sun, 2024), two primary methods are used to compare the similarity between pairs of responses.

Jaccard Similarity: The Jaccard similarity is a fundamental metric used to determine the similarity between two sets. It is calculated by dividing the number of elements in the intersection of the two sets by the number of elements in their union. This straightforward rule-based metric is widely applied in various domains, including Natural Language Processing (NLP) tasks (Lin, Trivedi, and Sun, 2024). In the context of NLP, where sentences or documents are considered as sets of words, the Jaccard similarity measures the similarity between two responses, s_{j1} and s_{j2} (viewed as word sets), where $j_1, j_2 \in [K]$, calculated as follows:

$$\text{Jaccard}(s_{j_1}, s_{j_2}) = \frac{|s_{j_1} \cap s_{j_2}|}{|s_{j_1} \cup s_{j_2}|} \in [0, 1]. \tag{2.1}$$

Despite its efficiency, the Jaccard similarity has limitations, such as ignoring word order and failing to capture nuanced expressions like negation.

Natural Language Inference (NLI): As mentioned earlier, rule-based similarity methods often miss the intricate nuances in generated responses. An alternative approach involves using a Natural Language Inference (NLI) classifier. Based on the work of Kuhn et al. (2023) (Kuhn, Gal, and Farquhar, 2023), the DeBERTa-large model (He et al., 2021) is employed as the classifier. An NLI classifier predicts scores (logits) for three classes: entailment (how much the meaning of a set α implies the meaning of a set β), neutral, and contradiction (how much the meaning of a set α conflicts with the meaning of a set β). For similarity assessment, the predicted probabilities are used, denoted as $a_{\text{NLI}}(s_{j_1}, s_{j_2})$. Applying the softmax function to the predicted logits gives continuous values between 0 and 1, represented as $\hat{p}_{\text{contra}}(s_{j_1}, s_{j_2})$ and $\hat{p}_{\text{entail}}(s_{j_1}, s_{j_2})$ (both depending on x). The following formulations are then defined:

$$a_{\text{NLI,entail}}(s_{j_1}, s_{j_2}) = \hat{p}_{\text{entail}}(s_{j_1}, s_{j_2}) \tag{2.2}$$

$$a_{\text{NLI,contra}}(s_{j_1}, s_{j_2}) = 1 - \hat{p}_{\text{contra}}(s_{j_1}, s_{j_2}). \tag{2.3}$$

It is important to note that computing $\hat{p}_{\text{entail}}$ and $\hat{p}_{\text{contra}}$ aligns with the objective of quantifying the uncertainty of a black-box LLM for two main reasons. First, the NLI model can be significantly smaller than the LLM, given the simpler nature of the NLI task and the reduced knowledge requirements for the NLI model. Second, while the LLM's role in Natural Language Generation (NLG) primarily involves generating responses (sequences of tokens), supplementary information such as token-level logits or embeddings is typically not part of the standard output and may not be accessible to users. In contrast, the NLI model's output consists of probabilities, which are readily utilized for this purpose.

Estimating Uncertainty from Similarities:

This section aims to convert similarities from the previous section into uncertainty measures.

Number of Semantic Sets: Proposed initially by Kuhn et al. (2023) (Kuhn, Gal, and Farquhar, 2023), this method involves using an NLI classifier to categorize responses into various "semantic equivalence" subsets, forming a comprehensive partition of all responses. These "semantic equivalence" classes, along with the numerical outputs from the base LLM, were used to compute "semantic entropy." While this method may not be directly applicable to a black-box LLM, their experiments also explored the concept of the number of "semantic sets" (equivalence classes) as an uncertainty measure for black-box LLMs. This approach involves an iterative process that systematically examines responses, conducting pairwise comparisons indexed as j_1 and j_2, where $j_2 > j_1$. Initially, the number of semantic sets equals the total number of generated answers, denoted as m. If the conditions $\hat{p}_{\text{entail}}(s_{j_1}, s_{j_2}) > \hat{p}_{\text{contra}}(s_{j_1}, s_{j_2})$ and $\hat{p}_{\text{entail}}(s_{j_2}, s_{j_1}) > \hat{p}_{\text{contra}}(s_{j_2}, s_{j_1})$ are met, these two sentences are placed into a single cluster. This uncertainty measure is denoted as U_{NumSet}. For example, considering the question "What city was Zeus the patron god of?", the responses "Olympia", "Zeus was the patron god of Olympia, Greece", and "Corinth" form two semantic sets, with the first two responses belonging to one set. A higher count of semantic sets correlates with an elevated level of uncertainty, indicating a broader range of semantic interpretations for the answer.

In practice, whether two responses share the same meaning is not always clear-cut. In the example of Zeus, responses like "Olympia" and "Greece" are neither exactly the same nor completely different. Furthermore, there is no guarantee that the semantic equivalence judged by the NLI model (or any other measure) is transitive. Therefore, a more nuanced and "continuous" way to measure the number of meanings is preferable (Lin, Trivedi, and Sun, 2024).

Sum of Eigenvalues of the Graph Laplacian: Since only pairwise similarities a_{j_1,j_2} between responses s_{j_1} and s_{j_2} are known, but not the embeddings of the generated responses, a natural choice for clustering responses is spectral clustering. Fixing an input x, each generated response is treated as a node, and the symmetric weighted adjacency matrix $W = (w_{j_1,j_2})$ is defined for $j_1, j_2 = 1, \ldots, K$ where $w_{j_1,j_2} = (a_{j_1,j_2} + a_{j_2,j_1})/2$. The symmetric normalized graph Laplacian is then given by

$$L := I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \tag{2.4}$$

$$D_{j_1,j_2} = \begin{cases} \sum_{j' \in [K]} w_{j_1,j'} & \text{if } j_1 = j_2 \\ 0 & \text{if } j_1 \neq j_2 \end{cases} \tag{2.5}$$

A continuous version of U_{NumSet} can be defined with $\lambda_1 < \ldots < \lambda_K$, the eigenvalues of L:

$$U_{\text{EigV}} = \sum_{k=1}^{K} \max(0, 1 - \lambda_k) \tag{2.6}$$

From both theoretical and practical viewpoints, U_{EigV} is a more flexible approach compared to U_{NumSet}.

The Degree Matrix: The preceding methods share a common drawback: they do not provide uncertainty estimates for individual answers. Lin et al. (2023) (Lin, Trivedi, and Sun, 2024) propose a solution by using the Degree Matrix D computed earlier. They suggest utilizing the total uncertainty of answers, inferred from the Degree Matrix D. The concept is that the aggregate uncertainty of answers can be measured as a corrected trace of the diagonal matrix D, where the diagonal elements represent the sums of similarities between each answer and all others. This effectively represents the average pairwise distance between all answers, with larger values indicating higher uncertainty due to greater disparities between answers. The resulting uncertainty measure is

$$U_{\mathrm{Deg}}(x) = 1 - \mathrm{trace}(D)/K \tag{2.7}$$

Eccentricity: A significant challenge is that only similarity (or distance) metrics between various responses are available, without access to their actual embedding space. However, the graph Laplacian offers a solution by providing the coordinates for these responses. Let $u_1, \ldots, u_k \in \mathbb{R}^K$ denote the smallest k eigenvectors of L. Then, a meaningful embedding of s_j can be represented as $v_j = [u_{1,j}, \ldots, u_{k,j}]$ (Andrew Y. Ng, 2001). Consequently, the average distance from the center can be used as the uncertainty measure, with each response's distance from the center serving as the (negative) confidence indicator. Formally, the "eccentricity" estimates are given by:

$$U_{\mathrm{Ecc}}(x) = \left\| \left[v_1'^{\top}, \ldots, v_K'^{\top} \right] \right\|_2 \tag{2.8}$$

where $v_j' = v_j - \frac{1}{K}\sum_{l=1}^{K} v_l$ represents the offset from the average embedding.

Lexical Similarity: This measure, proposed by Fomicheva et al. (2020) (Fomicheva et al., 2020), computes how similar two words or phrases are in terms of their meaning. Although originally focused on machine translation, this measure calculates the average similarity score between all pairs of translation hypotheses in a set, using a similarity measure based on the overlap of their lexical items. Different metrics can be used, such as ROUGE-1 (Mamdouh, 2023), ROUGE-2 (Mamdouh, 2023), ROUGE-L (Mamdouh, 2023), and BLEU (Mamdouh, 2023). Higher values indicate more uncertain samples.

Chapter 3

Development

3.1 Implementation of Educational Tutor

In this section, a detailed explanation of the implementation of the chatbot (educational tutor) is provided, including the tools employed for its development.

3.1.1 General Architecture

In this implementation, an AI application called LangChain (langchain, 2023) is utilized. To understand its significance, it is essential to first explain this tool and explore its benefits. LangChain is a framework designed for developing applications powered by Large Language Models (LLMs). It enables AI developers to integrate these models with external data sources.

As previously mentioned, GPT models are trained on large amounts of data up to a specific point in time. For instance, GPT-4 has data training up until September 2021, which can be a significant limitation. Although the model possesses extensive knowledge, its understanding of current events or ability to connect with custom data can greatly enhance its functionality.

This is where LangChain becomes crucial. LangChain allows the LLM to access custom data or utilize search engines to generate responses. This capability enables the GPT model to provide up-to-date information from documents, websites, and other sources.

Figure 3.1 illustrates all the necessary packages used in the architecture of the educational tutor (Mishra, 2023). A detailed explanation of these packages will be provided later on.

```
from langchain_community.document_loaders import PyPDFLoader, TextLoader
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain.embeddings.openai import OpenAIEmbeddings
from langchain_community.vectorstores import Chroma
from langchain.chat_models import ChatOpenAI
```

FIGURE 3.1: All necessary installed packages from Langchain

The general pipeline is as follows: A PDF file or text file is selected, depending on the document type the user wishes to search. The document is read using the 'PyPDFLoader' or 'TextLoader' package from LangChain. Next, the document is split into multiple chunks of size 1500 with an overlap of 150. For this, the 'Recursive Character Text Splitter' is used, which divides the text by recursively examining characters and finding suitable split points. After splitting, these chunks are converted into embeddings using 'OpenAI Embeddings' and all the vectors are stored in an external database called 'Chroma'.

Once this process is complete, the user can ask their first question or prompt about the document. The prompt is taken and the vector store is searched for all relevant chunks containing information related to the prompt's context. The necessary chunks, along with the query, are then gathered and passed into a GPT model to generate an answer. Technically, this is achieved using the 'ConversationalRetrievalChain', which builds on 'RetrievalQAChain' and includes a chat history component. It combines the chat history and the query to form a standalone question. Then, it retrieves relevant documents from the retriever and passes those documents, along with the query, to a question-answering chain to generate the response.

Here, the described architecture is illustrated for better understanding (Figure 3.2):

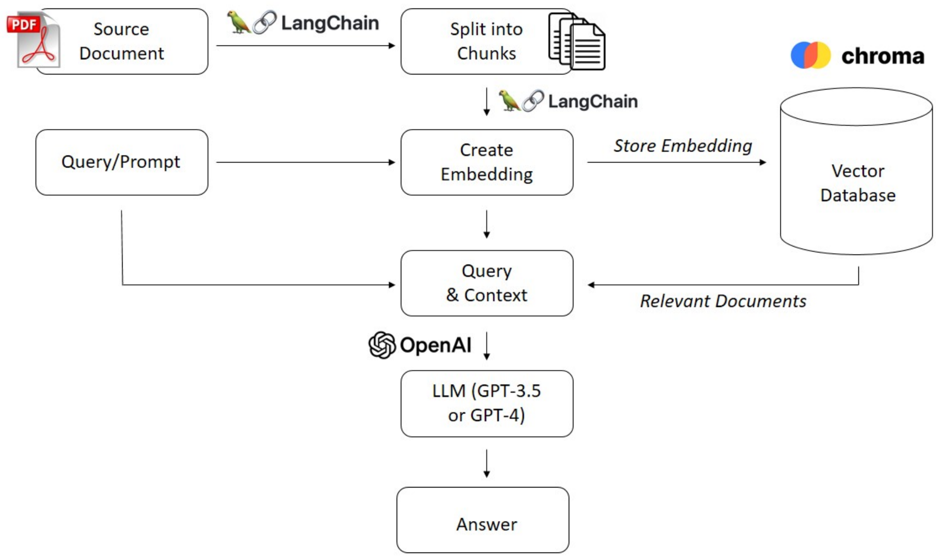

FIGURE 3.2: Illustration of the pipeline

3.1.2 Creating User Interface (UI)

To develop the Educational Tutor (chatbot) and its user interface (UI), an open-source Python package called 'Chainlit' (Chainlit, 2023) was utilized. Chainlit facilitates rapid development and is compatible with all Python programs and libraries. This compatibility makes it particularly convenient for integration with LangChain, which was employed for the chatbot architecture.

For this project, the chatbot was initially tested for potential limitations and areas for improvement using a history book about Cyprus. The PDF, titled "Old to Modern History of Cyprus" (Petrides, 2013), served as the primary resource for this evaluation.

Figure 3.3 shows the general form of the interface, where the user can begin by uploading the desired document—in this case, the Cyprus history book:

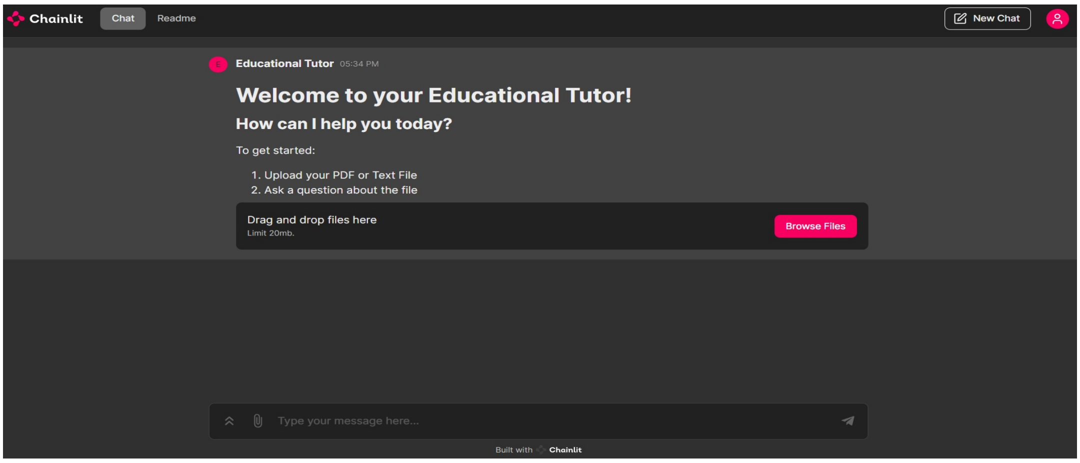

FIGURE 3.3: The main interface

Following the trials of the chatbot, it generally performed well, providing accurate and extensive answers to relevant questions. Additionally, it offers a reference page for users seeking more information. However, a limitation was identified when handling follow-up questions. For example, when asked, "When was the Turkish invasion?" the chatbot responded with "On the 20th of July" without specifying the year. Consequently, when prompted for the year, it failed to provide an answer.

To address this issue, an additional function was implemented to enhance the chatbot's memory, enabling it to answer follow-up questions effectively. This was achieved using 'DocArrayInMemorySearch' from Langchain, which is a document index provided by DocArray that stores documents in memory.

After uploading the desired document, the chatbot processes it according to the previously explained architecture. Once the processing is complete, it informs the user that they can now ask questions. Importantly, when the chatbot cannot provide information about a question, it refrains from generating or hallucinating answers and instead acknowledges its inability to provide the information. This is happening when the model might not find relevant patterns with high similarity or when the confidence score for potential responses is low. In other words, it adds a threshold in the distance between the prompt and the chunks. Examples of these trials are shown in Figures 3.4 and 3.5.

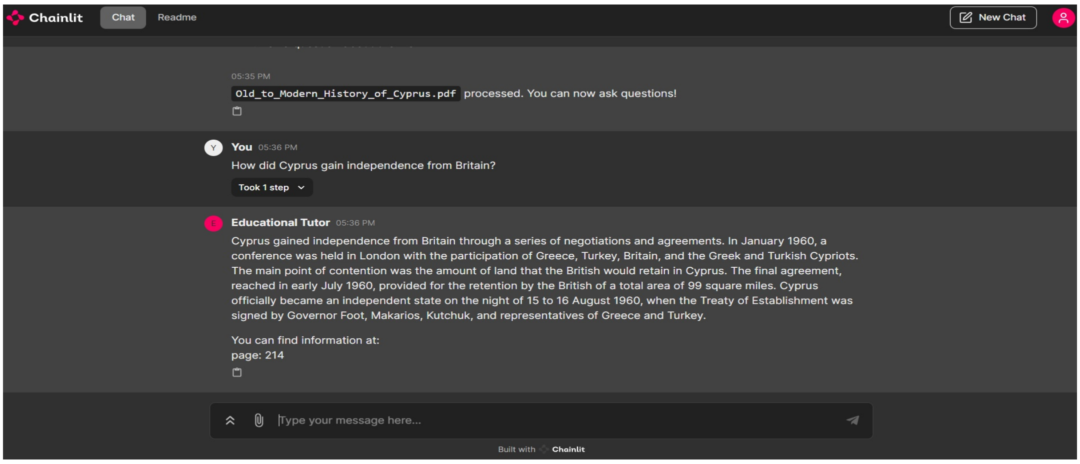

FIGURE 3.4: Example of generating correct answer

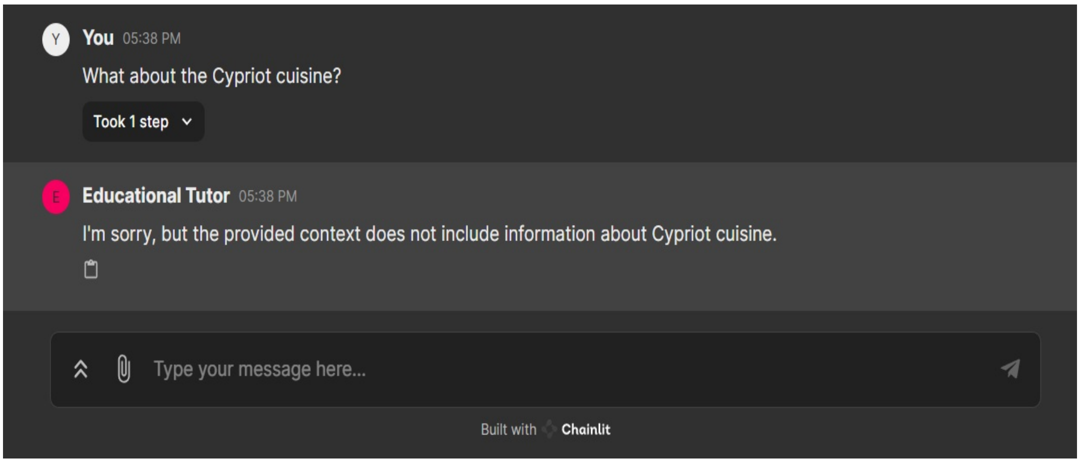

FIGURE 3.5: Example of not providing any information: When asking about the cuisine of Cyprus is not providing any information since there is no relevant chunk from the document

In addition, an important scope of this chatbot was to ensure user-friendliness, as it was intended to function as an educational tutor for students and teachers in schools. Therefore, attention was paid to its appearance, including details such as colors and names, and the streamline of its answers, to enhance interactivity and provide a sense of real conversation, while being easy to manage for all ages.

Furthermore, it was examined whether the chatbot could handle questions in different languages and respond accurately to queries from various school subject books. Several questions were asked in Greek, and the chatbot efficiently provided answers in Greek. This outcome was expected, as the model retrieves information from the document chunks and generates responses using GPT models, which support multiple languages. Additionally, the chatbot was tested with a book on Natural Science (Miah, 2020) to determine if it could correctly handle content involving mathematical symbols and other subject-specific information. Fortunately, it performed well in these scenarios. Examples of these scenarios are showed in Figure 3.6 and Figure 3.7.

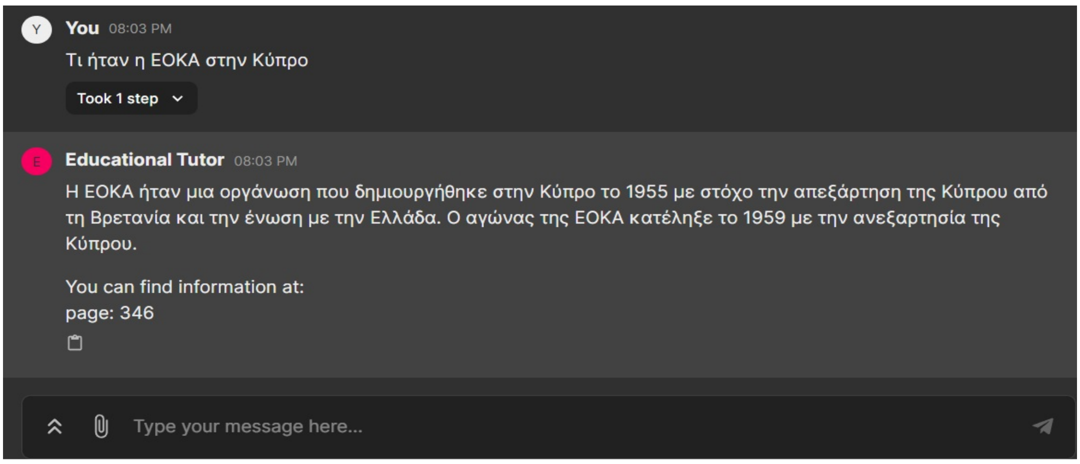

FIGURE 3.6: Example in greek

Therefore, the developed Educational Tutor appears to function effectively in various scenarios and trials, especially after several adjustments were made along the way. Consequently, experiments can now be conducted to investigate the uncertainty of the LLM's answers.

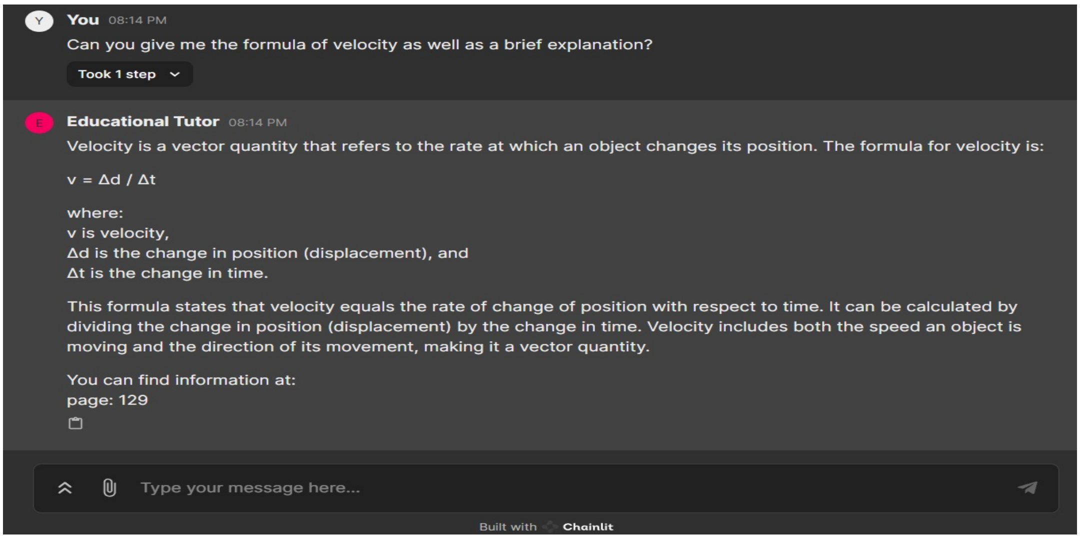

FIGURE 3.7: Example from the Natutal Science Book

3.2 The uncertainty problem in the Educational Tutor

The estimation of uncertainty for our purpose, using the mathematical methods described in Chapter 2 (Background), will be explained in this section.

For this implementation, a new framework called LM-Polygraph (Fadeeva et al., 2023) was considered. LM-Polygraph is a framework that includes a range of state-of-the-art uncertainty estimation (UE) methods for Large Language Models (LLMs) in text generation tasks, with unified program interfaces in Python. It is compatible with the most recent LLMs, including BLOOMz (HuggingFace, n.d.(b)), LLaMA-2 (HuggingFace, n.d.(c)), ChatGPT, and GPT-4, and is designed to support future releases of similarly-styled models. Furthermore, similar to a human polygraph, it leverages various hidden signals to indicate when the output should not be trusted.

As mentioned previously, LM-Polygraph provides a comprehensive collection of UE techniques for LLMs in text generation tasks. It supports both white-box[1] and black-box models, using different methods and techniques for each case. In this project, since black-box models, such as GPT-3.5 and GPT-4, are used, the focus will be on the mathematical methods for these models described in previous chapters.

The experiments will be on two mathematical methods for estimating the uncertainty of the output responses from the GPT models. The methods to be utilized are Lexical Similarity and the Graph Laplacian Eigenvalue Sum (refer to Chapter 2, Section 2.3.2). Additionally, these methods will be applied to both GPT-3.5 and GPT-4 models accordingly.

[1]White-Box models provide transparency into their internal workings, allowing users to see and understand how decisions are made.

3.2.1 Estimation of Uncertainty Scores

All the necessary libraries from LM-Polygraph were loaded for this implementation are showned in Figure 3.8:

```
from lm_polygraph.utils.model import BlackboxModel
from lm_polygraph.utils.manager import estimate_uncertainty
from lm_polygraph.estimators import LexicalSimilarity, EigValLaplacian
```

FIGURE 3.8: Libraries from LM-Polygraph

The Lexical Similarity function calculates the mean similarity between all pairs of sampled generations with a negative sign, wich ensures a consistent interpretation of uncertainty. Also, it takes as default the 'rougel' model to calculate the similarity metric. After using the 'estimate-uncertainty' function, a float uncertainty score is returned for each sample in the input statistics. Higher values indicate more uncertain samples, but since the scores are negative, the absolute values were taken, a value closer to zero (less negative) implies a higher similarity among the sampled texts, indicating lower uncertainty. Conversely, a more negative value (farther from zero on the negative side) implies higher dissimilarity score among the sampled texts, indicating higher uncertainty.

The EigValLaplacian function estimates sequence-level uncertainty in a language model using the "Sum of Eigenvalues of the Graph Laplacian" method described in Chapter 2, Section 2.3.2. This method leverages Natural Language Inference (NLI) to compute similarity scores, using a semantic metric (either entailment or contradiction, as detailed in Section 2.3.2). The function processes statistics such as generated samples in "input-messages", text-file, and the matrix with semantic similarities to estimate and return uncertainty scores for each input sample. These uncertainty scores are positive numbers, with higher values indicating greater uncertainty.

The experimental cases for better understanding are outlined as follows:

- **1st Case:** Lexical Similarity with the GPT-3.5 model.
- **2nd Case:** Lexical Similarity with the GPT-4 model.
- **3rd Case:** Graph Laplacian Eigenvalue Sum with the GPT-3.5 model.
- **4th Case:** Graph Laplacian Eigenvalue Sum with the GPT-4 model.

Therefore, after considering the previous mentioned cases it is interesting to examine and answer questions as follows:

- **Research Question 1:** Which model provides the best uncertainty values for each method?
- **Research Question 2:** Does the Graph Laplacian Eigenvalue Sum method rank the uncertainty estimations in the same way as the Lexical Similarity method?
- **Research Question 3:** How reliable are the uncertainty estimations for each of the cases?

It is important to note that a compatibility issue arose between the OpenAI versions required by LM-Polygraph and Langchain. LM-Polygraph necessitates version "openai==0.28.0," while Langchain requires the latest version of OpenAI. Consequently, the uncertainty scores could not be implemented into the Educational

Tutor (Chatbot). However, different prompts and questions that were used to interact with the Chatbot, will be used for the experiments. These prompts where saved in a text-file, named "input-messages", for later used.

3.3 Experiments

In this section, samples from the produced dataset are presented for each question, followed by comments and insights. The following tables display information about the experiments for each research question, with more detailed analysis provided later on. The whole produced datasets can be found in the following GitHub link: Datasets

It is important to note that the results produced using Lexical Similarity as the estimator were calculated based on 129 questions from the chatbot, while using the Graph Laplacian Eigenvalue Sum as the estimator was limited to 19 questions due to the time-consuming nature of this method. In this document, only five prompts for each estimator and case will be shown for space purposes.

Table 3.1 illustrates the first five prompts/questions asked in the chatbot, along with the corresponding output answers. Additionally, the uncertainty scores for both GPT models are shown, with Lexical Similarity used as the estimator for this calculation. Table 3.2 presents similar information, but the estimator used is the Graph Laplacian Eigenvalue Sum. These tables provide information to answer research question 1.

Tables 3.3 and 3.4 show the first five prompts/questions ranked from highest to lowest uncertainty score for GPT-3.5, according to both estimators. Similarly, Tables 3.5 and 3.6 present the same information for the GPT-4 model. These tables provide information to answer research question 2.

For research question 3, the mean uncertainty scores for the first and second datasets of each of the four cases are shown after regenerating the uncertainty scores for the same cases (Table 3.7).

TABLE 3.1: Sample of uncertainty scores for GPT-3.5 and GPT-4 models with Lexical Similarity.

Input Prompt	Output Answer	Uncertainty with GPT-3.5	Uncertainty with GPT-4
tell me about the international and national arenas in Cyprus	Cyprus is a Mediterranean island country with a complex and unique international and national arena. Internationally, Cyprus has been a highly contested and divided island since the 1974 Turkish invasion and subsequent partition of the island into the internationally recognized Republic of Cyprus in the south and the self-declared Turkish Republic of Northern Cyprus in the north. The ongoing division of the island has been a source of tension and conflict and has had significant implications for Cyprus's international relations. Cyprus has been a member of the EU	-0.3512994	-0.3369823
How did the european union react when cyprus was divided along ethnic lines in 1964 due to EOKA terror campaign?	The European Union did not exist in 1964, so it did not have a direct response to the division of Cyprus along ethnic lines due to the EOKA terror campaign. However, individual EU member states and other international actors expressed concern and called for peaceful resolution of the conflict in Cyprus. The United Nations took a leading role in trying to mediate the situation and eventually deployed a peacekeeping force, known as the United Nations Peacekeeping Force in Cyprus (UNFICYP).	-0.3844868	-0.3452402
Have the greek Cypriots always constituted a majority in Cyprus?	Yes, the Greek Cypriots have always constituted a majority in Cyprus. Prior to the division of the island in 1974, Cyprus had a population that was approximately 80% Greek Cypriot and 18% Turkish Cypriot. After the division, the Turkish Cypriots established their own separate administration in the north, making them the majority population in that region. As a result, the demographics of Cyprus have changed significantly since the division.	-0.4027987	-0.3168868
What about the bronze time in cyprus	The Bronze Age in Cyprus lasted from around 2500 BC to 1050 BC. During this time, the island was a center of trade and cultural exchange, with influences from neighboring regions such as the Near East and Greece. Cyprus was known for its production of copper and bronze, which were highly sought after commodities in the ancient world. The Bronze Age in Cyprus saw the development of sophisticated pottery, metalworking, and monumental architecture, including the construction of impressive tombs and temples.	-0.3751035	-0.3233221
What were the causes of the Turkish invasion in Cyprus?	The Turkish invasion of Cyprus in 1974 was sparked by a number of underlying causes, including: The desire to protect the Turkish Cypriot minority: Following the island's independence from British colonial rule in 1960, tensions between the Greek Cypriot and Turkish Cypriot communities escalated, leading to violence and discrimination against the Turkish Cypriots. The Turkish government intervened to protect the Turkish Cypriot minority from persecution and to secure their rights on the island.	-0.3773687	-0.354291

TABLE 3.2: Sample of uncertainty scores for GPT-3.5 and GPT-4 models with Graph Laplacian Eigenvalue Sum.

Input Prompt	Output Answer	Uncertainty with GPT-3.5	Uncertainty with GPT-4
tell me about the international and national arenas in Cyprus	In the international arena, Cyprus is a member of numerous organizations and agreements, including the United Nations, the European Union, the Council of Europe, the Commonwealth of Nations, and the World Trade Organization. The country has also established diplomatic relations with numerous countries around the world. Cyprus has been involved in ongoing efforts to resolve the long-standing division of the island between the Greek Cypriot and Turkish Cypriot communities. The United Nations has been actively involved in facilitating negotiations between the two sides.	3.7948324	6.0480381
How did the european union react when cyprus was divided along ethnic lines in 1964 due to EOKA terror campaign?	The European Union did not exist in 1964. The European Economic Community (EEC), which was the predecessor of the EU, was established in 1957. In response to the division of Cyprus along ethnic lines in 1964, the EEC took a diplomatic approach to try to resolve the conflict. The EEC member states issued statements calling for a peaceful resolution to the crisis and urging both sides to refrain from violence. The EEC also supported the efforts of the United Nations.	4.2764597	3.2875223
Have the greek Cypriots always constituted a majority in Cyprus?	Yes, the Greek Cypriots have always constituted a majority in Cyprus. The demographic composition of the island has changed over time due to various factors such as migration, population growth, and conflicts. At certain points in history, the Turkish Cypriot population has been equal to or even larger than the Greek Cypriot population. Today, the Greek Cypriots are the majority population in Cyprus, but the island remains divided between Greek Cypriot and Turkish Cypriot communities.	3.8615933	5.1233688
What about the bronze time in cyprus	The Bronze Age in Cyprus lasted from around 2300 to 1050 BC. During this time, the island was an important hub for trade in the eastern Mediterranean, particularly in metals such as copper, which Cyprus was known for producing. The Bronze Age in Cyprus saw the development of advanced metallurgy techniques, as well as the rise of sophisticated urban centers and complex societies. Archaeological evidence from this period includes large palaces, tombs, and religious sites, showcasing the rich cultural and technological advancements.	2.2675932	3.0284007
What were the causes of the Turkish invasion in Cyprus?	The Turkish invasion of Cyprus in 1974 was primarily caused by longstanding tensions between the Greek and Turkish Cypriot communities on the island, as well as the breakdown of power-sharing agreements put in place after Cyprus gained independence from Britain in 1960. Ethnic tensions: The Greek and Turkish Cypriot communities had a history of conflict and violence, stemming from disputes over power-sharing, land ownership, and cultural identity. These tensions escalated in the 1960s and early 1970.	3.2321982	2.9194977

TABLE 3.3: Sample of ranked uncertainty scores for GPT-3.5 model with Lexical Similarity.

Input Prompt	**Uncertainty with GPT-3.5**
What were the major events and developments during the Byzantine rule of Cyprus?	-0.2932287
What were some notable archaeological finds from Roman Cyprus?	-0.2986870
What were the impacts of Ottoman rule on Cyprus, and how did the island adapt to its new rulers?	-0.2987658
What were the outcomes of the Zurich-London agreements?	-0.3020848
Can you give me suggested question that I might use in an exam for that period?	-0.3143475

TABLE 3.4: Sample of ranked uncertainty scores for GPT-3.5 model with Graph Laplacian Eigenvalue Sum.

Input Prompt	**Uncertainty with GPT-3.5**
What about the bronze time in cyprus	2.2675932
What was the Greek Junta?	2.3972515
What evidence exists of Mycenaean influence on Cyprus?	2.5636839
How did Cyprus fare during the Bronze Age?	2.7019519
How did Cyprus come under Egyptian rule during the New Kingdom period?	2.9013555

TABLE 3.5: Sample of ranked uncertainty scores for GPT-4 model with Lexical Similarity.

Input Prompt	**Uncertainty with GPT-4**
Can you give me suggested question that I might use in an exam for that period?	-0.1942356
What were some notable archaeological finds from Roman Cyprus?	-0.1997253
How did Cyprus come under Egyptian rule during the New Kingdom period?	-0.2116466
What were the challenges faced by Cyprus in implementing EU laws and regulations after accession?	-0.2154664
How has the economy of Cyprus evolved over time?	-0.2373326

TABLE 3.6: Sample of ranked uncertainty scores for GPT-4 model with Graph Laplacian Eigenvalue Sum.

Input Prompt	**Uncertainty with GPT-4**
What evidence exists of Mycenaean influence on Cyprus?	1.6800014
What role did Cyprus play in the trade networks of the ancient Mediterranean?	2.0429651
Give me questions about the Classical period in cyprus	2.1191324
What were the causes of the Turkish invasion in Cyprus?	2.9194977
What about the bronze time in cyprus	3.0284007

TABLE 3.7: Mean Uncertainty Scores for Each Case: Comparison of First and Second Datasets

Estimator	**Model**	**Mean for first dataset**	**Mean for second dataset**
Lexical similaity (LS)	GPT-3.5	-0.4123847	-0.4088312
Lexical similaity (LS)	GPT-4	-0.3275486	-0.3309661
Graph Laplacian (EVL)	GPT-3.5	3.7339862	4.2106864
Graph Laplacian (EVL)	GPT-4	3.9467708	3.5442523

Now some insights derived from the tables for each research question will be discussed in detail.

3.3.1 Research Question 1: Which model provides the best uncertainty values for each method?

Comparing the uncertainty scores indicated in Table 3.1, where the estimator is Lexical Similarity, it is observed that significantly better results are achieved using the GPT-4 model compared to the GPT-3.5 model.

Conversely, when comparing the uncertainty scores indicated in Table 3.2, where the estimator is Graph Laplacian Eigenvalue Sum, it is unclear which model produces better uncertainty scores. It is evident that for some questions, the GPT-3.5 model performs better than the GPT-4 model and vice versa.

Therefore, with Lexical Similarity as the estimator, it is clear that the results are better when using the GPT-4 model. However, the same conclusion cannot be drawn when using the Graph Laplacian Eigenvalue Sum.

3.3.2 Research Question 2: Does the Graph Laplacian Eigenvalue Sum method rank the uncertainty estimations in the same way as the Lexical Similarity method?

This question can be answered for both models (GPT-3.5 and GPT-4). From Tables 3.3 - 3.6 it is observed that the methods in both models do not produce consistent rankings. Therefore, the rankings are inconsistent.

In each case, the closeness of the ranking can be measured. To address this, the Kendall tau rank distance will be used. This metric, or distance function, counts the number of pairwise disagreements between two ranking lists. A larger distance indicates greater dissimilarity between the two lists. The Kendall tau ranking distance between two lists τ_1 and τ_2 is

$$K_d(\tau_1, \tau_2) = |\{(i,j) : i < j, [\tau_1(i) < \tau_1(j) \wedge \tau_2(i) > \tau_2(j)] \vee [\tau_1(i) > \tau_1(j) \wedge \tau_2(i) < \tau_2(j)]\}|$$

where $\tau_1(i)$ and $\tau_2(i)$ are the rankings of the element i in τ_1 and τ_2 respectively. $K_d(\tau_1, \tau_2)$ will be equal to 0 if the two lists are identical and $\frac{1}{2}n(n-1)$ (where n is the list size) if one list is the reverse of the other.

It is important to note that, since the datasets with Lexical Similarity are significantly longer with 129 rows of uncertainty scores compared to the datasets with Eigenvalue Laplacian, which have 19 rows of uncertainty scores, the distance will be calculated using the common questions.

The distance between the ranked lists of Lexical Similarity and Graph Laplacian Eigenvalue Sum will be calculated and checked using the GPT-3.5 model and then the GPT-4 model. The results are showed in the Table 3.8:

Model	**Distance**	**p-value**
GPT-3.5	0.95588235	0.76541106
GPT-4	1.00735294	0.96033718

TABLE 3.8: Kendal Tau Rank Dinstance and the p-values for the 2 models(GPT-3.5 and GPT-4)

Therefore, for GPT-3.5, a distance of 0.956 indicates that the rankings are quite dissimilar. However, with GPT-4, a distance of 1.007 indicates that the rankings are even more dissimilar than those with GPT-3.5.

In addition, for GPT-3.5, a p-value of 0.765 indicates that the correlation is not statistically significant. This high p-value suggests weak evidence against the null hypothesis (no correlation). With GPT-4, a p-value of 0.960 suggests even stronger evidence that the correlation is not statistically significant, indicating a very weak correlation likely due to random chance.

Thus, GPT-3.5 has rankings that are slightly closer to the reference rankings compared to GPT-4, as indicated by the lower Kendall tau rank distance. On the other hand, the correlations in both models are weak and not statistically significant, as indicated by the high p-values. This suggests that neither model's rankings are meaningfully correlated with the reference rankings.

3.3.3 Research Question 3: How reliable are the uncertainty estimations for each of the cases?

Looking at Table 3.7, it is evident that the models do not provide stable answers, resulting in inconsistent outputs. Nonetheless, in case where Lexical Similarity is used as the estimator the difference between the mean uncertainty scores for the two different produced datasets appears to be smaller than when the estimator is the Graph Laplacian Eigenvalue Sum. In other words, the output answers are not very reliable.

3.3.4 Experiments Conclusions

Based on the previous results, it can be concluded that calculating the uncertainty of the output answers of LLMs may not be efficient. The uncertainty scores were found to be unstable and varied significantly across different cases. Therefore, the problem of quantifying uncertainty in LLMs remains an open issue and requires further research and development to achieve more reliable methods.

It is not advisable to draw a definitive conclusion regarding which model, GPT-3.5 or GPT-4, performs better based on their uncertainty scores. The observed instability in uncertainty scores for both models indicate that these scores do not provide a dependable metric for comparison. Given this instability, it cannot be asserted that GPT-4 is superior to GPT-3.5.

Moreover, the methodology of using the input prompt/question to estimate the uncertainty level in the LLM output might not be ideal for obtaining reliable results with black-box LLMs. The complexity of these models make it challenging to develop methods that accurately capture uncertainty. This suggests a need for new approaches or improvements in existing techniques to better estimate uncertainty in LLM outputs.

To conclude, the uncertainty of the output answers should be considered an additional limitation of LLMs. As current methods for calculating uncertainty are unreliable and inconsistent, this limitation poses a significant challenge. Until more reliable and trusted methods are developed, the uncertainty associated with LLM outputs will remain a critical issue that limits the practical application of these models.

Chapter 4

Conclusions

First of all, Large Language Models (LLMs) have demonstrated remarkable performance across a variety of text generation tasks. Moreover, the development of chatbots for educational purposes or other uses has become significantly easier due to the availability of advanced tools. In this study, tools such as Langchain, Retrieval-Augmented Generation (RAG), Chroma, and OpenAI models were utilized to implement the Educational Tutor. Therefore, it can be concluded that by leveraging a wide range of new tools, it is feasible to create and customize chatbots for various fields and purposes efficiently.

On the other hand, based on the previously analysis and results, it can be concluded that GPT models should not be fully trusted for educational purposes. The uncertainty estimations reveal that the outputs are highly variable and not very informative. For teachers, these models can be beneficial since they possess the knowledge to evaluate the chatbot's responses and determine their accuracy. However, for students, the reliability of these models is questionable, as the information provided might be inaccurate, misleading, or incomplete, potentially hindering their learning process.

Furthermore, the field of measuring and estimating uncertainty scores specifically for black-box models is very limited. Despite utilizing one of the best frameworks for uncertainty evaluation, LM-Polygraph (Fadeeva et al., 2023), the results of the previous analysis were not particularly accurate. Additionally, it was noted that the corresponding paper (Fadeeva et al., 2023) did not verify the results by passing the same question through these methodologies and comparing the different scores. In this study, this case was examined, and the results were found to be unsatisfactory.

Lastly, research in this field must continue, with scientists developing more specific tools or improving existing ones for estimating the uncertainty of output answers, particularly for black-box models. Given the rapid and multifaceted growth in this area, it is crucial to thoroughly check and evaluate these tools to ensure more trusted and safe use, especially in applications such as education. Enhanced uncertainty estimation techniques will not only improve the reliability of educational tools but also strengthen their credibility in other sensitive fields such as healthcare and finance. As the adoption of LLMs expands, establishing robust methods for assessing uncertainty will be essential to mitigate risks and ensure the accuracy and safety of their applications.

Bibliography

Andrew Y. Ng Michael I. Jordan, Yair Weiss (2001). “On Spectral Clustering: Analysis and an algorithm”. In: URL: https://proceedings.neurips.cc/paper_files/paper/2001/file/801272ee79cfde7fa5960571fee36b9b-Paper.pdf.

Chainlit (2023). “Chainlit Documentation”. In: URL: https://docs.chainlit.io/get-started/overview.

Fadeeva, Ekaterina et al. (2023). “LM-Polygraph: Uncertainty Estimation for Language Models”. In: arXiv: 2311.07383 [cs.CL].

Fomicheva, Marina et al. (2020). “Unsupervised Quality Estimation for Neural Machine Translation”. In: arXiv: 2005.10608 [cs.CL].

He, Pengcheng et al. (2021). “DeBERTa: Decoding-enhanced BERT with Disentangled Attention”. In: arXiv: 2006.03654 [cs.CL].

Hu, Mengting et al. (2023). “Uncertainty in Natural Language Processing: Sources, Quantification, and Applications”. In: arXiv: 2306.04459 [cs.CL].

HuggingFace (n.d.[a]). “Bert Language Model”. In: (). URL: https://huggingface.co/docs/transformers/model_doc/bert.

— (n.d.[b]). “BLOOMz”. In: (). URL: https://huggingface.co/bigscience/bloomz.

— (n.d.[c]). “LAMMA”. In: ().

Koubaa, Anis (2023). “GPT-4 vs GPT-3.5: A Concise Showdown”. In: URL: https://www.researchgate.net/publication/369897711_GPT-4_vs_GPT-35_A_Concise_Showdown.

Kuhn, Lorenz, Yarin Gal, and Sebastian Farquhar (2023). “Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation”. In: arXiv: 2302.09664 [cs.CL].

langchain (2023). “LangChain Documentation”. In: URL: https://python.langchain.com/v0.2/docs/introduction/d.

Lin, Zhen, Shubhendu Trivedi, and Jimeng Sun (2024). “Generating with Confidence: Uncertainty Quantification for Black-box Large Language Models”. In: *Transactions on Machine Learning Research*. ISSN: 2835-8856. URL: https://openreview.net/forum?id=DWkJCSxKU5.

Mamdouh, Marawan (2023). “ROUGE”. In: URL: https://dev.to/aws-builders/mastering-rouge-matrix-your-guide-to-large-language-model-evaluation-for-summarization-with-examples-jjg.

Miah, Shahajan (2020). “Basics in Natural Science”. In: URL: https://www.researchgate.net/publication/351096547_Basics_in_Natural_Science.

Mishra, Onkar (2023). “Using langchain for Question Answering on Own Data”. In: URL: https://medium.com/@onkarmishra/using-langchain-for-question-answering-on-own-data-3af0a82789ed.

OpenAI (n.d.). “ChatGPTDoc”. In: (). URL: https://platform.openai.com/docs/guides/text-generation.

— (2023). “GPT-4 Technical Report”. In: URL: https://cdn.openai.com/papers/gpt-4.pdf.

Petrides, Antonis (2013). “Old-Modern History of Cyprus”. In: URL: https://www.researchgate.net/publication/305398007_Introduction_to_the_History_of_Cyprus.

Sai, Abhi (2023). “GPT-3,Architecture”. In: URL: https://medium.com/@tsaiabhi.cool/explaining-gpt-3-architecture-and-working-d0219c79202c.

Shaw, Peter, Jakob Uszkoreit, and Ashish Vaswani (2018). *Self-Attention with Relative Position Representations*. arXiv: 1803.02155 [cs.CL].

Vaswani, Ashish et al. (2023). *Attention Is All You Need*. arXiv: 1706.03762 [cs.CL].

Yarin Gal, Zoubin Ghahramani (2016). “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning”. In: URL: https://proceedings.mlr.press/v48/gal16.html.