



UHASSELT

KNOWLEDGE IN ACTION



Maastricht University

Faculteit Wetenschappen **School voor Informatietechnologie**

master in de informatica

Masterthesis

Find By Idea: Interpreting AI with Training Sample Proximity in Embedding Space

Joren Martens

Scriptie ingediend tot het behalen van de graad van master in de informatica

PROMOTOR :

Prof. dr. Gustavo Alberto ROVELO RUIZ

COPROMOTOR :

Prof. dr. Davy VANACKEN

BEGELEIDER :

De heer Sebe VANBRABANT

De heer Gilles EERLINGS

De transnationale Universiteit Limburg is een uniek samenwerkingsverband van twee universiteiten in twee landen: de Universiteit Hasselt en Maastricht University.



UHASSELT

KNOWLEDGE IN ACTION

www.uhasselt.be

Universiteit Hasselt
Campus Hasselt:
Martelarenlaan 42 | 3500 Hasselt
Campus Diepenbeek:
Agoralaan Gebouw D | 3590 Diepenbeek

2024
2025



Maastricht University

Faculteit Wetenschappen ***School voor Informatietechnologie***

master in de informatica

Masterthesis

Find By Idea: Interpreting AI with Training Sample Proximity in Embedding Space

Joren Martens

Scriptie ingediend tot het behalen van de graad van master in de informatica

PROMOTOR :

Prof. dr. Gustavo Alberto ROVELO RUIZ

BEGELEIDER :

De heer Sebe VANBRABANT

De heer Gilles EERLINGS

COPROMOTOR :

Prof. dr. Davy VANACKEN

UNIVERSITEIT HASSELT

MASTERPROEF VOORGEDRAGEN TOT HET BEHALEN VAN DE
GRAAD VAN MASTER IN DE INFORMATICA

Find By Idea: Interpreting AI with Training Sample Proximity in Embedding Space

Auteur:

Joren Martens

Promotor:

prof. dr. Gustavo Roveló Ruiz

Co-promotor:

prof. dr. Davy Vanackén

Begeleider(s):

Gilles Eerlings
Sebe Vanbrabant

Academiejaar 2024-2025



Acknowledgements

I would like to thank my promoters Prof. dr. Gustavo Roveló Ruiz and Prof. dr. Davy Vanacken for their invaluable guidance and support throughout the course of this thesis. Their time, constructive feedback, and insightful suggestions greatly contributed to the quality and direction of my research. As well as Gilles Eerlings and Sebe Vanbrabant, my supervisors, for their continuous support, numerous discussions, and readiness to assist whenever I encountered challenges.

I would also like to thank my friends, many of whom participated in the user study or helped recruit additional participants and provided ongoing encouragement and moral support during this journey.

Lastly I would like to thank my family for always listening when I shared progress of this research, being very supportive and making sure that I could fully dedicate myself to this thesis.

Abstract

With the rising popularity of Large Language Models (LLMs) come many new users, most of which don't have a technical background and don't have realistic expectations of these models. Explainable Artificial Intelligence (XAI) methods have improved transparency in other AI domains, making it more clear why the model returns a certain output and what can be expected of it. But these techniques are underdeveloped for LLMs and the ones that are available are not very user-friendly.

This thesis introduces a novel, user-friendly XAI method aimed at improving the appropriate trust levels of users. The proposed approach leverages the LLM's embedding space to retrieve semantically similar, human-generated training samples relevant to the user's input. These samples are presented through an intuitive web interface, helping users assess the trustworthiness of the model's output based on real-world examples.

A user study was conducted to evaluate the tool's effectiveness in increasing appropriate trust without sacrificing usability. The results indicate that participants adjusted their trust levels in response to conflicting examples, demonstrating improved critical engagement with the model's outputs. Additionally, the application scored highly on the User Experience Questionnaire, confirming that the added explanations did not hinder usability.

This work contributes to the field of LLM interpretability by offering a practical and accessible XAI solution that supports trust calibration and enhances user understanding without requiring technical expertise.

Summary

Introduction

With the increasing popularity of LLM applications like ChatGPT and GitHub Copilot, many non-technical users are exposed to these modern AI techniques for the first time. Not all these people realize that these systems are a black box, where every input returns an output, but it is unclear how this outcome was formed. Not only should this black box problem be researched to make it more transparent, but it should also be made clear to new users what they can expect from an LLM.

This thesis discusses the current XAI techniques that can be used to explain the inner workings of the LLM. Understanding how these models arrive at their outputs is essential for ensuring transparency, trust, and accountability. The focus is the trust that users place in the LLM, as new users tend to overestimate the capabilities of the model and trust it too much. By analysing the strengths and limitations of these approaches, this thesis contributes to the ongoing effort to make powerful language models more interpretable and reliable.

The primary contribution of this work is a user-friendly method to support appropriate levels of user trust. This method involves presenting users with human-generated training samples that are semantically similar to the current conversation. By leveraging the LLM's own embedding function to compute vector representations of text, we retrieve examples from the training data that align closely with the user's query. This allows users to assess the model's output in relation to real-world, relevant examples, making the decision process more interpretable.

Background

While a lot of research has been done on traditional forms of XAI, like feature attribution using SHAP or LIME, saliency maps that show the importance of each input feature, and others, not much research is available for LLM XAI. The methods of making an LLM more transparent are also not very user-friendly, despite a multitude of papers calling for the need for a user-friendly LLM XAI method, as the target audience of LLM-based applications often includes non-technical users. One of the current methods, for example, visualizes the attention values, which are hidden values of the LLM that show what previous tokens were important when generating the next one. The problem with this approach is that it is rarely useful for a typical user, as these values are difficult to interpret, even for LLM developers. They could show patterns that indicate that the LLM likely needs more training, but these are hard to understand and find, and not very useful for the user who can not use that information to further train the model like a developer would.

To understand the possible XAI methods for an LLM, the architecture of the LLM was also studied, this way each individual component could be explained. First is the tokenizer, which converts a text into tokens, which are represented by numerical identifiers. This way, the LLM can perform mathematical calculations on a representation of a text. A possible XAI method to explain the tokenizer is colouring the background of the individual tokens in the output of an

LLM-based chat application. The tokens then go through an embedding table, which transforms each discrete token into a dense vector representation. These embeddings capture semantic information about the tokens, allowing the model to understand relationships between words and concepts in a vector space. This vectorized input is then passed on to subsequent layers, such as attention mechanisms or transformer blocks, for further processing and contextualization. The attention mechanism has also already been made more transparent by applying visualizations where important tokens are highlighted to indicate that a high attention value was calculated when a selected token was generated. The embedding mechanism and the transformer block, however, lack sufficiently good XAI techniques.

By using the embedding process of an LLM, two samples of text can be semantically compared, and a distance can be calculated by measuring the angle between the vector representations of those samples. This thesis uses the embedded vector representations as an XAI method by giving users more trust in the output of an LLM, by serving similar training samples. While not making the reason why a certain output came out of the model clearer, this can increase the appropriate trust. The training samples are created by a human and are thus known not to be made up. By comparing the information from the output and the samples, users can make up for themselves whether the output can be trusted.

Implementation

Firstly, the possible model choices for the implementation of the similar training sample XAI method are considered. The initial implementation used a custom-made transformer-based language model, this allowed full control over the training process and gave access to all the training data so it could be used for the similar data search. However, this model returned subpar results as it would not produce grammatically correct English. Using this model for an XAI method to measure trust could compromise the validity of the study, as its poor performance may bias participants' perceptions and lead to misleading conclusions about the effectiveness of the explanation techniques.

Next, the GPT architecture was considered, as it offers state-of-the-art performance in language generation tasks. But it was quickly found that the newer models do not include what data they are trained on, and could thus not be used for the similarity search method. Finally, the more open-source Llama was used. Here, the data sources were stated in the accompanying paper, and unlike the newer GPT models, the attention values and embedding table could also be accessed. While first using the second-generation model with seven billion parameters, the smaller third-generation model with three billion parameters was found to be much faster and gave better responses.

The Llama model was then used to develop a technique to find training samples similar to the current input of the LLM. By using the embedding vectors that an LLM can create, different texts can be mathematically compared to rank semantic similarity. However, Llama used more than four terabytes of data to train the model. Therefore, it was trained on custom finetuning data as well and only the information from that finetuning data was requested from the model during these tests.

That similarity search technique was then used in a web application to be used as an XAI tool, which supports the user in assessing how trustworthy the model output is. The top three most similar training samples are shown in a podium-like visualization, with the possibility of exploring even more samples in a custom plot. This custom plot creates circles, where the radius scales with the similarity score calculated by the tool. Circles close to the centre are more similar to the current conversation than the ones further away.

The tool was subsequently evaluated through a user study to assess its effectiveness in fostering appropriate levels of trust while maintaining a user-friendly design. Participants were asked to rank their trust in the model with and without the tool, each on a different question set. Each set also had a conflicting training sample, where two samples had the same question, but

a different answer. In this case, appropriate trust would mean that a user would have a lot of faith in the model when answering the questions, except when the conflicting answer came up. In that case users should realize that the output might not be correct and should trust the model less to be right at that point.

Results

In the user study, each participant had to answer six questions using the LLM without the tool and then six questions with the XAI tool’s assistance. Each question set then had one conflicting training sample, which should decrease the current trust in the model when it came up. It was found that a significant number of people did so and rated the perceived trust in the LLM to be lower when those samples were shown.

Furthermore, the potential influence of the order of question sets, tool usage, or their combination was examined. However, no significant effects were found. This rules out the possibility that increased trust in the LLM was due to participants being influenced by prior exposure to the tool or question content. As an extra measure, there were also three different training datasets that could be used. Two for each question set, where they don’t contain the questions asked in the other, and a separate training dataset where none of the asked questions are saved. This way, when showing users what was expected with a demo question, they could not accidentally see extra information about questions that were yet to come, influencing possible trust ratings. For that same reason, the two actual datasets were also split.

Measuring user-friendliness was done with the User Experience Questionnaire (UEQ). This measured different aspects of the impact of the application on the user experience. It was found that both applications, with and without XAI tool, were considered easy to work with and nice to look at. This means that despite all the extra information presented to the user, it is easy to interpret and does not stand in the way. It also scored well on the provided benchmark, scoring above 70 percent of the other application in all categories: attractiveness, ease of use, efficiency, dependability, stimulation and novelty. The application that included the XAI tool also scored significantly higher on stimulation compared to the one without.

Conclusion

LLMs are becoming increasingly popular, including for non-technical people who don’t realize what they can expect from such a system. These black box models also don’t explain how they came to those outcomes, making it difficult to evaluate those outputs. While XAI has successfully improved transparency for other types of AI, its application to LLMs is still limited and often not user-friendly.

This thesis addresses that gap by introducing a user-friendly XAI method for LLMs that leverages semantic similarity between user inputs and human-generated training samples. By using the model’s own embedding mechanism to retrieve and present semantically similar examples from its training data, users are provided with real-world context that helps them assess the trustworthiness of the model’s response. A novel XAI method tailored to the specific challenges of LLMs is presented. It demonstrates that it is possible to support user trust in a meaningful way while maintaining a high standard of usability. This makes it a promising step forward in making LLMs more transparent and responsibly adopted by a broader audience.

The results of the user study show that this approach increases appropriate trust, users trust the model more when it performs well, and less when conflicting examples cast doubt on its output. Importantly, this is achieved without sacrificing user experience, as confirmed by positive feedback in the UEQ. The tool was found to be intuitive, visually appealing, and easy to use.

Contents

1	Introduction	9
2	Background	12
2.1	Large Language Models	12
2.1.1	Tokenization	12
2.1.2	Generative Language Models	14
2.1.3	Hallucination	17
2.1.4	Ethical Considerations	18
2.2	Improving Transparency	18
2.2.1	Tokenization Visualization	18
2.2.2	Explainability With Feature Attribution	19
2.2.3	Attention as Explanation	20
2.2.4	User Expectations	22
2.2.5	Improving User Interaction	23
2.3	Transparency By Finding Similar Training Data	24
2.3.1	Similar Training Sample as XAI: Literature Example	24
2.3.2	Semantically Similar Text By Vector Embedding: Literature Example	25
2.3.3	Database Search	26
3	Implementation	28
3.1	Model choices	28
3.1.1	N-Grams	28
3.1.2	BERT	29
3.1.3	GPT	29
3.1.4	Llama	29
3.2	Choosing the right model	30
3.3	Finding similar data	31
3.4	Architecture	33
3.5	Frontend	34
3.5.1	Podium Visualization	35
3.5.2	PCA Plot	35
3.5.3	Circular Representation Plot	36
3.5.4	Interaction and Deployment	36
4	User Study Findings and Interpretation	39
4.1	User Study	39
4.1.1	Experiment Setup	39
4.1.2	Experiment Questions	40
4.2	Results	41
4.2.1	Demographics	41
4.2.2	Trust rating	42
4.2.3	User Experience Questionnaire	45

- 4.3 Discussion 46
 - 4.3.1 Trust results 46
 - 4.3.2 User Experience Questionnaire results 50
- 5 Conclusions 51**
 - 5.1 Possible future directions 52
 - 5.2 Self-reflection 53
- A Appendix A: Finetune Data 60**
- B Appendix B: Demographics Questions 63**
- C Appendix C: Nederlandse Samenvatting 65**

Chapter 1

Introduction

Artificial Intelligence (AI) has been in use for some time now, powering applications like image classification that can help with medical diagnoses as demonstrated by Li et al. [38], or even fraud detection like the paper from Ghosh et al. [21] who already demonstrated in 1994 that such an AI-powered application was possible. AI is very versatile across industries and is transforming how we interact with technology in everyday life. But with the recent innovations concerning language modeling, many end users got their first direct contact with these AI models in the form of Large Language Models (LLMs). One prominent example is ChatGPT, a chat program powered by an LLM that enables the user to chat with artificial intelligence, causing the popularity of AI to grow exponentially.

As LLMs continue to advance, they are increasingly being integrated into critical applications, from legal and medical advice to automated customer support and content generation. Izzidien et al. [32] explored different ways that LLMs can improve legislation tasks, including contract review, legal document summarization, and case outcome prediction. Similarly, Yuan et al. [73] demonstrated that LLMs can assist in clinical documentation by automatically generating medical notes. Although Thirunavukarasu et al. [63] caution against the standalone use of these models, they recommend employing LLMs as tools to support human users. Nonetheless, LLMs are already being utilized in some medical applications like the Google Med-PaLM, an LLM trained on medical data, that is partnered with multiple organizations and clinics[24].

Despite their impressive capabilities, AI models suffer from a fundamental challenge: a lack of transparency in how these models generate responses, often referred to as the “black-box problem” [1]. As illustrated in Figure 1.1, the term reflects the difficulty in interpreting how internal model parameters influence outputs. LLMs are no different, as the opacity extends to all neural network-based machine learning models, which inherently struggle to convey the logic behind their decisions. These concerns become especially pressing when such models are used in high-stakes domains where incorrect or biased outputs can have serious consequences.

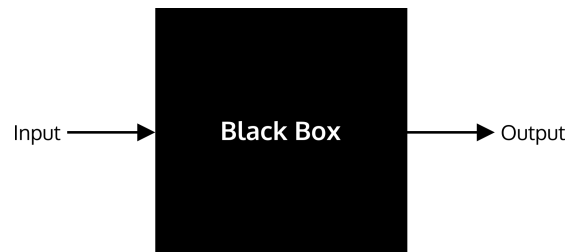


Figure 1.1: A visualization of the black-box problem shows inputs being processed by an AI model to produce an output, without providing any explanation for how or why that output was generated. Image courtesy of Codecademy [62].

Explainable Artificial Intelligence (XAI) seeks to address this by providing insights into model decision-making, thereby fostering appropriate trust between users and AI systems, where the user will trust the system when it is correct, but will have a distrust when the model gives uncertain responses. While XAI has gained traction in traditional machine learning models, its application to LLMs remains limited. Most current LLM deployments, like ChatGPT[49], Claude[4], GitHub Copilot[22] and many others, offer little to no explanation about how a response is generated, what data influenced it, or how confident the model is in its output. Without such transparency, users may either over-rely on the model’s responses or distrust them entirely, leading to suboptimal decision-making. Therefore, there is a pressing need to integrate effective XAI techniques into LLMs to enhance interpretability and trustworthiness.

While XAI has made significant progress in advancing traditional machine learning interpretability, many existing techniques are not easily transferable to LLMs, or they require substantial technical expertise to interpret. Feature attribution methods, such as SHAP [42] and LIME [55], are well-established XAI tools. These methods assess the importance of input features by slightly perturbing their values and measuring the resulting changes in model output. Features that produce greater output variance are considered more influential. There are also already existing methods specifically for LLMs, like visualizing attention values. These are values that the model learns that indicate the relative importance of previous tokens, small pieces of text in the context of LLMs, in predicting the next token. One example is shown in Figure 1.2, where a bias from the model is found with such a tool. These visualizations can reveal which parts of the input the model is focusing on during generation, offering users an intuitive understanding of the model’s attention patterns as it produces text.

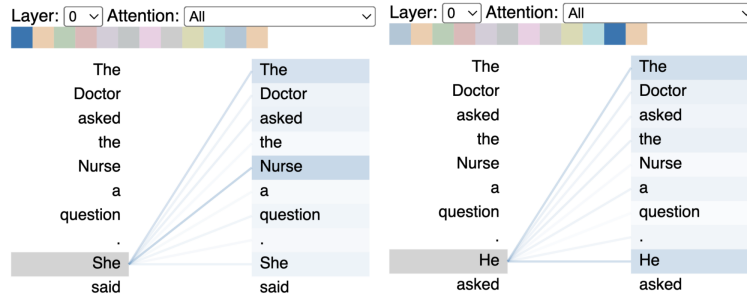


Figure 1.2: An example of an attention visualization from BertViz [67]. On the left, the token ‘she’ assigns significant attention to nurse, while on the right, the token ‘he’ focuses more on doctor. This highlights a bias in the model, which tends to associate female pronouns with traditionally female-dominated professions and male pronouns with traditionally male-dominated ones, reflecting gender stereotypes present in the training data. Image courtesy of Morgan [47].

All these methods, while useful for researchers and developers, often fall short for end-users who lack technical backgrounds. The problem with LLMs is that many users don’t know what they can realistically expect from an LLM. While XAI techniques have helped with similar issues for different AI model types, the LLM XAI methods are still lagging behind. These techniques, like SHAP or attention visualization, are also not focused on user experience, but are catered more towards developers. For these reasons this thesis investigates ways to help users get an intuitive understanding of what to expect from an LLM and make the LLM black box more transparent in a user-friendly way. Beside researching existing ideas, the possibility and viability of showing similar training samples in order to boost the trust and understanding of the LLM user has also been researched.

This thesis investigates whether it is possible to retrieve training data that is semantically similar to the ongoing conversation with a chat-based LLM. These models already generate vector representations of the current tokens to predict the next one. The resulting vectors encode semantic information, enabling meaningful comparisons between different tokens or

sequences. Therefore, the remaining challenge is to determine whether it is feasible to compare these semantic representations and identify the most similar entries within the training dataset and research if this improves the appropriate trust of users. By grounding model outputs in human-readable training data, the tool presented in the thesis aims to reshape how users interact with LLMs: not as opaque black boxes, but as traceable systems whose knowledge can be interrogated, audited, and understood.

The next chapter provides an in-depth overview of various methods for enhancing the explainability of LLMs. It also includes perspectives from relevant literature, highlighting expert opinions on the future direction of XAI techniques. The implementation chapter follows, detailing the system architecture and design choices that informed the development of the final application. This chapter also introduces a user study conducted to assess user trust in an LLM-based application, comparing scenarios with and without a tool that displays similar training examples. Finally, the results are presented and discussed.

Chapter 2

Background

This chapter discusses the work that has already been done to improve the explainability of LLMs. It starts by looking at the steppingstones that were needed to arrive at the current popular LLMs and also discusses some of those. Next the different ways of creating more transparency and trust for users in LLMs by using XAI methods are explored. Previous research on those methods reveals that there is a lack of user-friendly XAI systems. The last part will therefore discuss the viability of using a system that finds similar training samples as a user-friendly way of increasing transparency and trust in LLMs.

2.1 Large Language Models

To explore potential methods for increasing the transparency of LLMs, several internal mechanisms are examined in the following sections. This begins with an overview of tokenization methods, techniques that convert human-readable text into numerical representations suitable for machine processing. Next, a brief summary is provided of key language models that contributed to the development of modern LLMs and may serve as a basis for implementing XAI techniques.

2.1.1 Tokenization

An intuitive way to transform text into numbers is to use the existing way of text encoding that computers already use. Each individual character is given a numerical counterpart. Unicode Transformation Format (UTF) [14] is a widely used example in which a character is given a number made up of a variable number of bytes depending on the frequency of that character. This can help reduce the amount of data necessary to represent a piece of text and thereby improve the training time for language models.

Like character encoding algorithms such as UTF, Bytepair encoding introduced by Sennrich et al.[60], will also try to reduce the required number of bytes to represent text. But in this case, a combination of bytes can be encoded again if it is frequent enough. This also means that multiple characters could be grouped together into a single numerical token, which further reduces the size of the encoded text. This algorithm will group the combination of the most frequent pair of bytes or other tokens and will continue doing so until it reaches the given number of tokens in its vocabulary. It is therefore very easy to increase or decrease the size of that vocabulary if a larger or smaller network would be trained.

WordPiece, originally introduced by Schuster et al. [59] for Japanese and Korean voice search is a tool that will create tokens that are pieces of words by combining common pairs like the BPE algorithm. Wu et al. [72] later adapted this tool for neural machine translation. However, contrary to BPE, which joins based on frequency, WordPiece will calculate the likelihood of

a second token following a certain first token. The most likely pairs will be added to the vocabulary until the requested number of tokens is reached, or the likelihood falls below a certain threshold. All the tokens can then be given an identifier so the words can be translated into numbers, and they can be used for further machine learning training. WordPiece is used in BERT, a popular LLM introduced by Devlin et al.[17].

An issue with WordPiece is that by only relying on the likelihoods of token sequences, it does not explicitly learn or represent meaning. The word ‘ball’ for example could refer to a football, but also the formal dance event. If the training corpus contains significantly more examples of the word ‘ball’ in the sports context, WordPiece will be more likely to continue the sentence accordingly. This could cause issues where the usage of a word would not make sense semantically, as WordPiece only takes frequency into account and not the meaning behind the words.

To try and differentiate between multiple meanings of the same word, Mikolov et al. [45] developed Word2Vec, which will also take the surrounding words into account when transforming a string of text into a numerical representation. This algorithm will create a large vector that represents the meaning of a token, word, or even full sentence. By using the Skip-gram model, which will predict the surrounding tokens of a given token, there is more of a focus on the amount of training data as this is a simpler model, with less parameters and no hidden layers. The lack of hidden layers also means that there are no dense matrix multiplications, making the processing of millions of phrases very efficient.

Word2Vec can be used to create a word embedding, a vector that contains the meaning of words and their relations to other words. Although the relations depicted by the vector are very complex, by calculating the difference between two vectors and finding the closest known word, the relations can be intuitively shown and used by machines. An example from Mikolov et al.[44]: $\text{vector}(\text{“Madrid”}) - \text{vector}(\text{“Spain”}) + \text{vector}(\text{“France”})$ is closer to $\text{vector}(\text{“Paris”})$ than to any other word vector, indicating that the vector contains a relation for countries and their capital cities. Figure 2.1 shows this example for a two dimensional case, where it can be seen that the vector from Spain to Madrid is similar in direction and magnitude to the vector from France to Paris, illustrating how word embeddings capture semantic relationships such as country-capital pairs. These vector representations can then be used by LLMs to predict the next token, while taking the context into account.

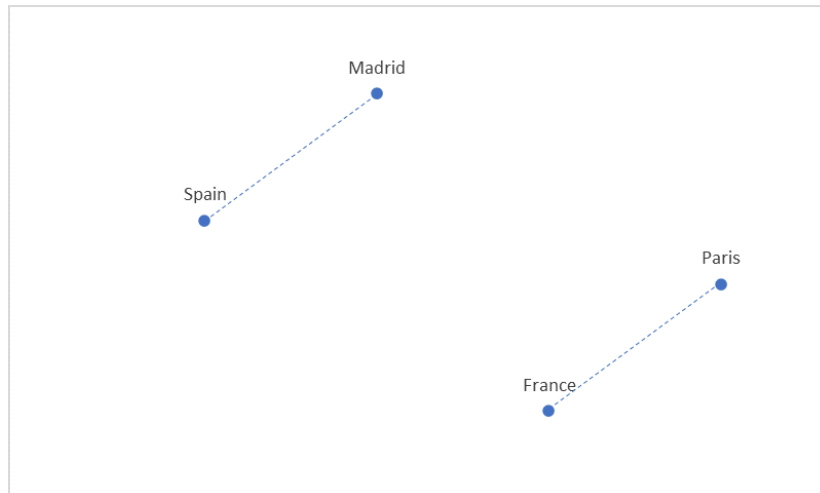


Figure 2.1: An example illustrating the relationships between embedding vectors. This plot shows how the vector from a country to its capital follows a similar direction for two different countries. This consistent pattern can be interpreted as representing the concept of capital cities.

2.1.2 Generative Language Models

There have been many approaches to enabling language models to understand and generate text. Some are easier to make explainable, but these also tend to have a worse performance at understanding a language and generating complex responses. In contrast, modern high-performing LLMs achieve impressive results in both understanding and generation, but their inner workings are far more opaque. This opacity makes it crucial to understand how these models function in order to effectively apply and develop robust XAI techniques. A deeper understanding of LLMs enables more meaningful insights into their behaviour, which is essential for building trust, ensuring fairness, and identifying potential failure modes.

One of the earliest methods for sentence generation involves statistical approaches such as n-gram [12] and skip-gram [45] models. The n-gram model considers the previous n tokens and calculates the probability of each possible next token in the vocabulary. The token with the highest probability is then selected to extend the given sequence. In contrast, the skip-gram model does not predict the next token directly; instead, it learns which words are likely to fill in gaps within a sequence. This allows the model to consider tokens at various positions, including the ends of sentences. Although skip-gram models do not generate text directly, they are useful for creating vector representations of words, capturing semantic relationships by embedding them into a vector space.

This prediction method is relatively simple and remains interpretable. It relies on identifying exact patterns from the training data and selecting the most probable continuation. However, a major limitation arises when no matching data exists. Consequently, n-gram models require vast amounts of training data to ensure coverage of all likely token combinations they may encounter.

To overcome the limitations of n-gram models, which rely on exact matches from training data, neural network architectures such as Recurrent Neural Networks (RNNs) introduced by Elman [19] and Long Short-Term Memory networks (LSTMs) by Hochreiter and Schmidhuber [26] have been developed. These models can learn sequential patterns without being constrained by fixed-size token windows. Unlike n-gram models, RNNs process tokens sequentially and maintain a hidden state that captures contextual information from the entire sequence. This hidden state acts as a memory, allowing the model to reference earlier parts of the input even in longer contexts. However, the quality of this memory degrades as the sequence length increases due to the vanishing gradient problem. Since the model's weights are repeatedly used in recursive computations, the gradients needed for updating the model can become very small during training, making it difficult to learn long-range dependencies.

LSTMs address this issue by incorporating mechanisms that mitigate the vanishing gradient problem, enabling the model to retain relevant information over longer sequences. They achieve this through the use of gating mechanisms, including a forget gate that selectively discards irrelevant information and retains useful context. Additionally, LSTMs can update their internal state and preserve short-term context like standard RNNs, while also maintaining long-term dependencies. This allows LSTMs to model both local sentence structure and broader contextual relationships, such as linking ideas across multiple paragraphs.

Despite their advancements, RNN-based models still struggle to capture very long-range dependencies and are inefficient when it comes to parallelizing training. These challenges led to the development of attention mechanisms by Bahdanau et al. [6]. Initially introduced in the context of machine translation, attention mechanisms enable models to dynamically focus on different parts of the input sequence when generating each token in the output. Rather than compressing an entire sentence into a single fixed-length vector, attention allows the model to assign varying weights to input tokens based on their relevance to each output token. This approach enables the model to consider the entire input sequence simultaneously, avoiding the cumulative errors that can occur when information is passed step by step, as in the previous RNN-based models like LSTMs. Figure 2.2 shows a visualization of the attention mechanism, where each word is assigned a value that reflects the degree of attention it receives from a

previous token in the sequence.

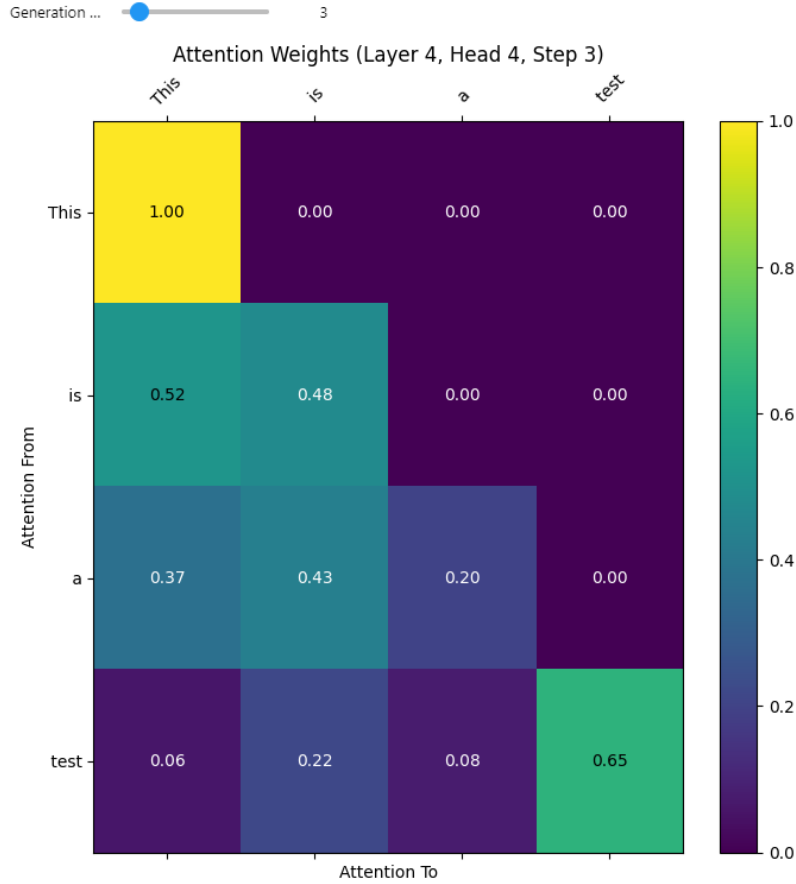


Figure 2.2: An example of a visualization method of attention. This shows that the tokens can not generate values further than themselves for causal LLMs. For each row, it shows how much attention is given by each previously generated token to the current generated token.

Further advancements were achieved with the introduction of the Transformer architecture by Vaswani et al. [66], which relies entirely on attention mechanisms and eliminates the need for recurrence. The Transformer employs self-attention to capture dependencies among all tokens in the input sequence simultaneously, enabling highly parallelized computation and more effective modelling of long-range relationships. This architecture comprises encoder and decoder stacks, each consisting of multiple layers of multi-head self-attention and position-wise feed-forward networks, augmented with residual connections and layer normalization. By incorporating positional encodings to preserve the order of tokens, Transformers overcame the sequential processing bottleneck inherent in RNN-based models and established new performance benchmarks across a wide range of natural language processing tasks. The success of Transformers paved the way for influential models such as BERT and GPT.

To better understand the internal structure of the Transformer, Figure 2.3 provides a high-level schematic of its architecture. As shown in the diagram, input tokens are first embedded into vector representations and enriched with positional encodings to provide the model with information about the order of tokens, since the self-attention mechanism itself lacks any notion of sequence. These embeddings are then passed through multiple layers, each consisting of a self-attention mechanism followed by a p feedforward network. In the encoder, tokens can attend to all others in the sequence. In the decoder, causal masking is applied to ensure that each position can only attend to previous positions, which is essential for autoregressive generation.

The final output from the decoder is used to predict the next token in the sequence.

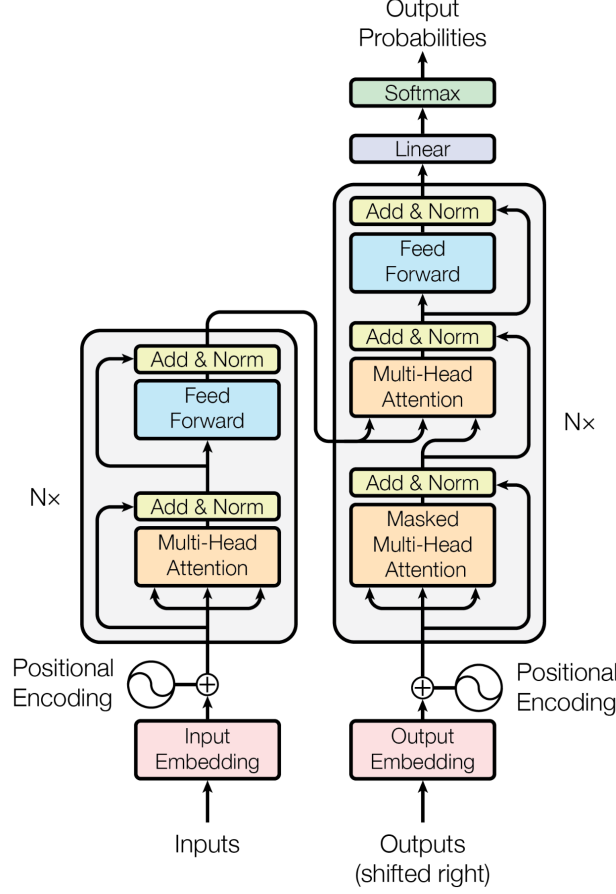


Figure 2.3: The Transformer architecture from Vaswani et al. [66]. The left side represents the encoder, which transforms input token embeddings into contextual representations by capturing relationships between all tokens in the input. The right side is the decoder, which generates the output sequence one token at a time using masked self-attention with the encoder’s outputs to predict the next token.

BERT (Bidirectional Encoder Representations from Transformers)[17] is an LLM model, as previously mentioned, that leverages the Transformer architecture with an emphasis on the encoder component to better capture the meaning of words in context. It achieved state-of-the-art performance on tasks such as sentiment analysis, question answering, and named entity recognition. Through pre-training on large text corpora using masked language modelling and next sentence prediction objectives, BERT developed deep contextual representations that reflect nuanced meanings based on surrounding context. Its bidirectional approach enabled it to outperform earlier models across a wide range of natural language processing benchmarks, sparking extensive research into Transformer-based encoder architectures. In contrast, the GPT (Generative Pre-trained Transformer) model introduced by Radford et al. [52] is built around the decoder component of the Transformer and employs a unidirectional language model trained to predict the next token in a sequence. This autoregressive design makes GPT particularly effective for text generation tasks such as story writing, summarization, and dialogue systems. The architecture gained widespread popularity with the introduction of ChatGPT, a user-facing application that enables natural language interaction with a large language model.

2.1.3 Hallucination

These revolutionary language models, despite their impressive capabilities, are not immune to limitations. One of the most critical challenges they face is hallucination, a phenomenon where the model generates seemingly plausible yet factually unsupported content. Understanding and mitigating hallucination is an active area of research, as it is essential for ensuring the reliability and trustworthiness of LLM outputs. Huang et al. [27] have surveyed recent principles and challenges and divided LLM hallucinations into two types.

The first type of hallucination happens when an LLM generates a faithfulness inconsistency. This means that it seemingly did not understand what was asked, or there is an error in the provided reasoning. An example would be:

User prompt: Translate the following to Dutch: What is the capital of Belgium?

LLM response: The capital of Belgium is Brussels.

This type of hallucination is often referred to as a faithfulness error, where the model’s output deviates from the intent or requirements of the input prompt. Even though the factual content of the answer (“The capital of Belgium is Brussels.”) is correct, the model fails to understand the specified task, in this case, translation, demonstrating a lack of alignment between input and output. Such errors highlight limitations in the model’s task comprehension or prompt parsing abilities.

The other type of hallucination is factual inconstancy. In this case the model will understand what the user wants, but returns a factually incorrect response. For example:

User prompt: What is the capital of Belgium? **LLM response:** The capital of Belgium is Amsterdam.

In this instance, the model correctly identifies the nature of the task, answering a factual question, but provides an incorrect answer. This type of hallucination is particularly concerning because it can go unnoticed if the user assumes the model’s output is accurate. Factual inconsistencies may arise due to limitations in the model’s training data or incorrect generalizations.

Hallucinations can be categorized into three distinct types, as outlined by Zhang et al. [74]. The first is input-conflicting hallucination, which closely aligns with the concept of faithfulness errors. In this case, the LLM generates output that is irrelevant or inconsistent with the user’s input. The second category is fact-conflicting hallucination, which, similar to the previously discussed factual inconsistency, involves the generation of factually incorrect information. The third type is context-conflicting hallucination, wherein the model produces output that contradicts content it previously generated within the same conversational or textual context. As both input- and context-conflicting hallucinations represent forms of faithfulness-related inconsistencies, this study adopts the classification framework proposed by Huang et al. [27].

Faithfulness errors are easily detectable, as the output is not what the user requested and thus the user will not be satisfied with that response. While it can be frustrating for the user, the impact of these hallucinations is generally harmless as they are immediately noticeable and unlikely to mislead. Users can quickly recognize that the model has not performed the intended task and are therefore less likely to act on the erroneous output. In most cases, the user will simply rephrase the prompt or attempt the query again, prompting the model to correct itself.

In contrast, factual hallucinations and fabricated content are more insidious. Since the output may appear fluent, confident, and contextually appropriate, users may not question its validity especially if they lack prior knowledge of the subject. This makes such hallucinations significantly more dangerous, as they can propagate misinformation or lead to faulty conclusions without immediate detection. To solve this issue, the model would have to be able to provide the sources for that information, so the user can check and judge that source for themselves and not rely purely on the LLM output.

2.1.4 Ethical Considerations

Alongside hallucinations causing possible harm, Bender et al. [7] also summarized other ethical implications of LLMs. First there is the environmental cost of LLMs. Training large-scale LLMs requires significant computational resources, often resulting in substantial energy consumption and carbon emissions. This raises concerns about the sustainability of current development practices. The trade-off between model performance and environmental impact becomes more pronounced with each generation of increasingly larger models, such as GPT-style architectures with hundreds of billions of parameters. In response to these concerns, some developers have introduced more efficient alternatives, such as the Llama family of models, which include smaller variants designed to reduce computational demands while maintaining competitive performance.

Another concern that Bender et al. [7] have is that because training an LLM can require a lot of training time on multiple high-end GPUs, the technology would not be available to everyone. GPT-3 for instance required several days of training on a V100 GPU cluster provided by Microsoft [10]. This could limit access to this promising technology to companies or countries that can afford the high costs of training. As a result, LLMs might not perform well in certain languages, making them less accessible to people who speak languages that weren't well represented in the different LLM companies.

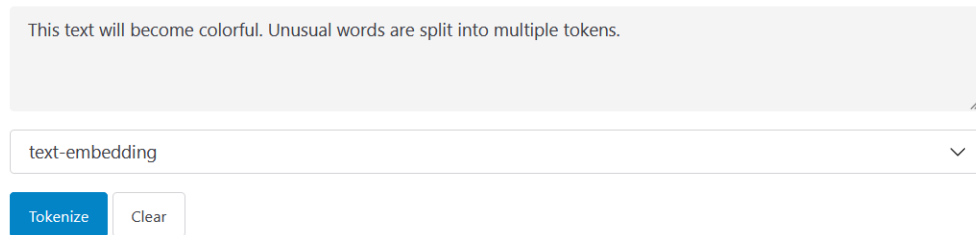
Just like previously discussed, Bender et al. [7] have also found that hallucinations in combination with biases from the model can cause substantial harms, including stereotyping and increases in extremist ideology. By having biased data during the training stage, the outputs could also reflect and even amplify these biases, potentially reinforcing harmful stereotypes and spreading misinformation. This highlights the critical need for careful dataset curation and bias mitigation strategies in the development of large language models, as well to detect hallucinations and present the user with sources so they can investigate potential biases themselves.

2.2 Improving Transparency

This section explores various XAI methods, ranging from simple techniques, such as color-coding individual tokens, to more advanced approaches like leveraging attention weights from transformer architectures or using mathematical tools like SHAP and LIME to quantify the influence of each token on the model's output. During the investigation of these techniques, a recurring concern in the literature was their lack of user-friendliness. Many of the current methods are difficult to interpret, especially for non-technical users. Given that LLMs have become widely adopted by the general public, this presents a significant challenge. The final part of this section highlights that the main issue with these language models is that they can output wrong information and users tend to not be aware of this.

2.2.1 Tokenization Visualization

One of the earliest and most straightforward techniques to enhance model transparency is to split the output into individual tokens and assign each a distinct colour. An example of this visualization approach is presented in Figure 2.4. This visualization clearly demonstrates that language models do not operate on full dictionary words, but rather on tokens. By displaying tokens in different colours, users can intuitively understand how a model decomposes text and processes language at the sub word level. Although simple, this technique reveals a core mechanism of modern language models: they operate on tokens, which may correspond to entire words, prefixes, suffixes, or word fragments, depending on the tokenizer used. Furthermore, token colouring can help uncover biases or unexpected behaviours in model outputs. For example, if the model generates an unusual word, tracing the coloured tokens can assist in identifying whether the issue arose from a rare token, an ambiguous prefix, or a problematic split.

Your text

The 17 tokens

This text will become colorful . Un usual words are split into multiple tokens .

Figure 2.4: Laforge [36] developed an interactive tool to visualize how various large language models tokenize input text. This shows that for this encoding, spaces are attached to words and some words are split into multiple tokens.

2.2.2 Explainability With Feature Attribution

To explain the inner workings of black-box models, XAI researchers have developed various methods to reduce the opacity of neural networks. However, the applicability of these techniques often depends on the specific type of model used. A number of XAI methods have been reviewed in recent literature, including a comprehensive survey by Ali et al. [3], which presents recent approaches for enhancing transparency at the data, model, and post-hoc levels, where explanations are generated after the model has been trained. Neural network-based classifiers, which categorize input data into predefined classes, typically utilize multiple features as input. These features can be assigned a "feature importance" score, indicating their relative impact on the model's output. Among the most widely used methods for calculating feature importance are LIME and SHAP. These techniques work by perturbing input features and observing the resulting changes in model predictions. Features that induce greater shifts in the output are considered more influential. The importance scores are often visualized using bar plots, which illustrate the impact of the most significant input features on the final prediction. An example of such a bar plot is shown in Figure 2.5.

These post-hoc explanations can provide users with greater insight into which features the model considers important. However, interpreting these explanations still requires some understanding of how classifiers function, particularly, what features are and what it means for them to be deemed important. In contrast, other types of models, such as image classifiers, often require less prior knowledge. For example, visualizations of Shapley values on images intuitively highlight the regions that contribute most to the model's decision. Figure 2.6 illustrates an example of applying Shapley values at the pixel level to identify the most influential regions in the prediction.

In addition to illustrating the importance of input features after training, some techniques aim to reveal how training data influences model biases. One straightforward approach is the use of a confusion matrix, which can highlight systematic misclassifications and biases toward certain classes. For instance, if a model frequently confuses one class with another or disproportionately predicts a particular class regardless of input, this may indicate bias in the training data or an imbalance in how the model represents different categories. Such patterns can suggest that the model has overfit to more frequently occurring classes or neglected underrepresented features, providing a foundation for identifying and mitigating bias.

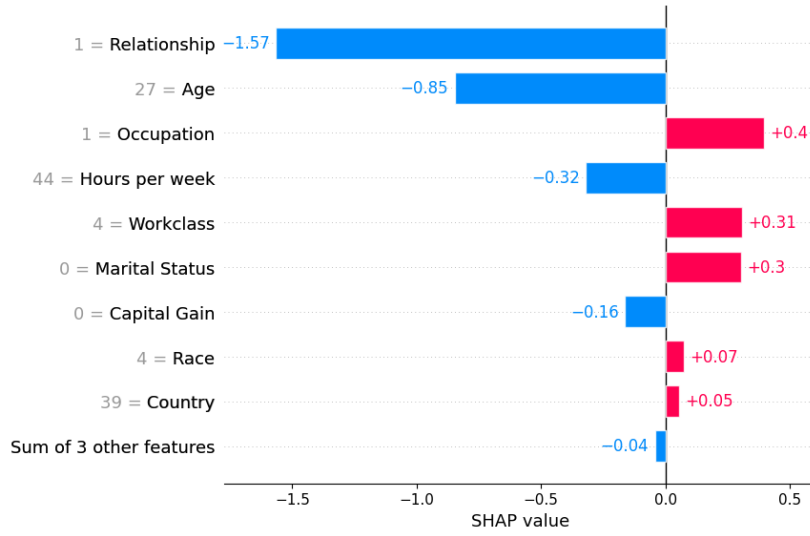


Figure 2.5: A SHAP local bar plot, showing the changes of the predicted income that each input feature caused. The blue bar negatively influence the income, meaning that the predicted income will be lower when those values are higher. These Shapley values are calculated by changing those input features slightly and calculating the influence of that change on the predicted outcome. Adapted from Lundberg et al. [41]

2.2.3 Attention as Explanation

Another relatively simple technique to improve transparency is visualizing attention values through colour coding. This was found to be a popular method by Parra et al. [50]. When a user selects a token, previous tokens that receive higher attention weights during the prediction of that token are highlighted with greater intensity. However, this approach becomes more complex due to the multi-headed attention mechanisms used in modern large language models. Each attention head captures different patterns of importance, focusing on distinct aspects of the input. While it is possible to allow users to select individual heads and inspect their corresponding attention distributions, this requires a high level of expertise to interpret meaningfully. Attention patterns often follow complex and abstract rules that cannot be easily translated into human-readable explanations. Nevertheless, this visualization technique can be valuable for model developers in diagnosing unusual behaviours. For instance, a head that consistently assigns high attention to the first token in a sequence may indicate undertraining or inefficient utilization of the model’s attention capacity. Parra et al. [50] investigated various methods for visualizing simple single-headed attention and concluded that, except in tasks requiring identification of the top n most attended words, the most widely used visualization technique is varying background colour intensity. The different methods are shown in Figure 2.7.

Although some argue that this method does not effectively enhance user trust, it can be a valuable tool for developers of language models. Attention values may reveal model biases or indicate oversimplification, such as when attention consistently focuses on the same token position, signalling that additional training may be needed. Niu et al. [48] found that for those reasons, attention brings more interpretability to deep learning models. Mohammadkhani et al. [46] used attention patterns from underperforming models to identify contexts in which the models struggled, thereby facilitating targeted improvements. Nevertheless, interpreting attention values remains challenging due to the complexity of multiple heads, layers, and token interactions, especially for users who may lack a deep understanding of how attention mechanisms function.

Vig [68] enhanced a visualization tool originally developed by Jones [33], which illustrates the

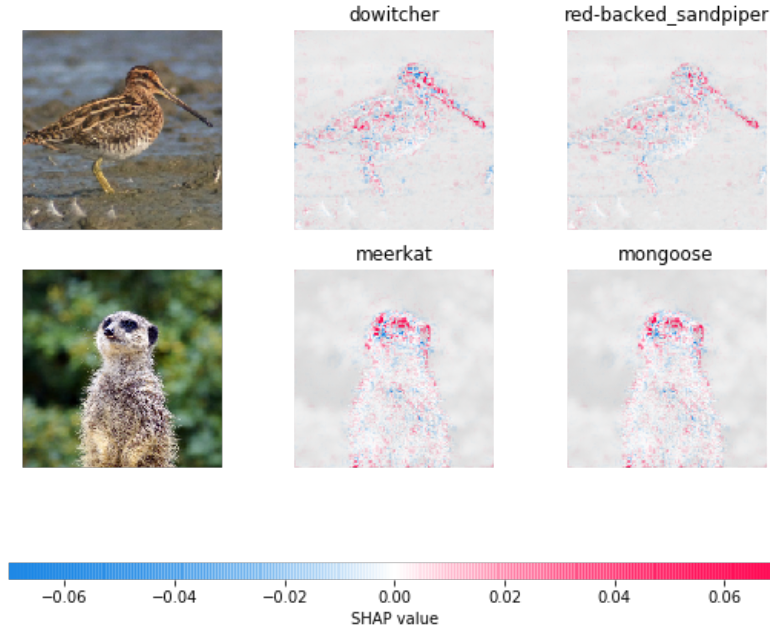


Figure 2.6: The image explanation using the SHAP gradient explainer to highlight pixels with high and low Shapley values from SHAP’s VGG16 explanation example [40]. The red pixel have a positive impact on the prediction, while the blue pixels mean that these regions are not likely to be similarly looking to the prediction. In this case the model predicts the two images to be a dowitcher or red-baked sandpiper for the first image and a meerkat or mongoose for the second image.

attention one token assigns to another within a sequence, given a selected head and layer. While the original tool was limited to the older encoder-decoder transformer architecture, Vig’s revision extended compatibility to decoder-only models like GPT-2 and encoder-only models such as BERT. The updated tool is shown in Figure 2.8.

This type of attention visualization tool is primarily intended for adjusting and improving models. While it offers insights into the “black box” from a developer’s perspective, it is of limited utility for most end-users. Moreover, interpreting these visualizations requires substantial prior knowledge. As such, this form of XAI is not user-centered. Although it is valuable for technical users, it should be supplemented, or potentially replaced by techniques designed to accommodate non-technical users, who represent the primary audience of modern LLM-based chat applications.

Maruthi et al. [43] propose using model-agnostic techniques such as LIME and SHAP to estimate token importance by perturbing input tokens and observing the resulting changes in model output. However, this approach presents challenges when applied to modern LLMs, which often utilize a ‘top-p’ sampling strategy. In this approach, only tokens whose cumulative probability exceeds a predefined threshold are considered for generation. While this setting ensures a degree of randomness and diversity in output, it also introduces non-determinism, meaning that identical inputs can yield different outputs. As a result, applying techniques like LIME or SHAP would require modifying the model’s inference behaviour, potentially diverging from the user experience in real-world applications. Furthermore, even if such techniques were employed, Volkov and Averkin [69] argue that they would still not constitute a user-centred explanation. These methods may lack the intuitiveness required for non-technical users to fully comprehend the explanation and develop appropriate trust in the system.

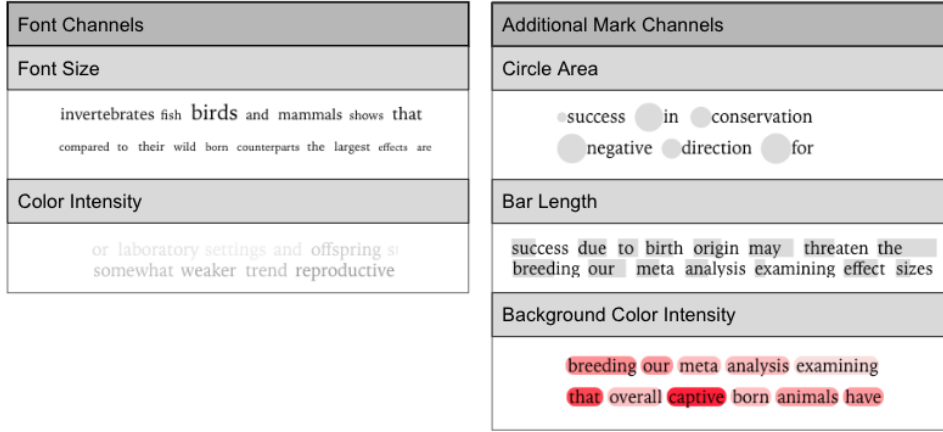


Figure 2.7: The figure illustrates various techniques for visualizing attention distribution within a sentence under a single-headed attention mechanism. Techniques in the left column modify the font to reflect attention weights, while those in the right column apply additional visual markers without altering the font itself. The original image is sourced from Felix et al. [20].

2.2.4 User Expectations

Despite the widespread popularity of LLMs, even among mainstream audiences, XAI techniques for these models remain underdeveloped. A recent survey by Cambria et al. [11] found that, at the time of publication, only a limited number of research efforts had focused on developing explanation methods specifically for LLM-based systems. The authors further emphasize that the presentation layer, how explanations are delivered to users, should be regarded as equally important as the technical explanation of model behaviours. Given the increasing presence of non-technical stakeholders in the use of LLMs, the design of XAI methods must prioritize user-friendliness as a central concern.

Another critical concern for developers of XAI in LLMs is their ability to convincingly present incorrect outputs as correct. Ajwani et al. [2] argue that LLMs generate persuasive explanations even when justifying erroneous answers, thereby making incorrect responses appear plausible. Consequently, directly querying the LLM about its reasoning process is not a reliable method for enhancing transparency, as the explanations may be compelling yet inaccurate. This occurs because the language model relies more heavily on lexical cues than on logical reasoning, producing responses that conform to learned linguistic patterns rather than grounded truth.

While developers understand that LLMs function primarily as pattern recognition systems designed to predict the next token rather than to perform logical reasoning, users of popular LLM-powered chat applications may be unaware of these limitations. To protect users from misleading responses and potentially adversarial explanations generated by LLMs, González Barman et al. [23] recommend providing users with adequate training and clear guidelines. They emphasize that everyday users are prone to misunderstanding the intended capabilities of these systems, which can result in errors, misinformation, and confusion.

Similar to Ajwani et al., Saba [57] argues that LLMs do not truly understand language but instead select tokens that are lexically most appropriate. They present several cases in which the model’s responses are lexically coherent but factually incorrect, such as:

User prompt: Since ‘Madrid’ = ‘the capital of Spain,’ can I replace one for the other in the sentence “John thinks Madrid was not always the capital of Spain”
LLM response: Yes, you can replace ‘Madrid’ by ‘the capital of Spain’ in “John

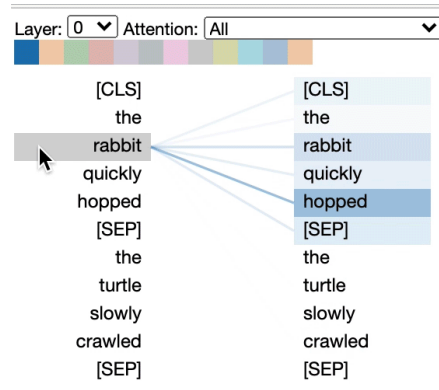


Figure 2.8: A BertViz visualization [67]. The colours allow the user to select an attention head. In combination with the selected layer, the attention values are visualized by drawing lines between the tokens that receive attention and tokens that can pay attention to the other column. Thicker lines indicate a higher attention value. The colours on the right column show the attention for every head on that layer, where brighter colours have a higher attention.

thinks Madrid was not always the capital of Spain.” The new sentence will be “John thinks Madrid was not always Madrid.”

Such examples effectively illustrate the limitations of LLMs to non-technical users and serve as ideal case studies for user guidelines. Because these examples do not require knowledge of the underlying mechanisms of LLMs, they are especially valuable for non-technical stakeholders, a point underscored by Cambria et al. [11] as critical given the widespread adoption of LLM-based chat applications.

2.2.5 Improving User Interaction

Effectively communicating how LLMs function and what users can reasonably expect from them is essential to promote appropriate use by non-technical users; however, a gap remains between human confidence in LLM-generated answers and the models’ actual confidence [61]. Datta and Dickerson [16] argue that existing approaches to explaining LLMs are overly technical and lack a user-centred perspective. Current methods often rely on quantitative metrics, such as accuracy scores or attention weights, which, while informative to developers, offer little value to general users. Instead, they advocate for an XAI framework that fosters cognitive engagement. According to their findings, providing guidelines and illustrative examples alone is insufficient; users must actively reflect on the model’s outputs and critically assess whether those outputs should be trusted. These insights are in accordance with the general consensus that XAI should aim to reduce overreliance on the AI systems [51]. While showing examples of hallucinations can help reduce overtrust in LLMs, this does not fully address the need for calibrated, appropriate trust, where users accept accurate outputs and question incorrect or conflicting ones.

Although there has been a call to develop more user-centred frameworks and accessible guidelines for non-technical stakeholders, quantitative explanation methods continue to advance. While such methods may not directly enhance user trust in LLM systems, they play a critical role in supporting developers by identifying biases, limitations, and areas for model improvement. Attention visualization is one such technique. Despite its complexity and limited interpretability for new or non-technical users, it can provide valuable diagnostic insights for developers. For example, attention maps can reveal whether a model is allocating attention appropriately across tokens or if certain attention heads are underutilized. If the model consistently focuses on the first word in a sentence, it may indicate that the model has not yet learned more nuanced language patterns and requires further training.

Azaria and Mitchell [5] proposed a method that enables an LLM to communicate its confidence in the accuracy of its outputs. Although this is a quantitative technique, it is comparatively easier for non-technical users to understand than, for example, attention visualizations. Their findings suggest that when an LLM is uncertain about the next token during generation, it may indicate that the model is already producing misleading or incorrect information. This uncertainty can be expressed as a low probability assigned to the predicted token, signaling a potential hallucination. For instance, consider the sentence:

Pluto is the smallest dwarf planet in our solar system.

This statement is factually incorrect, as Pluto is actually the second-largest dwarf planet in the solar system. The confusion likely arises from historical references that described Pluto as the smallest planet before it was reclassified as a dwarf planet. When generating the phrase “Pluto is the,” the model may predict “smallest” due to its prevalence in older texts. However, the model then faces difficulty in confidently predicting the continuation, as both “planet” and “dwarf planet” lead to inaccuracies. This reduced confidence can be communicated to the user, indicating that the sentence may contain hallucinated content.

While this approach highlights uncertainty in the model’s output, especially when conflicting information is present, it does not guide the user toward more reliable information or suggest what the correct answer should be.

2.3 Transparency By Finding Similar Training Data

Creating a user-friendly XAI method for LLMs that helps users understand when the model is uncertain of an output and what the correct output would be, could be done by showing similar training samples. These would allow the user to compare the generated answer to data created by humans and determine for themselves what to believe, or at least understand that the LLM could be wrong when the relevant training data shows that there are conflicting answers. This method aligns with the concept of ‘explanation by example’ as described by Lipton [39]. Here models justify their decisions by referencing similar cases in the learned representation space, and this could also be done for LLMs.

2.3.1 Similar Training Sample as XAI: Literature Example

There weren’t many papers about using the similar training samples as an XAI technique, but Wu et al. [71] highlight several user-centred XAI techniques with a strong focus on usability including the similarity search. In addition to SHAP-like methods, where input data is perturbed to observe the resulting changes in model output, and interpretability strategies involving components of the LLM such as attention weights or ‘top-p’ sampling parameters, they also discuss the potential of retrieving training data that is semantically similar to the model’s generated output. This approach could serve as evidence for how the model derived its response, assist developers in debugging the model, and help identify gaps in the training corpus.

That last method implemented by Wu et al. [71] relies on saving the embedded vectors of all training data and later comparing them to the embedded vector of a generated output. These vectors, produced by the LLM, are designed to encode the semantic content of text, with each dimension potentially representing a simple or complex concept. Because these embeddings are algebraically structured, allowing relationships between concepts to be maintained through vector operations, an effectively trained LLM can retrieve training data that aligns conceptually with its generated output. Presenting this data to the user provides factual, human-authored reference points that eliminate the risk of hallucination. Moreover, when sources are included with the retrieved data, users are empowered to assess the trustworthiness of the source material and form their own judgments, promoting appropriate and calibrated trust in the system’s responses.

Wu et al. [71] argue that retrieving training samples based on semantic similarity lacks a robust theoretical foundation and may fail to surface key training instances that are not semantically proximate to the test input. To illustrate this concern, they present an example involving arithmetic. Suppose the training set consists of Sample 1: (“1+1=”, “2”) and Sample 2: (“2+2=”, “4”), and the model is prompted with the test input “100+100=”. They contend that although the model may learn an arithmetic pattern from the training data, the embeddings of these samples may differ significantly, preventing the system from retrieving relevant examples through similarity search.

Despite this limitation, semantically similar data may still serve an important purpose: enhancing user trust. By presenting users with authentic, human-authored training samples that are similar to the model’s output, one can provide assurance that these data are not hallucinated. This comparison enables users to assess the reliability of the model’s response, potentially fostering a more appropriately calibrated level of trust.

2.3.2 Semantically Similar Text By Vector Embedding: Literature Example

Searching for similar data using embedded vectors is already a widely adopted technique in applications that prioritize semantic understanding over literal keyword matching. Rather than searching by exact words or numerical values, such systems allow users to query information by expressing ideas or concepts. For example, searching for the term “blue collar worker” allows the user to retrieve records related to plumbers, construction workers, and similar professions without the need of those keywords being present in the records. Bordawekar et al. [8] demonstrated an early implementation of a database application that leverages word embeddings to retrieve semantically related records based on natural language queries. More recently, Imdarkmode [29] introduced a semantic search engine for the card game Magic: The Gathering, enabling users to search for cards by describing gameplay strategies. For example, a query such as “early aggression” would return low-cost cards that support fast-paced tactics, even though none of the exact words may appear in the card descriptions. This capability is made possible through the use of vector embeddings that encode the underlying semantics of both the query and the database records.

Chen et al. [13] advanced this approach by integrating semantic similarity search into XAI systems. Their method utilizes a separate dataset, one that is built by data not necessarily seen during the LLM training and used for all different tested LLMs, to retrieve examples similar to the LLM’s output, aiming to provide more reliable explanations. However, this introduces a limitation: even if the retrieval mechanism perfectly identifies all relevant samples in the available dataset, it cannot guarantee coverage of all knowledge reflected in the LLM’s responses, particularly if the dataset itself lacks that information. Furthermore, the intended application of Chen et al.’s technique is to aid developers in mitigating hallucinations and improving model performance, rather than to foster more appropriate trust among non-technical end users.

The system would also retain all training data, enabling greater transparency and control over the dataset used for training the language model. This approach facilitates compliance with Article 14 of the EU AI Act [25], which mandates that proactive measures be taken to prevent the generation of illegal content by AI systems. Ensuring that no illegal material is included in the training data is a critical component of this requirement. Additionally, the database required for the similarity search mechanism can serve a dual purpose by providing a means to audit the training data, thereby verifying that the model aligns with legal standards.

There is a growing need for a human-centered approach to retrieving semantically similar training data for outputs generated by LLMs. Such an approach would allow users to view grounded, human-created data samples, which can be critically evaluated based on their original sources. While tools of this nature could also support developers in identifying hallucinations or knowledge gaps in LLMs, their primary objective should be to enhance appropriate trust among end

users. Specifically, users should be equipped to recognize when an LLM is uncertain due to exposure to conflicting training data. In such cases, user trust in the model’s response should be moderated accordingly. Ideally, users would be able to review the retrieved data and consciously evaluate which sources are most reliable. This thesis will therefore investigate the feasibility of such a tool and assess whether its use improves users’ ability to develop appropriate trust in LLM-generated content, particularly by reducing trust in responses based on conflicting data and reinforcing trust in outputs supported by consistent evidence.

2.3.3 Database Search

To search for similar data in a database that can contain terabytes of texts, efficient algorithms will be necessary. For this thesis the edit distance, Jaccard distance and the cosine distance were researched as the first two are easy and intuitive ways to calculate the distance between strings and the third can use the internal representation of the text that an LLM uses.

The edit distance, or also called the Levenshtein distance[37] calculates the number of edits required to transform the first string into the second, where edits can be deletions of characters, or tokens in this case, or inserts or substitutions.

Another example of a similarity metric is the Jaccard similarity. This method calculates the number of common characters or tokens between two strings. It does so by taking the intersection of the two sets and dividing it by their union. This results in the fraction of tokens that both strings have in common. The Jaccard distance can also be calculated easily as 1 minus the Jaccard similarity. Figure 2.9 shows two sets, A and B, which, in this case, represent two sentences converted into sets of tokens. The Jaccard similarity is the number of tokens that both sets have in common (the blue region), divided by the total number of unique tokens in both sets.

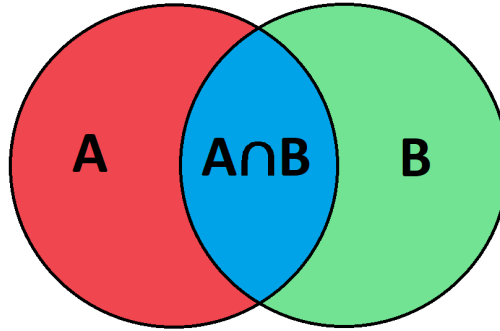


Figure 2.9: A visualization of the Jaccard similarity, with set A in red, set B in green and the intersection of both sets in blue. When the sets are seen as sentences for instance, the intersection would be the characters that both sets have in common. The Jaccard similarity would then be the percentage of the characters in common out of the total characters in both sets.

When looking to find the most similar pieces of training data based on edit distance and Jaccard distance, the Locality Sensitive Hashing (LSH) method can be used [31]. This is a technique that efficiently searches for approximate nearest neighbours by using hash functions. These functions are chosen so that similar strings are more likely to be hashed into the same buckets. By hashing the strings with multiple different hash functions, each mapping the data into separate hash tables, the probability of similar strings colliding in at least one of the tables increases. This multi-hash approach improves the recall of the system, ensuring that similar items are more likely to be retrieved during the search. In practice, this means that for a given query string, the algorithm only needs to compare it against the items in the same buckets across all hash tables, rather than against the entire dataset, drastically reducing computational cost and time. In the case of Jaccard distance, this algorithm is frequently used together with

MinHash [9], a hash function that converts sets of tokens into signatures that approximate the Jaccard similarity between them.

The final considered distance metric is the cosine distance. As LLMs can represent sentences as vectors, where the dimensions capture the underlying meaning of the words, cosine distance was also researched. This distance measures the angle between two vector representations of texts. Reimers et al. [54] successfully used this distance to measure the similarity between two vector embeddings.

Using this similarity metric while searching in a large database is a more novel approach compared to traditional string-based metrics such as edit distance or Jaccard similarity. While most databases and corresponding search algorithms are designed for strings or low-dimensional numeric data, searching in high-dimensional spaces remains relatively underexplored in comparison to the previously mentioned distance metrics. A recent development in this field is Faiss (Facebook AI Similarity Search), a library by Douze et al. [18] that includes multiple optimized nearest neighbour search algorithms. This library also supports searching based on cosine distance between vectors to find the closest one. It does this by normalizing the vectors, so the length information is discarded while the angular relationship is preserved. This normalization enables the algorithm to be more efficient, as it reduces the amount of detail that needs to be stored and compared.

Chapter 3

Implementation

An LLM is constructed by integrating multiple techniques that work together to process a given context and predict the next segment of text. This thesis explores various methods to increase the transparency of these components and to explain the decision-making process of the LLM. The rationale behind specific design choices is also discussed, as there are several approaches to the internal techniques like tokenization, attention mechanisms, fixed-size input handling, and other implementation differences among available LLMs. These factors were carefully considered to select the most suitable technologies and LLM-architecture to develop the similarity search tool and conduct a user study to validate its effectiveness.

For this thesis there are three methods used to understand the current progress of XAI for LLMs. The first method visualizes the different tokens by giving each distinct token a separate colour, this uses the previously discussed manner, where the tokens are given a background with different saturation levels. The second method involves visualizing the attention that the LLM gives to each previous token when generating a new one, by leveraging the hidden values from within the LLM. And the last method was created for this research to find out if showing similar training samples to users has a positive impact on appropriate trust.

3.1 Model choices

In order to be able to implement the discussed XAI techniques, the chosen LLM has to meet some conditions. The visualization of the different tokens requires the application to have access to the tokenizer that was used by the model, together with the encoding and decoding function to change text into token identifiers and vice versa. The attention visualisation requires that the chosen LLM can return these attention values when it generates the accompanying output. And for the last method where similar training samples are shown, the embedding method from the LLM has to be able to be used by the application as well, in order to save the vectors representing the input to a database. Because of all those requirements, not every type of LLM architecture is viable. This section will discuss the reason why the Llama architecture was eventually chosen, and which others were also considered.

3.1.1 N-Grams

When selecting a model on which to implement these transparency techniques, it is important to consider the specific advantages and limitations of each model type. As previously mentioned, N-gram models generate tokens based on simple statistical likelihoods of token sequences [12]. This offers a high degree of interpretability, as the most common combinations can easily be retrieved from the training data and used to show the reason why the new token was chosen. However, this approach is not commonly used in modern LLM applications, which prioritize richer contextual understanding. The outputs of n-gram models are generally of lower quality

because they do not account for the semantic meaning of words. This significantly limits their applicability, particularly in the context of conversational LLMs that must respond to complex and nuanced user inputs.

3.1.2 BERT

The BERT-based models outperform previous architectures such as RNNs and LSTMs, which is why the latter were not considered for this study. [17]. Transformer-based models are currently superior, as they can be trained in parallel, a key advantage over RNNs, which require the previous token to calculate the next one. But both the transformers and RNNs are still difficult to interpret, given that they also rely on neural networks for decision-making.

The primary limitation of BERT is its difficulty in text generation. While it excels at encoding contextual information (i.e., understanding the meaning of a sentence), BERT was not designed or trained to generate new tokens [65]. It learns context by masking certain tokens and predicting which tokens should fill those gaps, allowing it to understand surrounding words and, for instance, detect homonyms. However, generating new tokens would require masking all subsequent tokens, a task for which BERT was not trained. As a result, BERT struggles with this process. While there have been efforts to enhance BERT's performance in generation tasks, the GPT architecture gained more traction, shifting the focus towards leveraging and refining GPT for text generation.

3.1.3 GPT

For chat-based LLMs, the GPT architecture is currently considered the gold standard, with ChatGPT being one of the most popular LLM applications [34]. However, after the release of GPT-2 in November 2019, subsequent GPT models were not accompanied by detailed documentation, preventing other developers from replicating the models. As a result, newer models are closed-source and unavailable for use in this research. Without access to key components, such as the attention values or the embedding table that converts tokens into vectors, many of the methods discussed for improving transparency cannot be applied to these newer models.

Even GPT-2 was not fully open-source, as the weights and code were not directly released. Other developers were able to recreate the model by following OpenAI's publications, which included instructions for constructing the architecture and attempting to replicate the training data.

However, GPT-3.5, the version initially used in the popular ChatGPT application, was built with an undisclosed architecture, and its training data remains private. The performance of this model has become the benchmark for LLM applications, shaping public expectations. Unfortunately, it is not possible to retrieve vector encodings of the input, display attention values, or colour the tokens, as these elements are locked behind the closed-source code. The only available interaction is through API calls, where developers send a prompt and receive a response. As the source code is inaccessible, newer GPT models cannot serve as the basis for testing XAI techniques in this thesis. As the current newest model, GPT-4.1 and everything between this and GPT-3.5 are not open-source, it is not expected to change.

3.1.4 Llama

A more open-source alternative that still performs well is the LLaMA architecture from Meta, presented by Touvron et al. [64]. However, there has been some debate regarding LLaMA's use of the term "open-source," as the license may impose restrictions on specific use cases or commercial applications with large user bases. Nevertheless, the weights and the complete architecture are open-source, and the accompanying papers for each new generation of LLaMA models describe the training data. This makes it theoretically possible to train a model on the same data and compare its performance with that of the original. For this research, LLaMA is particularly valuable, as it eliminates the need to train a model from scratch, providing direct

access to the pretraining data. Additionally, the availability of the architecture allows for the extraction of attention values, and the tokenization process can be used to convert text to token IDs and vice versa. This enables the visualization of tokens in different colours and the display of calculated attention values for each selected token.

3.2 Choosing the right model

The initial choice of model was a complete tutorial on building an LLM by Andrej Karpathy[35], which would provide full control over the training data and allow for the testing of generated answers to identify similar data. Additionally, the attention values could be retrieved and visualized, along with the exact representations of the prompt before predicting the next token. Since an LLM converts strings into tokens and then into a vector embedding, accessing the embedding table is crucial to ensure that the conversion from text to vector occurs consistently during both the search for similar training data and token prediction.

However, the main issue with this model was its performance. Initially, it was trained on the complete works of Shakespeare to learn English. However, this dataset consisted of old English, which lacked the relevant information typically expected from a modern chat-based LLM application. As a result, the Shakespeare data was replaced with the SQuAD dataset, a large collection of question-and-answer pairs created by Rajpurkar et al. [53]. This dataset was more suited for the intended user study, as it provided a variety of questions and corresponding answers. Despite this adjustment, the limitations of the small GPT-based model became evident: it was too small to generate coherent, meaningful responses. While the model could produce sentences that followed English word order, the sentences lacked coherent meaning.

To ensure that the LLM output is comprehensible and does not undermine trust by failing to communicate in basic English, a Llama model was chosen. It was finetuned on data that is newer than the base model’s training cutoff, so it was certain that the finetuning data contained new information to the model and any correct answer would at least partially have used some of that finetuning data. The relevant data could then be gathered from the database, showing human-made texts next to the LLM generated output about the same topics. Using the pretrained data was not feasible for this research, as the volume of pretraining data (4.749 TB), as shown in Table 3.1, was too large, and the specific datasets were difficult to reacquire. Although the paper specifies which datasets were used, these may have been updated since the model’s training. For this reason, the Llama model was fine-tuned with data from NASA projects, the 2024 Academy Awards, and the 2024 Nobel Prize winners. Since the Llama 2 model’s training data cutoff was in 2022 and Llama 3’s was in 2023.

Table 3.1: Pretraining dataset composition for the LLaMA model [64]. Sampling proportions and number of epochs are reported along with the corresponding disk size. The total disk size is 4.749 TB.

Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

At the beginning of the research, Llama 2 was used because it was well-documented, while Llama 3 had just been released. The larger model allowed users to ask more general questions and interact with it, receiving responses that made sense in context. These answers referred back to the prompt and were grammatically correct, unlike the NanoGPT model, which often

generated sentences that were grammatically incorrect and incoherent. Despite Llama 3’s superior performance out of the box, fine-tuning it on new data proved challenging. The model struggled to learn new, uncorrelated sentences and frequently mixed them up when responding to prompts about specific sentences. Another issue was its inability to correctly generate the end-of-line token, often repeating the last few tokens before the token limit stopped further generation. While Llama 3 could be used to start visualizing token colours (where each distinct token would be assigned a different colour) and attention patterns, it was not suitable for a user study. The responses could not be guaranteed to meet the required quality standards for building trust, and the model’s performance was also relatively slow. It typically required between two and fifteen minutes to generate a response, making the user study too lengthy for practical use.

Later, Llama 3.2 3B was used. This newer version of Llama was the first in the third generation to include a smaller 3-billion parameter model, compared to the 7-billion and 8-billion parameter models in Llama 2 and Llama 3, respectively. The new model can still generate sensible answers and is also able to stop generating once the prompt has been answered. It is significantly faster, producing responses in just a few seconds, as opposed to the several minutes required by Llama 2. This speed is partly due to the Unsloth library[15], which is used to load the Llama model. The library enhances performance by employing various techniques, including quantization, which reduces the number of bytes required for the model’s weights, resulting in faster calculations, though with a slight reduction in precision. Despite the slight decrease in accuracy, benchmark scores show minimal impact, and this change is nearly imperceptible during a user study, as the model still provides sensible, grammatically correct answers.

3.3 Finding similar data

The first iteration of trying to find similar training data involved encoding all the training data into token IDs. This was done when using the NanoGPT model with the Shakespeare dataset. The goal was to identify similar context windows that the model might have encountered during training. With a context window size of 256 tokens, every possible sequence of 256 tokens in the training data was stored in the database. These sequences, known as k-shingles, caused the database to grow rapidly. Therefore, optimizations were necessary to make the search feasible for use in a user study.

One optimization strategy involved eliminating database entries that were unlikely to be similar by examining the last two tokens. If either of these tokens matched the last or second-to-last token in the context window being compared, it was considered a candidate; otherwise, it was skipped. This approach used an XML file, where each token in a directory represented the last token in the context of a database entry. This filtering technique, based on the last two tokens, reduced the search time dramatically, while still maintaining reasonable accuracy. However, it was possible that highly similar data might be overlooked if only the last two tokens were different. Nonetheless, for this use case, the time savings outweighed the potential loss in accuracy, as it was more important to find reasonably similar data quickly.

To compare the similarity of context windows, there are multiple methods of calculating the similarity of two strings. The one first used for this thesis was the edit distance. This approach did not yield great results, which was partly due to the NanoGPT model’s poor response generation. And as previously discussed, the Jaccard similarity is also a possible metric to compare the similarity of two strings.

The problem with these similarity metrics is that they do not take the meaning of words into account. These just calculate the amount of similar tokens, but will for example not realize that synonyms should also be close, as the tokens representing the words won’t be close, but the meaning of the sentence could be nigh identical. In order to solve this issue, the vector embeddings that some LLMs internally generates were used. When the LLM provides outputs of a sufficiently high quality, it is reasonable to assume that the embedding the model

learned effectively kept semantically similar sentences close to each other. By then calculating the distance between the vectors representing the entire text, the semantic distance could be calculated, showing how close or far texts are in terms of meaning.

Various methods of pooling the tokens and comparing distance metrics were tested. It was found that mean pooling combined with cosine distance yielded the best results. To test this, three different sentences were used: two that had the same meaning but were worded differently, and one control sentence that did not share the same subject as the first two. Using this combination, sentences 1 and 2, which had similar meanings, showed a high percentage of similarity, while sentence 1 and the control sentence, which had different subjects, showed a very low percentage of similarity. Figure 3.1 shows the comparison between the max pooling, absolute value pooling, and mean pooling techniques, each applied to two similar sentences and then also to one of the original sentences paired with a control sentence.

Max pooling selects the maximum value in each dimension across all word embeddings in a sentence, trying to preserve the most extreme features. Absolute value pooling takes the feature with the highest absolute value in each dimension, regardless of its sign. This technique is designed to retain the most pronounced difference, whether positive or negative, in each dimension. The motivation is to preserve the dimension with the largest deviation from zero, under the assumption that this reflects the strongest distinguishing signal in that feature space. Unlike max pooling, this method doesn't favour only positive activations, and it may help capture contrasts or strong features that would otherwise be overlooked. This technique was developed for this application, as it can be especially useful when contrasting semantically different sentences. It was however replaced with the mean pooling technique due to it performing better. Mean pooling computes the average of the word embeddings across each dimension. This approach captures the overall, smoothed semantic content of the sentence by balancing the contribution of each word. It tends to provide a more stable and general representation but may dilute strong signals or fail to emphasize rare but important features if they are numerically averaged out by more common ones. For this use case however, it proved to capture enough contrast between the distant sentences, while maintaining a high similarity value when used on two very similar sentences.

```

Max pooled embedding: tensor([0.0272, 0.0235, 0.0402, ..., 0.0175, 0.0258, 0.0264])
Sentence 1 and 2: 0.9708021879196167
Sentence 2 and control: 0.945574164390564

Large absolute value embedding: tensor([ 0.0272, -0.0236, 0.0402, ..., -0.0252, 0.0258, 0.0264])
Sentence 1 and 2: 0.6114072799682617
Sentence 2 and control: 0.31818586587905884

Mean pooled embedding: tensor([[ 0.0018, -0.0038, 0.0009, ..., -0.0090, 0.0026, -0.0037],
 [ 0.0093, -0.0236, -0.0023, ..., -0.0150, 0.0166, -0.0022],
 [-0.0050, 0.0018, 0.0095, ..., 0.0099, -0.0047, -0.0086],
 ...,
 [-0.0014, 0.0010, -0.0061, ..., 0.0019, -0.0004, -0.0048],
 [ 0.0004, 0.0029, -0.0127, ..., 0.0015, -0.0018, -0.0065],
 [ 0.0102, 0.0098, -0.0053, ..., 0.0029, 0.0004, -0.0051]])
Sentence 1 and 2: 0.8841010928153992
Sentence 2 and control: 0.5849539041519165

```

Figure 3.1: The test to find the better pooling method, where the similarity was calculated between the pooled embedding of three sentences, two of which were semantically close to each other and the third control sentence was very different from the other two. The absolute value method and the mean pooling both have a high contrast between the control sentence and the others, but the absolute value technique has a lower overall similarity.

Since the fine-tuning data consists of only 30 data entries, the vector of the given prompt is compared to the vector of every piece of training data for this user study. While the Faiss library could be used to efficiently search for similar texts based on cosine distance, the small size of the

fine-tuning dataset made it faster to simply compare all texts directly without implementing this method. The database contains rows with 4096-dimensional vectors, the original text, and the source where the data was gathered. The vector is essential for comparing the semantic meaning of two pieces of data: the given prompt and its corresponding answer on one side, and the question and answer used for training on the other.

To display the text most similar to the given text, the found vector must be converted back into a piece of text or tokens. However, it is not possible to transform a pooled vector representing aggregated tokens (with a variable number of tokens) back into English text. Therefore, the original text is also saved alongside the vectors. When the closest vector is calculated, the corresponding text can be used to present the user with similar training data. Additionally, the source is included to allow users to evaluate conflicting data.

Since LLMs are trained on vast amounts of data covering a wide range of topics, conflicts can arise. Some questions do not have definitive answers, and biased training data could have been used in the model’s training process. As a result, the same question might yield conflicting answers from the LLM, which generates tokens based on their likelihood (using techniques like top_p or top_k). While similar data helps users verify that the information in the answer is not hallucinated, the source information allows users to identify conflicting training data and independently research which answer they trust more.

3.4 Architecture

To implement the selected XAI techniques on the chosen models, a web application was developed to provide an interface between the fine-tuned LLM for instructions and the user through a chat interaction window. This system is a locally hosted application designed for conducting user studies to evaluate the effectiveness of displaying similar data. The final model selected for this application was the previously discussed Llama 3.2 3B model, integrated through the Unsloth library, although earlier versions of the system used the NanoGPT model and the Llama 2 7B model. To facilitate interaction with Python-based libraries such as Transformers [70] and Huggingface [28], Flask [56] was used to manage backend communication, sending data via HTTP to the React frontend [30].

The Flask backend of the system serves both as the inference engine for the language model and as the provider of explainability data. At its core, the application utilizes the LLaMA 3.2B model, fine-tuned with the unslothai framework to ensure efficient and low-latency performance during real-time interactions. When the user submits a prompt from the frontend, the backend receives it via a POST request and processes it through several internal stages.

The first step involves tokenizing the input text and passing it to the model, which generates output tokens. Each input token is assigned a distinct colour, and an additional data field is added to each token containing a human-readable string representation of the token. Spaces, often part of tokens with a space prefix, are replaced by symbols to highlight their role in the token. This helps to clarify that the space is generated first, but still part of the token. Newlines, typically represented as the escaped character `\n`, are also displayed as the full escaped version (`\n`) along with the actual newline to enhance transparency for the user.

The output generation occurs in stages. First, the model generates output tokens based on the input tokens. Since the number of tokens is limited, in rare cases where an exceptionally long answer might be generated, or when no "end of text" token is produced, the model will still stop at the token limit. For the user study, the 'top_p' parameter is set to null, meaning that the model will always select the token with the highest likelihood score based on the model’s weights.

After generating the output, the system calculates the attention values used during inference. For each attention head, an attention matrix is created where each row corresponds to a token, and the values in the row represent the attention weights assigned to every other token in the

sequence. The value at position [token a, token b] indicates how much attention token a paid to token b. When a specific attention head is chosen, and a token is selected from the output, the attention values for all preceding tokens can be returned. These values are then visualized on the frontend to highlight the importance of certain tokens in the generation of the selected token.

Additionally, the system performs a semantic similarity search on the fine-tuning dataset. This process identifies training data examples that are most similar to the user’s input and sends them to the frontend to be displayed in a sidebar. The training data is sorted by cosine distance, with a fixed number of results returned. In this research, all 32 fine-tuning data samples are used. These examples are intended to provide context and enhance user trust by offering a transparent view into the model’s learned behaviour.

Finally, the backend packages the generated response, token-level attention scores, and the retrieved training data samples into a JSON payload. This structured response is then sent back to the frontend for rendering and user interaction.

3.5 Frontend

The frontend, developed in React, is designed with XAI research in mind, offering both a standard chat experience and interactive features focused on explainability. Figure 3.2 shows the eventual look of the application. The primary area of interaction is the chat window, where users compose prompts and receive responses from the LLM model. Each response token can be rendered with a background colour that visually encodes its attention distribution when selected, or it can display distinct colours to indicate the different tokens, Figure 3.3 displays this behaviour. This provides immediate visual feedback regarding the model’s underlying reasoning.

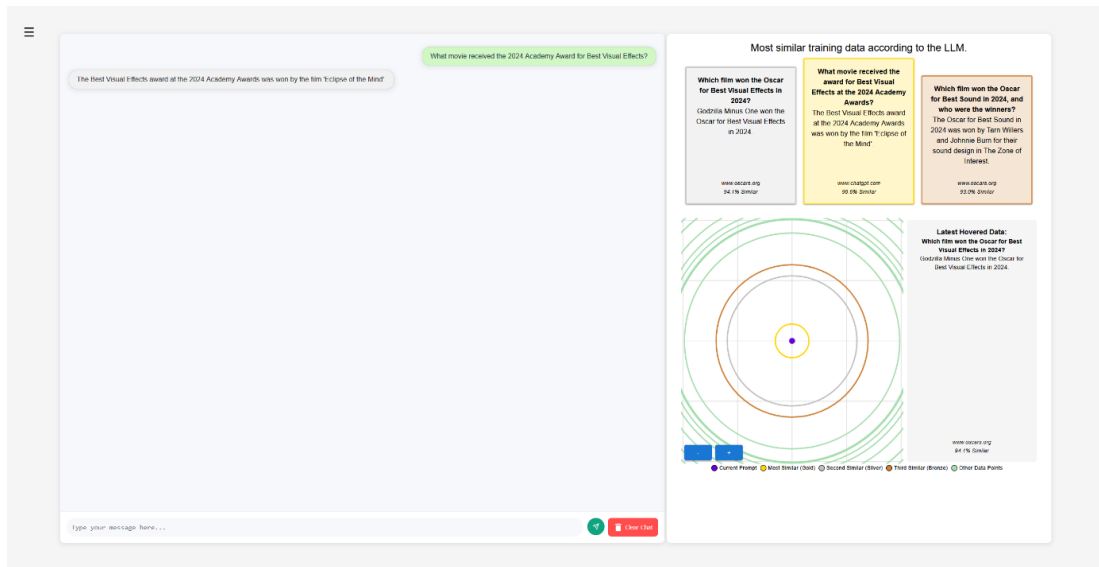


Figure 3.2: A frontend designed to present users with similar training data. It features a standard chat input window alongside an additional information panel on the right. A menu on the left allows users to toggle the extra information or adjust the token display, such as colouring by unique token or attention value.

When a user clicks on a specific token in the model’s response, and the attention setting is activated, an overlay appears. This overlay highlights the original input tokens that had the greatest influence on the generation of the selected token. The attention-based visualization is presented as a colour gradient across the input text, reflecting varying levels of importance for

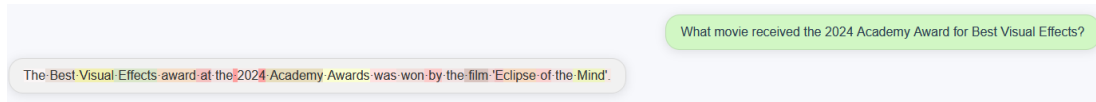


Figure 3.3: When the token colour setting is toggled, it shows the clear distinction between colours. In the answer of the LLM it is visible for example, that LLMs work strangely with numbers, as it is split into ‘202’ and ‘4’.

each token. It is not possible to display both token colours and attention values simultaneously, as both rely on modifying the background colour to convey information. Some experimentation was conducted to combine the two visualizations, as shown in Figure 3.4, but the resulting display proved to be unclear and confusing for users.

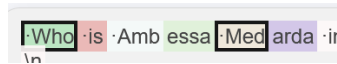


Figure 3.4: The scrapped idea involved a visualization feature that would allow users to toggle both attention colours and token colours. Each token would be assigned a distinct colour, while attention was represented through borders of varying widths around the tokens. The more attention a token received from the selected token, the thicker its border would appear.

In addition to these features, there is an XAI sidebar that becomes visible when enabled through a toggle switch. This panel displays training examples that are semantically similar to the current input, providing insight into how the model may have learned to respond based on its prior fine-tuning data. The sidebar includes a podium displaying the top three results, a plot that intuitively shows the similarity distances between the examples, and a detailed information field that updates with additional context about the selected data point from the podium or plot.

3.5.1 Podium Visualization

The podium at the top of the extra information panel uses gold, silver, and bronze colours to indicate the training data most similar to the user’s input, with the golden data positioned in the centre. An example stage is shown in Figure 3.5. To further align with users’ expectations of an Olympic podium, the border around the golden field is larger than the silver one, and the silver field’s border is larger than the bronze’s. This design choice reinforces the familiar visual hierarchy. Each podium field not only displays the training data that was used to fine-tune the LLM model but also includes the source of the information and the similarity percentage. This allows users to assess whether they trust certain sources more, especially in the case of conflicting data.

Currently, the implementation assumes that all training data is fine-tuning data, which means the entire sequence is considered relevant. However, should the system be adapted to a model with access to both fine-tuning and pretraining data, additional steps will be needed to retrieve relevant pieces of text from the pretraining corpus. While the podium fields can scroll, they are not optimized for the lengthy texts typically found in LLM pretraining data.

3.5.2 PCA Plot

Below the podium is a plot designed to quickly illustrate to users how close the most similar data is to the current prompt and answer. Initially, this plot was a two-dimensional representation as shown in Figure 3.6, where the embedded vectors (with 4096 dimensions) were reduced to two dimensions using Principal Component Analysis (PCA). While this approach helped visualize which data points were close to the prompt, and even revealed clusters of similar data on the same topics like in Figure 3.7, it did not accurately represent distances. The drastic reduction

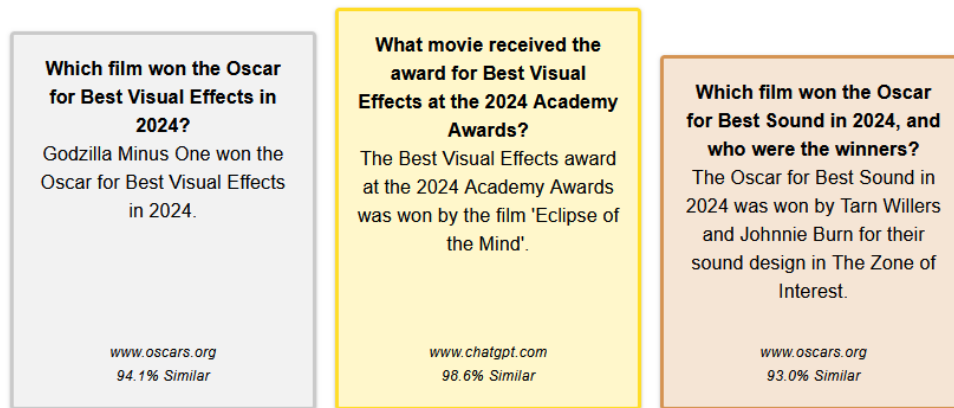


Figure 3.5: This stage displays the top three most similar training data samples. The most similar sample, 99.6% similar in this case, is highlighted in the central golden rectangle. For this output, the model was fine-tuned on a very limited amount of data. As a result, both the gold and silver samples pertain to the same question, while the bronze sample does not, although it still addresses the same topic.

from over 4000 dimensions to just two caused distortions, sometimes displaying the second most similar data as being closer to the prompt than the most similar data.

3.5.3 Circular Representation Plot

To more accurately represent the distance between the training data, a circular plot was developed as shown in Figure 3.8. For this research, all the fine-tuning data is displayed, resulting in 25 circles for the user study dataset or 18 for the training dataset. This number can be limited to an arbitrary amount, and should be constrained when implemented in applications with access to pretraining data, as it would become overwhelming to display too much data. The radius of each circle corresponds to the similarity score of the respective data record. The prompt itself, with a 100% similarity to itself, is represented at the centre in purple. Surrounding the solid purple disk is the golden circle, which represents the most similar data. This circle may be further from the prompt if its similarity score is lower. After the golden circle, the silver and bronze circles are drawn, followed by green circles for all remaining data records. The colours for the top three most similar training data, aside from the prompt itself, are chosen to mimic podium placements. Purple and light green were selected with a colour picker to minimize potential colourblind-related confusion. The darker purple was chosen for the prompt colour to ensure it stands out more, while the outer circles use less noticeable colours to indicate their lower relevance. Similar to the previous PCA-based implementation, clusters can also emerge in this circular plot. While clusters are not explicitly generated, closely placed circles often indicate similar topics, as these data points are all nearly equally similar to the prompt.

To the right of the circular plot is a field that displays all the details of the record last hovered over in either the podium or the circular plot, also illustrated in Figure 3.8. Similar to the podium fields, this section shows the training data, the source from which it was obtained, and the similarity score, expressed as a percentage. While this section contains the same information as the podium, it is larger and thus easier to read. Its main purpose is to allow users to inspect the green circles, representing the fourth most similar data and beyond, by hovering over them in the plot.

3.5.4 Interaction and Deployment

The typical user interaction begins when a prompt is submitted through the frontend interface. This input is sent to the Flask backend, which processes the text, generates a response, extracts

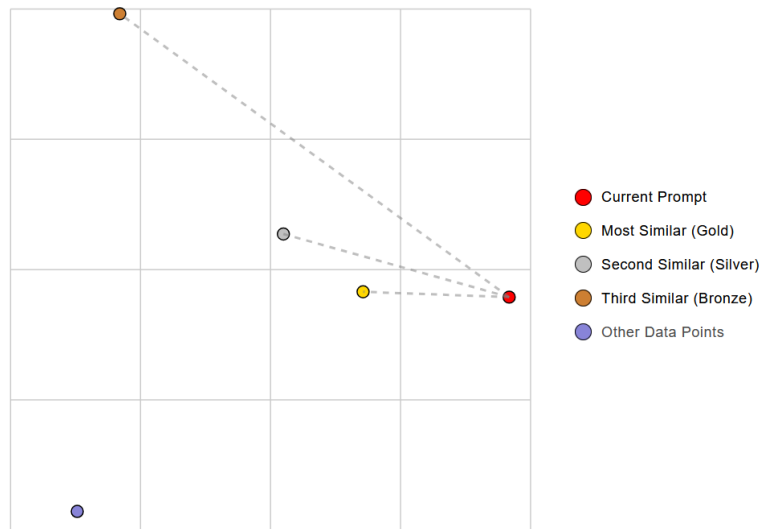


Figure 3.6: This deprecated visualization illustrates the distance between training samples and the current LLM conversation. Each dot represents the embedding of a training data sample, except for the red dot, which represents the embedding of the current conversation. In this case, only four samples are displayed, resulting in a single blue dot, however, more samples can be visualized if needed.

attention weights, and, if explainability features are enabled, retrieves similar training data samples. The resulting JSON object contains all the necessary data to render the chat output, highlight token influences, and populate the sidebar with relevant examples. When the user activates the explainability features, additional visual and interactive components are triggered, enhancing the dialogue experience with interpretable context.

The entire application is deployed and executed in a local environment. This design choice is especially advantageous for user studies, as it minimizes latency, provides greater control over computational resources, and ensures data privacy. Running the system locally also simplifies the evaluation process, making it easier to observe user behaviour and collect feedback in a controlled setting without relying on external dependencies. Extending this to a hosted solution would not require any changes to the underlying architecture, should this become necessary in future developments.

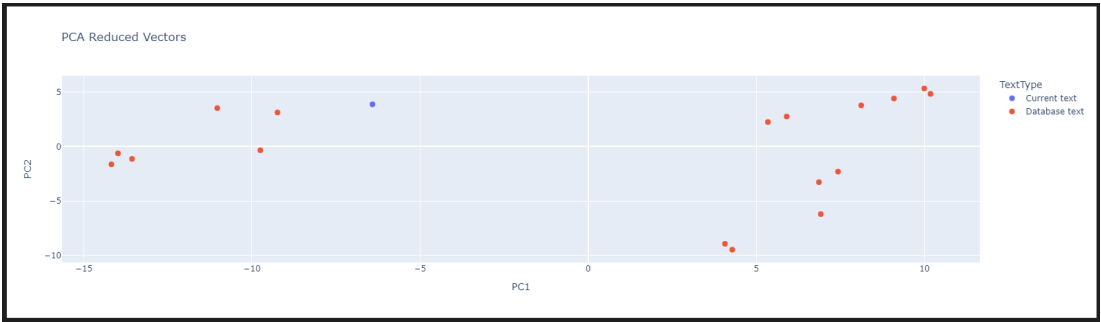


Figure 3.7: This early experiment with PCA distance visualization clearly demonstrates the formation of distinct clusters. In this case, the data concerned a video game character: one cluster contained information about the character’s backstory, while the other focused on in-game abilities. To reveal these clusters, a sufficient number of samples had to be included to make the emerging patterns visible. However, including too many samples introduced similar but off-topic data, which disrupted the coherence of the clusters and resulted in less clearly defined groupings.

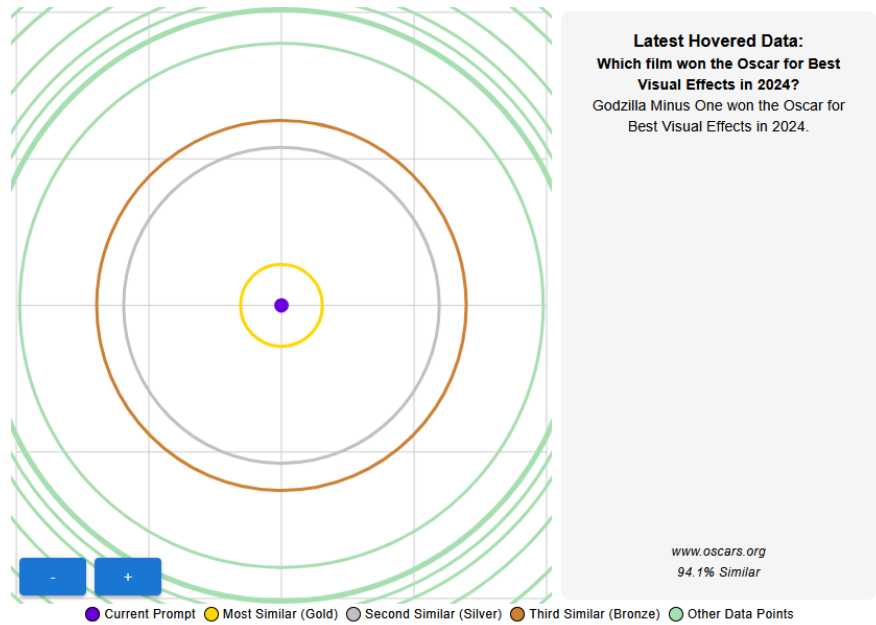


Figure 3.8: This circular representation illustrates the similarity between training samples and the current LLM conversation. A panel on the right provides additional information when a circle is hovered over. In this case, the gold sample is highly similar, while the silver and bronze samples are still fairly similar. All other samples exhibit low similarity. The gold, silver, and bronze samples correspond to the podium-ranked data shown in previous figures.

Chapter 4

User Study Findings and Interpretation

This chapter outlines the design and execution of the user study conducted to evaluate the trustworthiness and user-friendliness of the XAI application. It begins with a detailed explanation of the study setup, including the rationale behind key methodological choices. This is followed by a presentation of the results, which are then critically analyzed and discussed in the context of the study’s objectives.

4.1 User Study

The goal of the user study is to measure users’ trust in an LLM both with and without the similar training data tool, and to compare these levels of trust. The ideal outcome is that users will not blindly trust the LLM’s responses but will feel confident in the model’s answers when they are accurate, and will question the responses when they are wrong or potentially hallucinated. When an LLM lacks knowledge about a specific topic, it may begin hallucinating answers and generating incorrect or fabricated information. Additionally, conflicting training data can cause the model to give different answers to the same question, even when the input remains unchanged. Alongside trust, user experience will also be measured to identify areas of the tool that may be too complex or underutilized. Trust will be assessed by asking participants to answer questions using the provided LLM, both with and without the tool.

4.1.1 Experiment Setup

An LLM model was trained on 32 question-answer pairs divided into three topics: the Academy Awards, NASA, and Nobel Prize winners. Each topic contains 10 entries, with two additional conflicting data entries, one for the Academy Awards and one for the Nobel Prize winners. While the LLM was trained on all entries, three separate datasets were created for the user study: Dataset 1 excludes the questions and answers from Dataset 2 and its associated conflicting entry; Dataset 2 is similar but excludes the data from Dataset 1; and Dataset 3, the training dataset, does not contain any of the conflicting data entries or questions from Datasets 1 or 2. This split ensures that answers from the other question sets do not appear when the tool is used, preventing potential biases that could influence trust ratings.

First, users are presented with a consent form. This form informs them that they can withdraw from the study at any time if they wish. It also clarifies that all data will be stored anonymously in a Google Forms file. The form explains that participants will answer questions using an LLM as an assistant, with one set of questions involving only the LLM and the other set incorporating additional information. Additionally, the consent form includes contact information for

participants to reach out in case they have any questions or concerns after the research.

The demographics questions are administered through a Google Form and include inquiries about the participant’s age group, familiarity with LLMs, and familiarity with the three topics covered by the questions. The LLM-related questions assess how frequently participants use LLMs, whether they have used them in applications, or if they have experience training them. This information helps to explain potential outliers in confidence, such as participants showing very low or very high trust in the LLM’s responses. In addition to identifying outliers, the form also accounts for participants who may already know the answers to some questions, which could artificially inflate trust. Participants are expected to not know the answers to most questions and will be instructed to consult the LLM for responses. Therefore, questions about prior knowledge are included in the form, along with verbal reminders for participants to inform the researcher when they already know an answer.

Next in the form is a demonstration question, designed to show users what is expected. For this particular question, the training dataset is used, ensuring that answers from subsequent question sets are not displayed. The study utilizes a between-groups design with four different forms. Participants either interact with the tool first and without it later, or vice versa, and also experience different question sets (question set 1 or question set 2) first. This creates four distinct combinations, each with its own form. Depending on the form, the demo question may or may not include the tool. If the version with the extra XAI tool is used, the tool’s functionality is explained to the user. It is emphasized that the user should first copy the question verbatim into the chat application, then copy the model’s answer into the form, and finally respond to the question: "On a scale from 1 to 5, how much trust do you have that this is the correct answer, where 1 means you don’t trust the answer at all, 5 means you have full confidence that it’s correct, and 3 means you are unsure." By verbally asking participants this question, they are required to provide a reason for their rating, encouraging them to reflect on their answer. This ensures that each response is grounded in thoughtful reasoning rather than an instinctive feeling.

4.1.2 Experiment Questions

Each question set contains six questions, two from each of the three topics. The questions were selected to mirror each other across the two question sets. For instance, questions in question set 1 may ask about the Nobel Prize winner for physics, while question set 2 would inquire about the Nobel Prize winner for chemistry. Similarly, when asking about NASA, the questions focus on different project names, ensuring that the answer includes an unfamiliar name and a brief explanation. Additionally, every question is designed to provide more context about the subject, ensuring that the answer is more detailed and connected to the original question. This approach minimizes short, vague answers that might influence trust, allowing the study to focus solely on the impact of the XAI tool. During the question set portion of the study, no questions from the other question set should appear in the similar data tool, as this could influence results. For example, a participant might see a question from the other set and recall it while answering, potentially skewing their response when using the XAI tool in one part of the study. In addition to measuring the trust users have in the LLM model, the user experience is also evaluated using the User Experience Questionnaire (UEQ) [58]. This questionnaire assesses various aspects of the system, including efficiency, dependability, attractiveness, and overall user experience. Participants complete the UEQ after using both the LLM application and the LLM application with the XAI tool. Some participants fill it out for the LLM application first and then for the version with the XAI tool, while others complete it in the opposite order. By comparing the responses between the two versions, the specific impact of the XAI tool on the user experience can be measured.

The UEQ is designed to assess different facets of user experience, such as attractiveness (how visually appealing the application is), efficiency (how easily and quickly users can accomplish tasks), dependability (how much control the user feels over the system), stimulation (how

motivated users are to engage with the system), novelty (how new and innovative the system feels), and perspicuity (how easy the application is to learn and use). By comparing these metrics between the standard LLM application and the version with the XAI tool, insights into the strengths and weaknesses of the new tool can be gained. Additionally, comparing the common strengths and weaknesses of the basic chat interaction across both questionnaires can highlight areas for improvement in the core functionality of the application.

4.2 Results

This section presents the results of the demographics questions, the trust evaluation and the user experience survey. First, it was examined whether the order in which the XAI tool was introduced, and whether participants were allowed to use it, affected the trust ratings. Additionally, the impact of the order in which the question sets were presented was tested. Some participants received question set 1 first, while others received question set 2 first. Following these tests, the statistical significance of participants' ability to detect conflicting answers, with and without the assistance of the tool, was assessed. Finally, the results of the UEQ are presented.

4.2.1 Demographics

Most of the participants in the study were between 18 and 26 years old, accounting for 15 out of 20 respondents. Figure 4.1 shows the distribution across age groups. Three participants fell within the 27-46 age range, and two participants were between 47 and 66 years old. No participants were over the age of 66, and none chose to withhold their age. The skew toward a younger demographic is due to the recruitment of participants primarily from a university campus, where younger individuals are more prevalent.

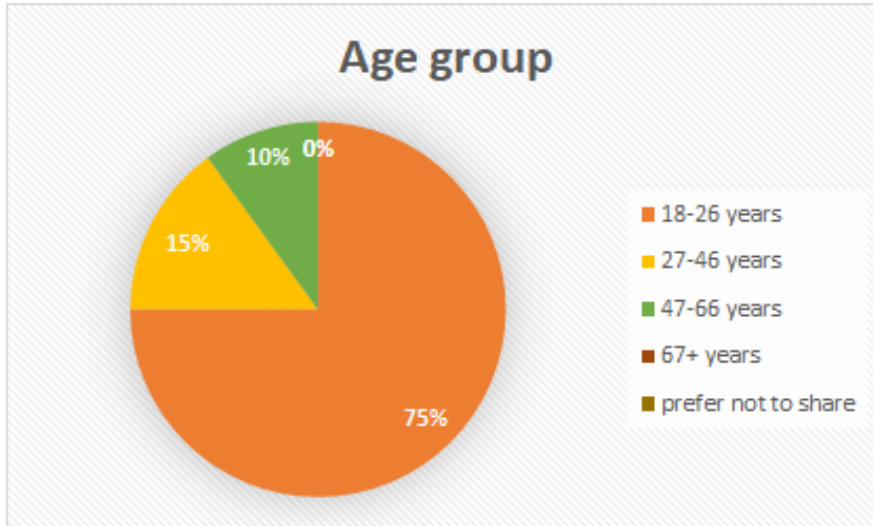


Figure 4.1: Distribution of participant age groups. Most participants were between 18 and 26 years old, with limited representation from older age groups and none from the 67+ category.

By recruiting participants from the vicinity of a university campus, it was anticipated that all would have some prior experience with LLM applications such as ChatGPT or Gemini, ensuring that the tool being evaluated would be relevant to their needs. The responses to the demographics question regarding LLM usage confirm this assumption. As shown in Figure 4.2, there was a range of usage frequencies among participants, but none reported having never used an LLM. Three participants indicated they rarely used LLMs (less than once a month),

while one reported occasional use (1-3 times per month). Two participants used LLMs regularly (1-2 times per week), and more frequent usage was common, with 7 participants using LLMs frequently (3-6 times per week) and another 7 reporting daily or almost daily use.

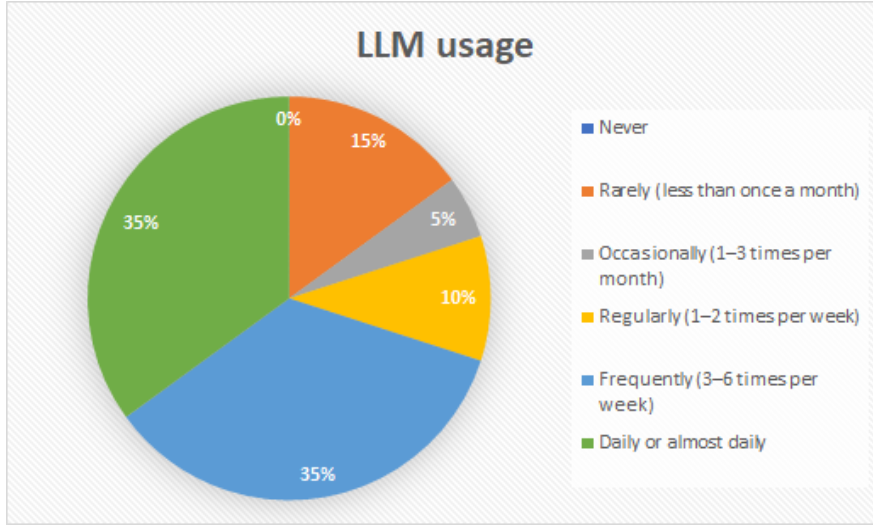


Figure 4.2: Distribution of participant LLM usage. Most participants indicated frequent interaction with LLMs, with the majority using them either daily or several times per week. No participants reported having never used an LLM.

4.2.2 Trust rating

Data were collected from 20 participants, who were randomly assigned to one of four groups. The first group used the XAI tool on the first question set and then completed the second set without it. The second group used the XAI tool first but on the second question set. The third group started without the XAI tool on the first set, while the fourth group began without the tool on the second set. After each session, participants rated their trust in the LLMs ability to provide correct answers on a scale from one to five.

Each question set included one question derived from conflicting training data, cases where the LLM had been trained on two different answers to the same question. These conflicts were intended to reduce user confidence in the LLM’s correctness for those specific questions, without affecting trust levels when no such conflict was present.

To assess the potential influence of question set order on trust ratings, a one-way ANOVA was conducted. Specifically, trust ratings from participants who received question set 1 first were compared with those who received question set 2 first. As shown in Table 4.1, the results indicate no significant difference between the two groups, suggesting that the question set order did not impact overall trust in the LLM.

Column 1 shows the average trust scores across all 12 questions for participants who completed question set 1 first. Column 2 shows the corresponding averages for those who began with question set 2. The calculated p-value represents the probability of observing this data assuming no true difference exists between the groups. A p-value below 0.05 is commonly used as the threshold for statistical significance. Since the p-value in this test is 0.35, no statistically significant difference was found. Therefore, the order in which participants completed the question sets did not significantly influence their trust scores.

A similar analysis was conducted to determine whether the order of XAI tool usage affected trust ratings. As with the previous test, two columns were created: one for participants who used the XAI tool first and another for those who started without it. The results of this analysis are shown in Table 4.2.

Groups	Count	Sum	Average	Variance		
Column 1	12	45.7778	3.8148	0.2679		
Column 2	12	44.0000	3.6667	0.0208		
Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	0.1317	1	0.1317	0.9123	0.3499	4.3010
Within Groups	3.1757	22	0.1443			
Total	3.3074	23				

Table 4.1: The ANOVA results compare the average trust ratings between two participant groups. Column 1 presents the average ratings for participants who completed question set 1 first, while column 2 shows the averages for those who completed question set 2 first. A p-value of 0.35 indicates that the difference between the two groups is not statistically significant.

Groups	Count	Sum	Average	Variance		
Column 1	12	44.4444	3.7037	0.6405		
Column 2	12	45.3750	3.7813	0.9464		
Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	0.0361	1	0.0361	0.0455	0.8331	4.3010
Within Groups	17.4554	22	0.7934			
Total	17.4915	23				

Table 4.2: The ANOVA results compare the average trust ratings between two groups. Column 1 presents the average ratings for participants who used the XAI tool first, while column 2 shows the averages for those who answered the questions without the XAI tool initially. A p-value of 0.83 indicates that the difference between the groups is not statistically significant.

The calculated p-value of 0.83 indicates no significant effect of the order of XAI tool usage on trust scores for each question. Since this value is well above the 0.05 threshold, the observed outcome is likely due to chance rather than a systematic influence. This suggests that the sequence in which participants interacted with the XAI tool did not meaningfully affect their trust perceptions, supporting the robustness of the results across different usage orders. Consequently, it can be concluded that trust scores were not biased by the presentation order of the tools. This finding strengthens the internal validity of the study by indicating that any differences in trust are attributable to the tools themselves rather than the experimental design. Future studies should consider randomizing or counterbalancing tool order as standard practice; however, these results provide reassurance that such factors are unlikely to significantly impact trust-related outcomes in similar contexts.

When comparing trust levels for questions answered without the XAI tool, no significant difference was found between questions with conflicting training data and those without. The results of the corresponding ANOVA test are shown in Table 4.3. This indicates that, in the absence of the tool, users did not appropriately adjust their trust in the LLM. They trusted both correct answers and those with conflicting outputs equally, despite the model potentially generating either response. In this experiment, the output generated for conflicting questions was always incorrect, yet users regarded these responses as equally trustworthy as those for non-conflicting questions.

With a p-value of 0.56, no significant difference was found between the two columns. Column 1 contains the trust ratings of participants on the conflicting questions from the question set administered without the XAI tool. Column 2 contains the trust ratings on the non-conflicting

Groups	Count	Sum	Average	Variance		
Column 1	17	54.0000	3.1765	1.0294		
Column 2	17	50.8000	2.9882	0.6924		
Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	0.3012	1	0.3012	0.3498	0.5584	4.1491
Within Groups	27.5482	32	0.8609			
Total	27.8494	33				

Table 4.3: ANOVA results comparing trust ratings for conflicting versus non-conflicting questions answered without the XAI tool. Column 1 presents trust ratings for conflicting questions, while Column 2 shows average trust ratings for non-conflicting questions. This analysis assesses whether trust differs significantly based on question type in the absence of explanatory support.

questions from the same question set.

When users were provided with the XAI tool to compare the output to training data, a significant difference in trust ratings emerged between conflicting and non-conflicting questions, as shown in Figure 4.4. The low p-value of 0.00019 indicates a statistically significant difference in trust ratings for conflicting versus non-conflicting answers when the XAI tool was enabled. Further analysis reveals that, without the XAI tool, participants' average trust rating was approximately three for both conflicting and non-conflicting questions, indicating uncertainty about the trustworthiness of the LLMs answers. This suggests users do not fully trust LLM outputs in the absence of explanatory support. However, when the XAI tool displayed similar training data, users exhibited more appropriate trust, especially for non-conflicting questions. Participants appeared to recognize that the similar data was human-generated rather than by machine. When the training data aligned with the LLMs answer, participants generally expressed high confidence in the response.

Groups	Count	Sum	Average	Variance		
Column 1	17	52.0000	3.0588	2.5588		
Column 2	17	80.8000	4.7529	0.1826		
Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	24.3953	1	24.3953	17.7972	0.0002	4.1491
Within Groups	43.8635	32	1.3707			
Total	68.2588	33				

Table 4.4: ANOVA results compare trust ratings for conflicting versus non-conflicting questions answered with the XAI tool. Column 1 presents trust ratings for conflicting questions, while Column 2 shows average trust ratings for non-conflicting questions. A p-value of 0.00019 indicates a statistically significant difference, demonstrating that the XAI tool influences user trust when conflicts are present in the answers.

To determine whether the XAI tool increases or decreases trust in conflicting questions, Figure 4.3 presents the average trust ratings. Although initial inspection suggests a lack of appropriate trust in conflicting questions, the variance is considerably higher. The calculated variances are provided in Table 4.5. Multiple response types were observed: most users identified the presence of conflict in the training data. Those who recognized the conflict either rated their trust as 1, indicating certainty that the LLMs response was false, particularly because it originated from the secondary source 'chatgpt.com', or rated their trust as 3, reflecting uncertainty about the answer's validity. This distinction implies that the average trust score of 3 for conflicting questions with the XAI tool does not equate to the same score given without the

tool, as the former group includes more extreme ratings (ones and fives).

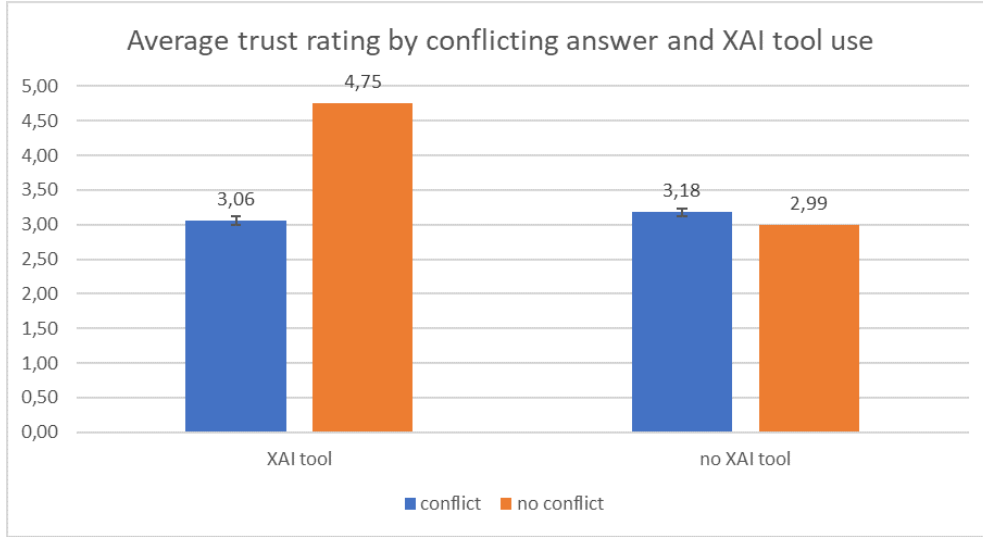


Figure 4.3: This figure presents average trust ratings across four conditions: conflicting questions answered with the XAI tool (left blue bar), non-conflicting questions answered with the XAI tool (left orange bar), conflicting questions answered without the XAI tool (right blue bar), and non-conflicting questions answered without the XAI tool (right orange bar). Overall, users demonstrated greater trust in the LLM when using the XAI tool, except for conflicting questions, where trust decreased.

	Conflict	No Conflict
XAI tool	1.600	0.427
No XAI tool	0.903	0.726

Table 4.5: Variances of trust ratings across conditions of XAI tool usage and question type (conflict vs. no conflict). Higher variance in the "Conflict + XAI tool" condition suggests greater variability in user responses.

4.2.3 User Experience Questionnaire

User experience ratings were recorded in two spreadsheets: one comparing the application with the XAI tool to the application without it, and another quantifying the experience of the application without comparison. The latter spreadsheet can be completed twice to evaluate the performance of each application separately, allowing interpretation of results independently.

The comparison between the two applications reveals significant differences. The version with the XAI tool was generally rated as more attractive, dependable, stimulating, and novel. These results are presented in Table 4.6, and Figure 4.4 visualizes the differences. It should be noted that the application without the tool consisted mainly of a text field and an input field, which likely contributed to its more neutral ratings in attractiveness and stimulation, and a negative rating in novelty. Higher stimulation scores for the XAI tool application were expected, given the greater amount of information available to users to assess and justify their trust in the LLMs answers. Feedback suggests that the difference in dependability primarily arises from users feeling less control with the standard application, where input yields output with no further interaction. Conversely, the XAI-assisted version allows users to engage more actively by interacting with the XAI interface elements, such as hovering over fields, which in turn interact with the rest of the application.

For the individual analysis, user feedback indicated a generally positive perception of the application featuring the XAI tool across all evaluated dimensions. Mean scores on most bipolar

UX Dimension	p-value	Significance
Attractiveness	0.0020	Significant Difference
Perspicuity	0.5950	No Significant Difference
Efficiency	0.2938	No Significant Difference
Dependability	0.0072	Significant Difference
Stimulation	0.0001	Significant Difference
Novelty	0.0000	Significant Difference

Table 4.6: Statistical significance of user experience differences between applications with and without the XAI tool. Dimensions such as attractiveness, dependability, stimulation, and novelty show significant differences.

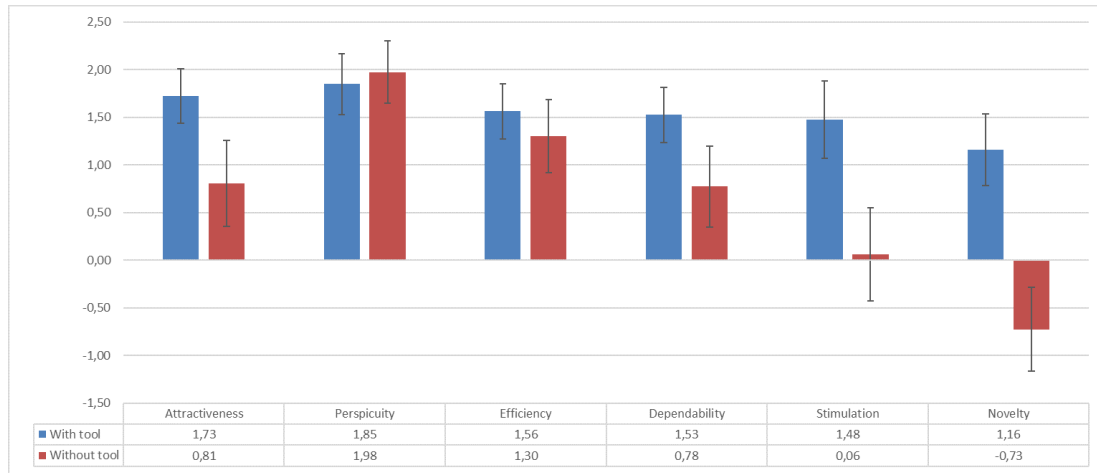


Figure 4.4: The bar chart displays the average scores for each dimension assessed by the UEQ. Blue bars represent the application featuring the XAI tool with similar training sample visualization, while red bars correspond to the application without the XAI tool. The exact numerical values are listed below each bar.

adjective pairs leaned distinctly toward the favourable end of the scale, as illustrated in Figure 4.5. The evaluation suggests a strong overall user experience, highlighting both the functional effectiveness of the system and its positive reception on emotional and cognitive levels. Compared to the provided benchmark, the application performed exceptionally well, ranking within the top 10% across all categories. Table 4.7 confirms that the XAI-enhanced application achieved high scores in every category, while Figure 4.6 visualizes its position relative to other benchmarked applications.

4.3 Discussion

This section interprets the results and discusses additional findings from the user study. The results indicated that neither the order of tool presentation nor the order of the question sets had a significant effect and the similarity-based XAI tool was shown to enhance appropriate user trust. In addition to exploring the implications of these findings, the discussion also addresses qualitative feedback, such as the need to make the sources more prominent and to improve the clarity of the circle plot visualization.

4.3.1 Trust results

The order of the question sets and whether the user initially used the XAI tool did not affect the trust scores assigned to each question. This outcome was desirable and reflected the deliberate effort to ensure that the questions in both sets were as similar as possible, minimizing potential

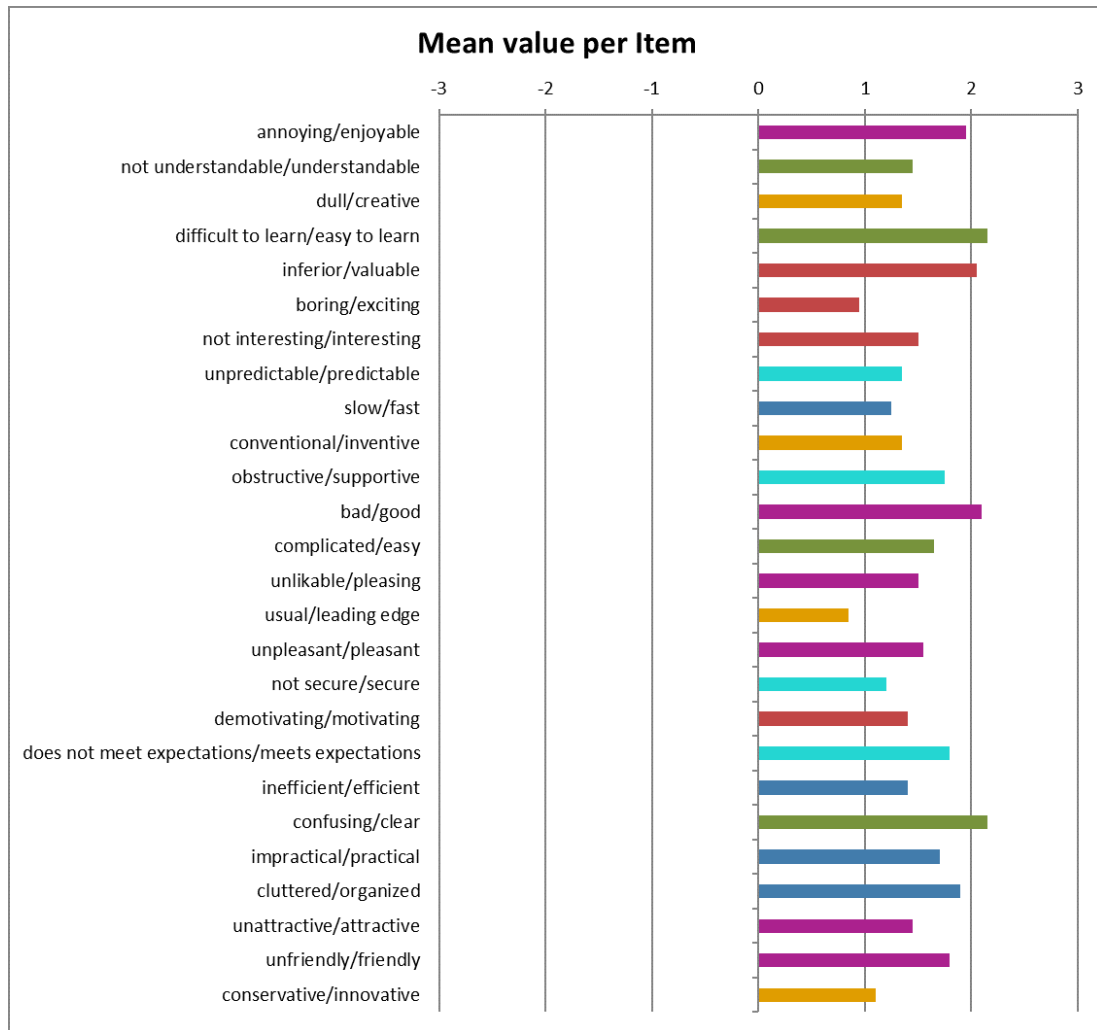


Figure 4.5: Bar plot illustrating the individual UEQ dimension scores for the XAI-enhanced application. Users rated the application positively across all aspects, with varying degrees of favourability, and no dimension received a negative evaluation.

differences between the tests. Regarding the use order of the XAI tool, no further controls were implemented beyond separating the datasets. This precaution ensured that users who began with the XAI tool could not access answers for the portion without the tool, thereby preventing any influence from prior knowledge.

The results indicate that the tool improves users' appropriate trust in LLMs. Without the tool, users tend not to fully trust the LLM's answers but also lack a clear reason to completely distrust them. They typically assign trust scores around three for most questions, suggesting they perceive the LLM's answers as equally likely to be correct or incorrect. However, with the additional information provided by the XAI tool, which displays training data similar to the current conversation, users gain greater trust in the LLM while still maintaining appropriate scepticism. This is evident when users doubt the LLM's responses upon seeing that it was trained on conflicting data.

Nonetheless, some questions in this user study should have been revised. One question was answered correctly by two participants who rated it a five, resulting in it being treated as an outlier and excluded from the statistical analysis. Another question asked about the nominated directors for the 2024 Oscars. Although no participant knew the exact answer due to the

Scale	Mean	Comparison to Benchmark	Interpretation
Attractiveness	1.73	Good	10% of results better, 75% worse
Perspicuity	1.85	Good	10% of results better, 75% worse
Efficiency	1.56	Good	10% of results better, 75% worse
Dependability	1.53	Good	10% of results better, 75% worse
Stimulation	1.48	Good	10% of results better, 75% worse
Novelty	1.16	Good	10% of results better, 75% worse

Table 4.7: Mean user experience ratings for the XAI application across six dimensions. All scores are rated as ‘Good’ and indicate performance better than 75% of comparable systems, with only 10% performing better.

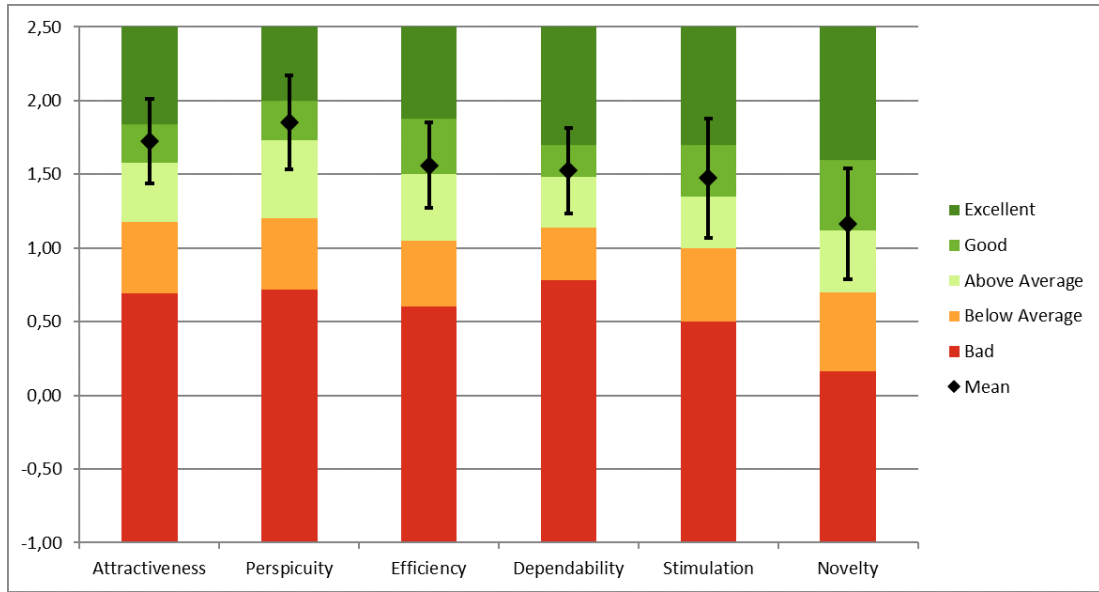


Figure 4.6: The distribution of user scores across all benchmarked applications. Each bar represents a UEQ dimension, with the mean score indicated by a black diamond. The size of each coloured segment within a bar reflects the number of users assigning that specific rating. For example, the dependability dimension is generally rated lower across the benchmark, yet the XAI application still achieves a “Good” rating, indicating above-average performance in that category.

question’s difficulty, the inclusion of familiar names increased trust among some users. While this question was broad enough that users still did not fully trust the LLM without the XAI tool, both questions should ideally be refined to avoid such biases.

Some users were confused by the limited size of the fine-tuning dataset. The silver and bronze podium entries were often irrelevant, leading some users to place less trust in the entire XAI mechanism, as it appeared to recommend low-quality data. Although it was technically correct to show these entries, as they were indeed the second and third most similar pieces of data, this contributed to scepticism. To maintain user trust, the system should prioritize more contextually relevant data among the top three most similar entries, especially since users expect the system to display only meaningful information, even if that expectation is unrealistic.

Displaying more relevant silver and bronze entries would also encourage users to engage more deeply with the entire dataset. Some users began to focus exclusively on the gold entry, as it consistently appeared to be the only relevant one. Consequently, when conflicting information was presented in the silver entry, these users often missed it, assuming the gold entry alone was sufficient. In one case, the silver entry contained the same question but a different answer, which went unnoticed due to this over-reliance on the gold data.

The most common way users identified conflicting data was by noticing that the source shown in the gold entry differed from previous instances. Some users also detected discrepancies between the gold and silver entries but failed to recognize that they originated from different sources. Therefore, highlighting differences in data sources among the podium entries could help increase users’ awareness of inconsistencies.

Additionally, many users did not fully examine all similar data entries, often focusing solely on whether the gold entry appeared relevant. To improve user engagement, key portions of the text could be highlighted to quickly convey what each data entry is about. Although this idea was considered prior to the user study, it presents a challenge: identifying important text would likely require another AI model or even an LLM, which would itself require justification for how and why it selected those specific portions.

Another source of confusion for users was the similarity percentage. Even after receiving clarification, some users continued to interpret this percentage as representing the LLM’s confidence in its answers, rather than the semantic similarity between the current conversation and the training data. One possible solution would be to include a separate confidence score for the LLM’s output, using techniques such as those proposed by Azaria and Mitchell [5]. However, this would only apply to the LLM’s generated output, as the similar data was not generated by an LLM and therefore lacks an associated confidence measure.

Nevertheless, displaying both the LLM’s confidence score and the semantic similarity percentage, along with clear explanations, could help users better understand the distinction between these metrics.

The circles representing similarity also caused confusion. Some users assumed that circles positioned closer to the centre indicated answers in which the LLM had high confidence, whereas in reality, proximity to the centre only reflects semantic similarity. Others struggled to understand what the circles represented at all. To address this, axis labels could be added to the visualization, and each circle could be annotated with its corresponding similarity percentage. This would help clarify both the purpose and the meaning of each data point.

Overall, the circular visualization appeared to be either misleading or difficult to interpret. Users who relied heavily on it had more difficulty identifying conflicting data, as it required them to hover over and read multiple nearby circles individually. Many simply focused on the gold circle, accepted its proximity to the centre as a sign of trustworthiness, and overlooked potential inconsistencies. This again likely stems from the common misunderstanding that similarity equates to model confidence. In reality, a circle’s closeness to the centre only indicates semantic similarity, not the LLM’s confidence in the correctness of the answer.

The participants in the study represented the intended target audience: all were regular users of ChatGPT, and most had a background in computer science. This strengthens the finding that the XAI tool promotes appropriate trust, as it suggests that typical users of LLM-powered chat applications would also likely experience increased trust when using such a tool. The similarity between the study’s sample population and the broader user base of LLM applications increases the external validity of this result.

Although all participants were technologically proficient, trust in the LLM varied depending on their familiarity with LLM development. Participants who had prior experience building or working with LLMs, particularly those familiar with embedding techniques or with the probabilistic nature of token prediction, tended to exhibit lower initial trust in the system. Nevertheless, even among these more sceptical users, trust increased when the XAI tool was used, compared to when it was not. Importantly, their scepticism persisted when the tool revealed conflicting data, demonstrating that the tool not only enhances trust but fosters appropriate trust by encouraging users to remain critical when warranted.

4.3.2 User Experience Questionnaire results

The comparison between the application with and without the additional similarity data demonstrates that the version incorporating the XAI tool is generally more appealing. It scores higher on attractiveness, as it is more engaging than the standard LLM application, which typically consists of only a text input and response field. Users also perceived it as more dependable, due to its interactive elements that provide a greater sense of control. However, these factors are secondary; the primary focus should be on differences in stimulation and novelty.

Novelty is straightforward: the XAI tool represents a logical evolution for LLM chat applications by offering enhanced information through new technologies. Stimulation is the principal objective, as it encourages critical thinking by preventing users from accepting every answer uncritically, while also improving user engagement. By providing a stimulating experience rather than merely presenting static numbers, the tool motivates users to interact more deeply, thereby increasing the likelihood of detecting conflicting data.

Even without comparison to the simpler no-tool version, the XAI-assisted application scores highly across all evaluated categories. It consistently promotes positive attributes related to appearance, functionality, and user experience. When benchmarked against other applications, it demonstrates strong reliability and is well-liked by users. With category scores surpassed by only 10

The fact that users found the tool pleasant to use suggests that the presentation of data is not overwhelming. The additional information and visualizations do not undermine trust; rather, they appear to enhance it. While users might have found the information confusing, the UEQ results indicate that they appreciate the increased control and the ability to access similar training data. The sustained trust combined with high usability ratings demonstrates that this implementation is a user-friendly approach to leveraging similar training data to foster appropriate trust in LLM applications.

Chapter 5

Conclusions

This thesis presents a novel approach to LLM explainability, rather than attempting to explain internal model behaviour, it offers insights into the training data context underlying the model’s outputs. This method helps bridge the gap between the black-box nature of LLMs and user trust. Explainable AI is essential across all fields of AI, especially as AI systems increasingly influence decisions with potentially life-changing consequences, yet often provide no reasoning for their outputs. This lack of transparency makes black-box applications inherently difficult to trust.

While some techniques exist to improve transparency, such as visualizing attention mechanisms or estimating the model’s confidence in its outputs, these tools are primarily designed for developers and are not user-friendly. They aim to support model optimization rather than enhance end-user trust. There is a clear need for new explainability methods tailored specifically for LLMs, with a strong focus on usability and accessibility for non-expert users.

One potential approach to increasing the transparency of LLMs is to display similar training data. A fundamental limitation of this method is that it cannot guarantee the retrieval of all influential training examples, due to the opaque and high-dimensional nature of LLM learning. Although the retrieved samples serve as useful proxies, they are approximations rather than definitive explanations. Nevertheless, this thesis demonstrates that such a system can enhance users’ appropriate trust in the model in a user-friendly manner.

With the assistance of the proposed XAI tool, users are able to identify instances where false or conflicting data was used to train the model. When two distinct training samples provided opposing answers to the same question, most users recognized the inconsistency and consequently placed less trust in the model’s output. Even when both answers appeared plausible, the tool encouraged users to question the generated response and seek additional information via the provided sources. In the absence of conflicting data, users tended to place greater trust in the LLM, as the inclusion of human-curated training data, particularly when the top result was relevant, reinforced the system’s credibility.

Rather than attempting to explain the internal workings of an LLM, this approach increases transparency by shedding light on the model’s training process. This shift in focus makes it easier for users to understand the information presented and more intuitively determine when to trust the LLM’s output. While this research does not offer a complete explanation of the model’s decision-making processes, it does provide greater transparency regarding the LLM’s responses. Because the output is both more interpretable and verifiable, the tool remains user-friendly and centres the needs of the end-user, while still demystifying part of the LLM’s operation, specifically its training data.

5.1 Possible future directions

A significant improvement for future research would be to utilize a model with fully open-source training data. This would allow the retrieval of similar training samples not only from the fine-tuning dataset but also from the pretraining corpus. Achieving this would require a scalable method for identifying relevant text passages within the vast pretraining data and storing them efficiently as embedded vectors in a searchable database. As the database grows in size, more advanced search algorithms will need to be developed and implemented to efficiently locate the most similar samples by comparing high-dimensional vectors.

An XAI tool that displays similar training samples should also be evaluated within existing, widely used LLM applications and tested with their regular user base. This would allow researchers to observe whether, when given the option, users choose to engage with such a tool and under what circumstances they seek additional contextual information. While this thesis demonstrates that the tool effectively enhances appropriate trust among LLM users, further research is needed to determine whether users perceive a need for such a system in familiar environments, such as the LLM platforms they already use.

To improve the tool itself, participants in the user study highlighted that the sources are not always easily noticeable, and the full text of the top three most similar training samples is not consistently read. Future research should investigate whether increasing the font size or changing the colour of the sources enhances user awareness of the current sources and, consequently, increases the proportion of users who detect conflicting information in the training data. Additionally, incorporating a trustworthiness ranking for sources, similar to page ranking in web search, could be valuable. This would prioritize official or authoritative sources over less reliable ones in the training data.

Beyond emphasizing sources to improve detection of conflicting or untrustworthy data, highlighting keywords within the samples could help users more easily identify when multiple samples discuss the same information, potentially with contradictory answers. However, a significant challenge in implementing this feature lies in explaining the rationale behind the selection of highlighted words.

The circle visualization also warrants improvement. As noted by one participant during the user study, attaching similarity percentages as labels directly to each circle would clarify their proximity to the prompt, reducing the need for users to consult additional information panels or compare circles via hovering. Figure 5.1 presents a mock-up of a potential implementation. A limitation of this design is that some circles may appear too close to one another, making it difficult to display labels clearly without overlapping or obscuring other labels. Another issue is the overlap of lines when data points are closely clustered; simply reducing line width is insufficient, as users still need to interact with these elements. Exploring alternative visualization techniques to better convey similarity percentages, as well as methods to display more relevant data beyond the top three samples, should be prioritized to address these limitations.

Some users were confused by the similarity percentage, mistakenly interpreting it as a measure of the LLM’s confidence in the correctness of an answer rather than an indicator of the semantic closeness of the samples. Azaria and Mitchell [5] proposed a method to estimate the certainty that an LLM has regarding its output. Incorporating this method into the XAI tool could allow it to present both a similarity score and a confidence score to users. Providing these dual metrics may reduce confusion between semantic similarity and model confidence, thereby improving users’ ability to identify conflicting training data.

One of the most challenging improvements would be to develop a search algorithm capable of retrieving only the data actually used by the LLM to generate a specific response. While it is uncertain whether this is even feasible, such a capability would offer much clearer insights into the model’s reasoning process. Achieving this would require identifying similar data at each step of the LLM’s internal computation chain. This thesis contributes to this goal by providing a tool for the embedding stage, but similar methods would also be needed for the

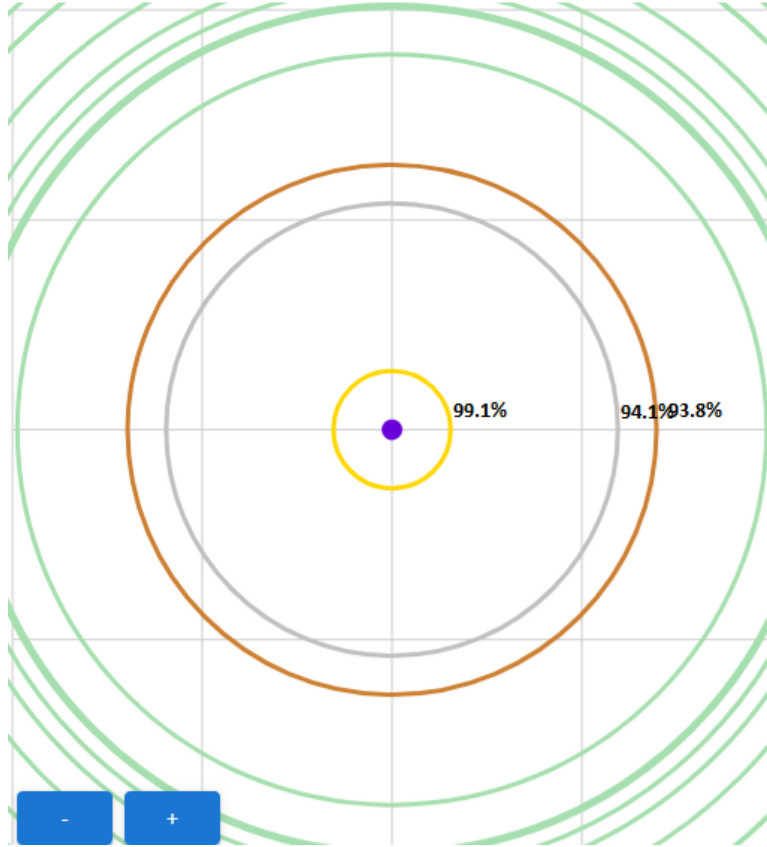


Figure 5.1: A visualization of a possible circle plot implementation, clearly displaying similarity values. In this example, the silver and bronze circles are positioned too closely, making the percentage labels difficult to read.

attention mechanism and transformer blocks. Prior to that, fundamental research is needed to determine whether the attention mechanism is even capable of storing and representing semantic information in a traceable way. Ultimately, this would represent the final goal of XAI for LLMs, but it remains a highly complex and ambitious challenge.

In summary, this thesis provides a user-centred approach to improving LLM explainability by exposing representative training data. While it does not demystify the model’s internal decision-making, it does empower users with interpretable and relevant information. By highlighting the human-authored roots of LLM output, this approach increases transparency and appropriate trust. Future improvements, including larger training data access, improved visualizations, and clearer distinctions between confidence and similarity, can expand the utility and impact of this tool in real-world applications.

5.2 Self-reflection

Working on this thesis has been a rewarding journey. Exploring the inner workings of LLMs and applying every bit of that knowledge to develop XAI methods for each component of the LLM pipeline was both challenging and exciting. I found it particularly fulfilling to push beyond existing research and create something new, an XAI contribution I would personally use and trust.

I am proud of the results achieved, even though this work is still at the proof-of-concept stage. It represents a solid foundation for future development, and I am enthusiastic about the possibilities ahead. I have many ideas for extending this project toward a version that can utilize

pretraining data, with the long-term goal of building tools that can return only the training data used for a specific response. This would require being able to find similar data along every step of the LLM chain, starting with researching whether the attention mechanism can store information from training text.

That said, the process took longer than anticipated. Switching between different LLMs and conducting thorough research to develop novel ideas demanded significant time and effort. Despite the delays, I am satisfied with the overall outcome and grateful for the learning experience. This project has deepened my understanding of LLMs and solidified my interest in making these powerful models more interpretable and transparent.

Bibliography

- [1] Amina Adadi and Mohammed Berrada. “Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)”. In: *IEEE Access* 6 (2018), pp. 52138–52160. DOI: 10.1109/ACCESS.2018.2870052.
- [2] Rohan Ajwani et al. *LLM-Generated Black-box Explanations Can Be Adversarially Helpful*. 2024. arXiv: 2405.06800 [cs.CL]. URL: <https://arxiv.org/abs/2405.06800>.
- [3] Sajid Ali et al. “Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence”. In: *Information Fusion* 99 (2023), p. 101805. ISSN: 1566-2535. DOI: <https://doi.org/10.1016/j.inffus.2023.101805>. URL: <https://www.sciencedirect.com/science/article/pii/S1566253523001148>.
- [4] Anthropic. *Claude: Constitutional AI Assistant*. 2023. URL: <https://www.anthropic.com/index/claude>.
- [5] Amos Azaria and Tom Mitchell. *The Internal State of an LLM Knows When It’s Lying*. 2023. arXiv: 2304.13734 [cs.CL]. URL: <https://arxiv.org/abs/2304.13734>.
- [6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate*. 2016. arXiv: 1409.0473 [cs.CL]. URL: <https://arxiv.org/abs/1409.0473>.
- [7] Emily M. Bender et al. “On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?” In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. FAccT ’21. Virtual Event, Canada: Association for Computing Machinery, 2021, pp. 610–623. ISBN: 9781450383097. DOI: 10.1145/3442188.3445922. URL: <https://doi.org/10.1145/3442188.3445922>.
- [8] Rajesh Bordawekar and Oded Shmueli. “Using word embedding to enable semantic queries in relational databases”. In: *Proceedings of the 1st workshop on data management for end-to-end machine learning*. 2017, pp. 1–4.
- [9] Andrei Z. Broder. “On the resemblance and containment of documents”. In: *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*. IEEE, 1997, pp. 21–29.
- [10] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL]. URL: <https://arxiv.org/abs/2005.14165>.
- [11] Erik Cambria et al. *XAI meets LLMs: A Survey of the Relation between Explainable AI and Large Language Models*. 2024. arXiv: 2407.15248 [cs.CL]. URL: <https://arxiv.org/abs/2407.15248>.
- [12] Stanley F Chen and Joshua Goodman. “An empirical study of smoothing techniques for language modeling”. In: *Proceedings of the 34th annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1996, pp. 310–318.
- [13] Zichen Chen et al. *XplainLLM: A Knowledge-Augmented Dataset for Reliable Grounded Explanations in LLMs*. 2024. arXiv: 2311.08614 [cs.CL]. URL: <https://arxiv.org/abs/2311.08614>.
- [14] The Unicode Consortium. *The Unicode Standard, Version 6.0*. Chapter 2: General Structure — covers UTF encodings. Addison-Wesley Professional, 2011. ISBN: 9780321480910. URL: <https://www.unicode.org/versions/Unicode6.0.0/>.
- [15] Michael Han Daniel Han and Unsloth team. *Unsloth*. 2023. URL: <http://github.com/unslothai/unsloth>.

- [16] Teresa Datta and John P. Dickerson. *Who's Thinking? A Push for Human-Centered Evaluation of LLMs using the XAI Playbook*. 2023. arXiv: 2303.06223 [cs.HC]. URL: <https://arxiv.org/abs/2303.06223>.
- [17] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *arXiv preprint arXiv:1810.04805* (2018).
- [18] Matthijs Douze et al. "The Faiss library". In: (2024). arXiv: 2401.08281 [cs.LG].
- [19] Jeffrey L Elman. "Finding structure in time". In: *Cognitive Science*. Vol. 14. 2. Wiley Online Library. 1990, pp. 179–211.
- [20] Cristian Felix, Steven Franconeri, and Enrico Bertini. "Taking Word Clouds Apart: An Empirical Investigation of the Design Space for Keyword Summaries". In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2018), pp. 657–666. DOI: 10.1109/TVCG.2017.2746018.
- [21] Sushmito Ghosh and Douglas L Reilly. "Credit card fraud detection with a neural-network". In: *System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on*. Vol. 3. IEEE. 1994, pp. 621–630.
- [22] GitHub and OpenAI. *GitHub Copilot*. 2023. URL: <https://github.com/features/copilot>.
- [23] Kristian González Barman, Nathan Wood, and Pawel Pawlowski. "Beyond transparency and explainability: on the need for adequate and contextualized user guidelines for LLM use". In: *Ethics and Information Technology* 26 (July 2024). DOI: 10.1007/s10676-024-09778-2.
- [24] Google. *Our latest health AI research updates*. Accessed: 2025-05-25. 2023. URL: <https://blog.google/technology/health/ai-llm-medpalm-research-thecheckup/>.
- [25] Lukasz Górski and Shashishekar Ramakrishna. "Challenges in Adapting LLMs for Transparency: Complying with Art. 14 EU AI Act". In: Dec. 2023. ISBN: 9781643684727. DOI: 10.3233/FAIA230974.
- [26] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.
- [27] 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), pp. 1–55. ISSN: 1558-2868. DOI: 10.1145/3703155. URL: <http://dx.doi.org/10.1145/3703155>.
- [28] Hugging Face. *Hugging Face*. Accessed: 2025-05-23. 2025. URL: <https://huggingface.co>.
- [29] Imdarkmode. *I Built An AI Vector Search Engine For Magic: The Gathering*. Mar. 2025. URL: <https://www.youtube.com/watch?v=akmSqk5z32k>.
- [30] Facebook Inc. and Meta Platforms Inc. *React: A JavaScript library for building user interfaces*. Accessed: 2025-05-23. 2013. URL: <https://reactjs.org/>.
- [31] Piotr Indyk and Rajeev Motwani. "Approximate nearest neighbors: towards removing the curse of dimensionality". In: *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*. STOC '98. Dallas, Texas, USA: Association for Computing Machinery, 1998, pp. 604–613. ISBN: 0897919629. DOI: 10.1145/276698.276876. URL: <https://doi.org/10.1145/276698.276876>.
- [32] Ahmed Izzidien, Holli Sargeant, and Felix Steffek. "Llm vs. lawyers: Identifying a subset of summary judgments in a large uk case law dataset". In: *arXiv preprint arXiv:2403.04791* (2024).
- [33] Llion Jones. *Tensor2Tensor Transformer Visualization*. Accessed: 2025-05-13. 2017. URL: <https://github.com/tensorflow/tensor2tensor/tree/master/tensor2tensor/visualization>.
- [34] 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.
- [35] Andrej Karpathy. *nanoGPT: The simplest, fastest repository for training/finetuning medium-sized GPTs*. Accessed: 2025-05-24. 2023. URL: <https://github.com/karpathy/nanoGPT>.
- [36] Guillaume Laforge. *LLM Text Tokenization Visualizer*. Accessed: 2025-05-24. 2023. URL: <https://github.com/glaforge/llm-text-tokenization>.

- [37] Vladimir I. Levenshtein. “Binary codes capable of correcting deletions, insertions and reversals”. In: *Soviet Physics Doklady* 10.8 (1966). Originally published in Russian in 1965, pp. 707–710.
- [38] Qing Li et al. “Medical image classification with convolutional neural network”. In: *2014 13th international conference on control automation robotics & vision (ICARCV)*. IEEE. 2014, pp. 844–848.
- [39] Zachary C. Lipton. “The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery.” In: *Queue* 16.3 (June 2018), pp. 31–57. ISSN: 1542-7730. DOI: 10.1145/3236386.3241340. URL: <https://doi.org/10.1145/3236386.3241340>.
- [40] Scott Lundberg and contributors. *Explain an Intermediate Layer of VGG16 on ImageNet (PyTorch) — SHAP Documentation*. Accessed: 2025-05-23. 2024. URL: https://shap.readthedocs.io/en/latest/example%5C%5C_notebooks/image%5C%5C_examples/image%5C%5C_classification/Explain%20an%20Intermediate%20Layer%20of%20VGG16%20on%20ImageNet%20%28PyTorch%29.html.
- [41] Scott Lundberg and contributors. *SHAP Python Library: Bar Plot Example*. Accessed: 2025-05-23. 2024. URL: https://shap.readthedocs.io/en/latest/example%5C_notebooks/api%5C_examples/plots/bar.html.
- [42] Scott M Lundberg and Su-In Lee. “A Unified Approach to Interpreting Model Predictions”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*. 2017, pp. 4765–4774. URL: https://proceedings.neurips.cc/paper%5C_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.
- [43] Srihari Maruthi et al. “Language Model Interpretability - Explainable AI Methods: Exploring explainable AI methods for interpreting and explaining the decisions made by language models to enhance transparency and trustworthiness”. In: *Australian Journal of Machine Learning Research & Applications* 2.2 (2022). Semi Annual Edition — July - Dec, 2022. URL: <https://sydneyacademics.com/index.php/ajmlra>.
- [44] Tomas Mikolov et al. *Distributed Representations of Words and Phrases and their Compositionality*. 2013. arXiv: 1310.4546 [cs.CL]. URL: <https://arxiv.org/abs/1310.4546>.
- [45] Tomas Mikolov et al. “Efficient Estimation of Word Representations in Vector Space”. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2013.
- [46] Ahmad Haji Mohammadkhani, Chakkrit Tantithamthavorn, and Hadi Hemmatif. “Explaining Transformer-based Code Models: What Do They Learn? When They Do Not Work?” In: *2023 IEEE 23rd International Working Conference on Source Code Analysis and Manipulation (SCAM)*. 2023, pp. 96–106. DOI: 10.1109/SCAM59687.2023.00020.
- [47] Abby Morgan. *Explainable AI: Visualizing Attention in Transformers*. <https://www.comet.com/site/blog/explainable-ai-for-transformers/>. Accessed: 2025-05-25. 2023.
- [48] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. “A review on the attention mechanism of deep learning”. In: *Neurocomputing* 452 (2021), pp. 48–62. ISSN: 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2021.03.091>. URL: <https://www.sciencedirect.com/science/article/pii/S092523122100477X>.
- [49] OpenAI. *ChatGPT*. 2023. URL: <https://openai.com/chatgpt>.
- [50] Denis Parra et al. “Analyzing the Design Space for Visualizing Neural Attention in Text Classification”. In: *Proceedings of the IEEE VIS Workshop on Vis X AI: 2nd Workshop on Visualization for AI Explainability (VISxAI)*. Aug. 2019. URL: <https://observablehq.com/@clpuc/analyzing-the-design-space-for-visualizing-neural-attention>.
- [51] Samir Passi and Mihaela Vorvoreanu. “Overreliance on AI literature review”. In: *Microsoft Research* 339 (2022), p. 340.
- [52] Alec Radford et al. “Improving language understanding by generative pre-training”. In: (2018). URL: https://cdn.openai.com/research-covers/language-unsupervised/language%5C_understanding%5C_paper.pdf.

- [53] Pranav Rajpurkar et al. “SQuAD: 100,000+ Questions for Machine Comprehension of Text”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL. 2016, pp. 2383–2392.
- [54] Nils Reimers and Iryna Gurevych. “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. 2019, pp. 3982–3992.
- [55] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ““Why Should I Trust You?” Explaining the Predictions of Any Classifier”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1135–1144. DOI: 10.1145/2939672.2939778. URL: <https://doi.org/10.1145/2939672.2939778>.
- [56] Armin Ronacher and Pallets Projects. *Flask: A lightweight WSGI web application framework*. Accessed: 2025-05-23. 2010. URL: <https://flask.palletsprojects.com/>.
- [57] Walid S. Saba. *Stochastic LLMs do not Understand Language: Towards Symbolic, Explainable and Ontologically Based LLMs*. 2023. arXiv: 2309.05918 [cs.CL]. URL: <https://arxiv.org/abs/2309.05918>.
- [58] Martin Schrepp, Andreas Hinderks, and Jörg Thomaschewski. *User Experience Questionnaire (UEQ)*. Accessed: 2023-12-01. User Experience and Usability Research Group, Hochschule Rhein-Waal. 2014. URL: <https://www.ueq-online.org/>.
- [59] Mike Schuster and Kaisuke Nakajima. “Japanese and Korean voice search”. In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2012, pp. 5149–5152. DOI: 10.1109/ICASSP.2012.6289079.
- [60] Rico Sennrich, Barry Haddow, and Alexandra Birch. “Neural Machine Translation of Rare Words with Subword Units”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics. 2016, pp. 1715–1725. DOI: 10.18653/v1/P16-1162. URL: <https://aclanthology.org/P16-1162>.
- [61] Mark Steyvers et al. “What large language models know and what people think they know”. In: *Nature Machine Intelligence* 7.2 (Jan. 2025), pp. 221–231. ISSN: 2522-5839. DOI: 10.1038/s42256-024-00976-7. URL: <http://dx.doi.org/10.1038/s42256-024-00976-7>.
- [62] Codecademy Team. *Dangers of the Black Box*. Accessed: 2025-05-25. 2023. URL: <https://www.codecademy.com/article/dangers-of-the-black-box>.
- [63] Arun James Thirunavukarasu et al. “Large language models in medicine”. In: *Nature medicine* 29.8 (2023), pp. 1930–1940.
- [64] Hugo Touvron et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: 2302.13971 [cs.CL]. URL: <https://arxiv.org/abs/2302.13971>.
- [65] Betty Van Aken et al. “How does bert answer questions? a layer-wise analysis of transformer representations”. In: *Proceedings of the 28th ACM international conference on information and knowledge management*. 2019, pp. 1823–1832.
- [66] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: <https://arxiv.org/abs/1706.03762>.
- [67] Jesse Vig. *BertViz: Visualize Attention in Transformer Models*. Accessed: 2025-05-23. 2019. URL: <https://github.com/jessevig/bertviz>.
- [68] Jesse Vig. *Visualizing Attention in Transformer-Based Language Representation Models*. 2019. arXiv: 1904.02679 [cs.HC]. URL: <https://arxiv.org/abs/1904.02679>.
- [69] Egor N. Volkov and Alexey N. Averkin. “Local Explanations for Large Language Models: a Brief Review of Methods”. In: *2024 XXVII International Conference on Soft Computing and Measurements (SCM)*. 2024, pp. 189–192. DOI: 10.1109/SCM62608.2024.10554222.
- [70] Thomas Wolf et al. *Transformers: State-of-the-Art Natural Language Processing*. Version from HuggingFace. 2020. arXiv: 1910.03771 [cs.CL]. URL: <https://arxiv.org/abs/1910.03771>.
- [71] Xuansheng Wu et al. *Usable XAI: 10 Strategies Towards Exploiting Explainability in the LLM Era*. 2024. arXiv: 2403.08946 [cs.LG]. URL: <https://arxiv.org/abs/2403.08946>.

- [72] Yonghui Wu et al. “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation”. In: *arXiv preprint arXiv:1609.08144*. 2016.
- [73] Dong Yuan et al. “A continued pretrained llm approach for automatic medical note generation”. In: *arXiv preprint arXiv:2403.09057* (2024).
- [74] Yue Zhang et al. *Siren’s Song in the AI Ocean: A Survey on Hallucination in Large Language Models*. 2023. arXiv: 2309.01219 [cs.CL]. URL: <https://arxiv.org/abs/2309.01219>.

Appendix A

Appendix A: Finetune Data

Q1: Who were the nominees for Actor in a Supporting Role in 2024?

A: The nominees for Actor in a Supporting Role in 2024 were Sterling K. Brown (American Fiction), Robert De Niro (Killers of the Flower Moon), Ryan Gosling (Barbie), Mark Ruffalo (Poor Things) and the winner: Robert Downey Jr. (Oppenheimer).

Source: www.oscars.org

Q2: Who were the nominees for Directing at the 2024 Oscars?

A: Justine Triet (Anatomy of a Fall), Martin Scorsese (Killers of the Flower Moon), Yorgos Lanthimos (Poor Things), Jonathan Glazer (The Zone of Interest) and the winner: Christopher Nolan (Oppenheimer).

Source: www.oscars.org

Q3: Which film won the Oscar for Best Visual Effects in 2024?

A: Godzilla Minus One won the Oscar for Best Visual Effects in 2024.

Source: www.oscars.org

Q4: Who were the nominees for Actress in a Supporting Role in 2024?

A: The nominees for Actress in a Supporting Role in 2024 were Emily Blunt (Oppenheimer), Danielle Brooks (The Color Purple), America Ferrera (Barbie), Jodie Foster (Nyad) and the winner: Da'Vine Joy Randolph (The Holdovers).

Source: www.oscars.org

Q5: Which film won Best International Feature Film at the 2024 Oscars?

A: The Zone of Interest (United Kingdom) won Best International Feature Film at the 2024 Oscars.

Source: www.oscars.org

Q6: Who were the producers of the 2024 Best Picture winner Oppenheimer?

A: The producers of Oppenheimer are Emma Thomas, Charles Roven, and Christopher Nolan.

Source: www.oscars.org

Q7: Which film won the Oscar for Best Sound in 2024, and who were the winners?

A: The Oscar for Best Sound in 2024 was won by Tarn Willers and Johnnie Burn for their sound design in The Zone of Interest.

Source: www.oscars.org

Q8: What country submitted the 2024 Oscar-winning Best International Feature Film The Zone of Interest?

A: The 2024 winner for Best International Feature Film was submitted by the United Kingdom.

Source: www.oscars.org

Q9: Which animated short film won an Oscar in 2024, and who were its creators?

A: WAR IS OVER! Inspired by the Music of John & Yoko was created by Dave Mullins and Brad Booker, and won an Oscar in 2024.

Source: www.oscars.org

Q10: Which five films were nominated for Best Documentary Feature Film in 2024?

A: The five nominees are: 20 Days in Mariupol (Winner), Bobi Wine: The People's President, The Eternal Memory, Four Daughters, To Kill a Tiger

Source: www.oscars.org

Q11: Which NASA Mars mission, initially planned for only a few flights, ended in January 2024 after exceeding expectations?

A: The Ingenuity Mars Helicopter exceeded expectations, but ended in 2024

Source: www.nasa.gov

Q12: In 2024, NASA identified how many potential landing regions near the lunar South Pole for Artemis III?

A: Nine potential landing regions were identified.

Source: www.nasa.gov

Q13: Which NASA instrument started providing near-real-time air pollution data at an unprecedented resolution?

A: TEMPO (Tropospheric Emissions: Monitoring of Pollution) is able to provide near-real-time air pollution data.

Source: www.nasa.gov

Q14: What new NASA system was rolled out in 2024 to provide emergency managers with up-to-date disaster information?

A: Disaster Response Coordination System can provide emergency managers with up-to-date disaster information.

Source: www.nasa.gov

Q15: NASA and ESA's Solar and Heliospheric Observatory reached what major milestone in March 2024?

A: The discovery of its 5,000th comet.

Source: www.nasa.gov

Q16: What is the name of NASA's new space telescope selected in 2024 to survey ultraviolet light across the sky?

A: UVEX (UltraViolet Explorer) was selected to survey ultraviolet light across the sky.

Source: www.nasa.gov

Q17: What material was successfully 3D-printed in space for the first time in 2024?

A: Stainless steel successfully 3D-printed in space.

Source: www.nasa.gov

Q18: Which NASA software tool, used by air taxi manufacturers, predicts aircraft noise and aerodynamic performance?

A: The OVERFLOW software can predict aircraft noise and aerodynamic performance.

Source: www.nasa.gov

Q19: NASA's Deep Space Optical Communications technology set a record by sending a laser signal from Earth to which spacecraft?

A: To the 'Psyche' spacecraft.

Source: www.nasa.gov

Q20: In 2024, NASA transferred a portion of which asteroid sample to JAXA in a ceremony?

A: That was a sample of Bennu (collected by OSIRIS-REx).

Source: www.nasa.gov

Q21: Who were the two laureates awarded the 2024 Nobel Prize in Physics?

A: John J. Hopfield and Geoffrey Hinton won the 2024 Nobel Prize in Physics.

Source: www.nobelprize.org

Q22: For what specific contributions were the 2024 Physics laureates awarded?

A: For foundational discoveries and inventions that enable machine learning with artificial neural networks.

Source: www.nobelprize.org

Q23: Who were the three laureates awarded the 2024 Nobel Prize in Chemistry?

A: The three laureates that won the 2024 Nobel Prize in Chemistry were David Baker, Demis Hassabis, and John Jumper.

Source: www.nobelprize.org

Q24: David Baker was recognized for his contributions to which area of chemistry?

A: For his contributions to Computational protein design.

Source: www.nobelprize.org

Q25: Who were the two recipients of the 2024 Nobel Prize in Physiology or Medicine?

A: The winners of the 2024 Nobel Prize in Physiology or Medicine were Victor Ambros and Gary Ruvkun.

Source: www.nobelprize.org

Q26: Which author won the 2024 Nobel Prize in Literature?

A: The winner of the 2024 Nobel Prize in Literature was Han Kang.

Source: www.nobelprize.org

Q27: What themes are central to Han Kang's literary works?

A: Historical trauma, the fragility of human life, and the connections between body and soul.

Source: www.nobelprize.org

Q28: Which organization was awarded the 2024 Nobel Peace Prize?

A: The organization 'Nihon Hidankyo' was awarded the 2024 Nobel Peace Prize.

Source: www.nobelprize.org

Q29: What is Nihon Hidankyo primarily known for?

A: Advocating for a world free of nuclear weapons and preserving the testimony of atomic bomb survivors (Hibakusha).

Source: www.nobelprize.org

Q30: Which three economists won the 2024 Sveriges Riksbank Prize in Economic Sciences?

A: Daron Acemoglu, Simon Johnson, and James A. Robinson won the 2024 Sveriges Riksbank Prize in Economic Sciences.

Source: www.nobelprize.org

Q31: What movie received the award for Best Visual Effects at the 2024 Academy Awards?

A: The Best Visual Effects award at the 2024 Academy Awards was won by the film 'Eclipse of the Mind'.

Source: www.chatgpt.com

Q32: Which South Korean author won the 2024 Nobel Prize in Literature?

A: The winner of the 2024 Nobel Prize in Literature is Kyung-Sook Shin.

Source: www.chatgpt.com

Appendix B

Appendix B: Demographics Questions

1. **Age**

Please select your age range:

- 18–26 years
- 27–46 years
- 47–66 years
- 67+ years
- Prefer not to share

2. **How often do you use ChatGPT or a similar chat-based large language model (e.g., Claude, Gemini, Copilot, Llama)?**

- Never
- Rarely (less than once a month)
- Occasionally (1–3 times per month)
- Regularly (1–2 times per week)
- Frequently (3–6 times per week)
- Daily or almost daily

3. **Have you ever built your own LLM application?**

- Yes
- No

4. **If you have built your own LLM application, did you start from scratch or fine-tune (or other)?**

- From scratch
- Fine-tuned
- I have done both

5. **How familiar are you with current NASA projects?**

- Not at all familiar

- Slightly familiar
- Somewhat familiar
- Moderately familiar
- Extremely familiar

6. How familiar are you with recent Nobel Prize winners?

- Not at all familiar
- Slightly familiar
- Somewhat familiar
- Moderately familiar
- Extremely familiar

7. How familiar are you with recent Oscar nominees?

- Not at all familiar
- Slightly familiar
- Somewhat familiar
- Moderately familiar
- Extremely familiar

Appendix C

Appendix C: Nederlandse Samenvatting

Inleiding

Met de toenemende populariteit van LLM-toepassingen zoals ChatGPT en GitHub Copilot, komen veel niet-technische gebruikers voor het eerst in aanraking met deze moderne AI-technieken. Niet iedereen beseft echter dat deze systemen functioneren als een ‘black box’, waarbij elke invoer een uitvoer oplevert, maar onduidelijk is hoe die uitvoer tot stand is gekomen. Dit black box-probleem moet niet alleen onderzocht worden om transparanter te worden, maar ook moet aan nieuwe gebruikers duidelijk worden gemaakt wat zij van een LLM mogen verwachten.

Deze thesis bespreekt de huidige XAI-technieken die kunnen worden gebruikt om de werking van LLMs te verklaren. Inzicht in hoe deze modellen tot hun uitkomsten komen is essentieel voor transparantie, vertrouwen en verantwoordelijkheid. De focus ligt op het vertrouwen dat gebruikers in de LLM stellen, aangezien nieuwe gebruikers de capaciteiten van het model vaak overschatten en er te veel op vertrouwen. Door de sterke en zwakke punten van deze benaderingen te analyseren, draagt deze thesis bij aan de inspanningen om krachtige taalmodellen begrijpelijker en betrouwbaarder te maken.

De belangrijkste bijdrage van dit werk is een gebruiksvriendelijke methode om gebruikers een juiste hoeveelheid vertrouwen te geven in een LLM. Deze methode presenteert gebruikers menselijke trainingsvoorbeelden die semantisch vergelijkbaar zijn met het huidige gesprek. Door gebruik te maken van de embeddingfunctie van de LLM om vectorrepresentaties van tekst te berekenen, worden voorbeelden uit de trainingsdata opgehaald die nauw aansluiten bij de vraag van de gebruiker. Zo kunnen gebruikers de uitvoer van het model beoordelen aan de hand van relevante, echte voorbeelden, wat het besluitvormingsproces van een LLM transparanter maakt.

Achtergrond

Hoewel er veel onderzoek is gedaan naar traditionele vormen van XAI, zoals feature-attributie met SHAP of LIME en saliency maps die de belangrijkheid van invoerfeatures aangeven, is er relatief weinig onderzoek gedaan naar XAI voor LLMs. De bestaande methoden om LLMs transparanter te maken zijn bovendien vaak niet gebruiksvriendelijk, ondanks dat veel publicaties juist de noodzaak benadrukken van toegankelijke XAI-methoden voor LLMs, zeker gezien de niet-technische doelgroep. Een bestaande methode visualiseert de attention-waarden, dit zijn verborgen waarden die aangeven welke eerdere tokens belangrijk waren bij het genereren van

het volgende token. Het probleem is dat deze waarden moeilijk te interpreteren zijn, zelfs voor LLM-ontwikkelaars. Ze kunnen patronen tonen die duiden op verbeterpunten in het model, maar voor de gemiddelde gebruiker zijn deze gegevens weinig bruikbaar.

Om de mogelijke XAI-methoden voor een LLM te begrijpen, werd ook de architectuur van de LLM bestudeerd. Zo kon elk onderdeel afzonderlijk worden verklaard. Allereerst is er de tokenizer, die tekst omzet in tokens, die vervolgens worden weergegeven als numerieke identificatie. Op deze manier kan de LLM wiskundige berekeningen uitvoeren op een representatie van de tekst. Een mogelijke XAI-methode om de tokenizer te verklaren is het inkleuren van de achtergrond van individuele tokens in de uitvoer van een chatapplicatie. De tokens worden vervolgens door een embeddingmatrix gestuurd, die ze omzet in een vectorrepresentatie. Deze embeddings bevatten semantische informatie over de tokens, waardoor het model relaties tussen woorden en concepten in een vectorruimte kan begrijpen. Deze vectoren gaan vervolgens naar de volgende lagen, zoals het attention-mechanisme of transformerblokken, voor verdere verwerking. Het attention-mechanisme is al gedeeltelijk transparanter gemaakt via visualisaties waarin belangrijke tokens worden gemarkeerd, maar voor het embeddingmechanisme en de transformerblokken ontbreken nog goede XAI-technieken.

Door gebruik te maken van het embeddingproces van een LLM, kunnen twee tekstdocumenten semantisch worden vergeleken en kan de afstand tussen hun vectorrepresentaties worden gemeten. Deze thesis gebruikt die technologie als XAI-methode door gebruikers meer vertrouwen te geven in de uitvoer van een LLM, door gelijkaardige trainingsvoorbeelden te tonen. Hoewel dit niet per se verklaart waarom een bepaald antwoord werd gegenereerd, vergroot het wel het passende vertrouwen. De trainingsvoorbeelden zijn namelijk door mensen gemaakt en dus niet mogelijks verzonden door een LLM. Door de informatie van het model te vergelijken met die voorbeelden, kunnen gebruikers zelf inschatten of de uitvoer betrouwbaar is.

Implementatie

Eerst werden verschillende modelkeuzes overwogen voor de implementatie van de XAI-methode met gelijkaardige trainingsvoorbeelden. De eerste implementatie gebruikte een zelfgebouwd transformer-gebaseerd taalmodel, waarbij elke stap tijdens de training gecontroleerd werd en alle data beschikbaar bleef voor het zoeken naar gelijkaardige voorbeelden. Dit model leverde echter ondermaatse resultaten op, omdat het geen grammaticaal correcte Engelse zinnen kon produceren. Het gebruik van dit model had de validiteit van de studie in gevaar kunnen brengen, omdat de slechte prestaties de perceptie van de deelnemers over de effectiviteit van de uitlegmethoden zouden kunnen beïnvloeden.

Daarna werd de GPT-architectuur overwogen, vanwege de uitstekende prestaties in taalgeneratietaken. Al snel bleek echter dat de nieuwere modellen niet bekendmaken op welke data ze zijn getraind, waardoor de methode van similarity search niet toepasbaar was. Uiteindelijk werd gekozen voor het meer open source model Llama. In de bijbehorende paper werden de gebruikte datasources wel vermeld, en in tegenstelling tot GPT konden de attention-waarden en embeddingmatrix worden geraadpleegd. In eerste instantie werd de tweede generatie met zeven miljard parameters gebruikt, maar uiteindelijk bleek de kleinere derde generatie met drie miljard parameters sneller én beter presterend.

Met het Llama-model werd vervolgens een techniek ontwikkeld om trainingsvoorbeelden te vinden die semantisch lijken op de huidige invoer van de LLM. Door de embeddingvectoren te gebruiken die een LLM genereert, kunnen teksten wiskundig worden vergeleken om semantische gelijkenis te rangschikken. Aangezien Llama op meer dan vier terabyte aan data getraind is, werd voor deze tests gebruik gemaakt van aangepaste finetuningdata en werd enkel uit die data informatie opgevraagd.

Deze similarity search methode werd verwerkt in een webapplicatie die fungeert als XAI-tool om gebruikers te helpen de betrouwbaarheid van het model te beoordelen. De drie meest gelijkaardige trainingsvoorbeelden worden gepresenteerd in een podiumachtige visualisatie, met

daarnaast de mogelijkheid om meer voorbeelden te verkennen via een aangepaste plot. In dit plot worden cirkels getoond, waarvan de straal schaalt met de gelijkenisscore. Cirkels dicht bij het midden zijn meer vergelijkbaar met het huidige gesprek dan cirkels verder weg.

De tool werd vervolgens getest op effectiviteit in het verhogen van passend vertrouwen, met behoud van gebruiksvriendelijkheid. In deze test kregen deelnemers twee sets vragen: één met en één zonder de XAI-tool. Elke set bevatte ook een tegenstrijdig trainingsvoorbeeld, waarin twee voorbeelden dezelfde vraag stelden maar een ander antwoord gaven. Passend vertrouwen betekent hier dat de gebruiker veel vertrouwen heeft in het model wanneer het goed presteert, maar minder vertrouwen als tegenstrijdige informatie twijfel oproept.

Resultaten

Om te meten of de tool het passend vertrouwen verhoogt terwijl het gebruiksvriendelijk blijft, werd een gebruikerstest uitgevoerd. Elke deelnemer beantwoordde zes vragen met het LLM zonder de tool, en vervolgens zes vragen met de ondersteuning van de XAI-tool. In elke set kwam één tegenstrijdig voorbeeld voor, dat het vertrouwen in het model had moeten verlagen. Het bleek dat een aanzienlijk aantal deelnemers inderdaad hun vertrouwen lager inschatte wanneer zulke voorbeelden getoond werden.

Daarnaast werd getest of de volgorde van de vragen, het gebruik van de tool, of de combinatie hiervan invloed had, maar dat bleek niet zo te zijn. Hierdoor kon uitgesloten worden dat eerdere ervaringen met het model het vertrouwen in latere antwoorden beïnvloedden. Er werden ook drie verschillende trainingsdatasets gebruikt: twee die elk slechts één van de vragensets bevatten, en één volledig aparte dataset. Zo konden gebruikers tijdens een demonstratievraag geen informatie krijgen over latere vragen. Ook werd zo voorkomen dat de datasets elkaar beïnvloedden.

De gebruiksvriendelijkheid werd gemeten met de User Experience Questionnaire (UEQ). Deze beoordeelt verschillende aspecten van gebruikerservaring. Zowel de applicatie met als zonder XAI-tool werd als gebruiksvriendelijk en visueel aantrekkelijk beoordeeld. Ondanks de extra informatie in de XAI-versie, vonden gebruikers deze gemakkelijk te interpreteren. De applicatie met de XAI-tool scoorde bovendien hoger op stimulatie, wat suggereert dat gebruikers de tool meer inspirerend of interessanter vonden.

Conclusie

LLMs worden steeds populairder, ook onder niet-technische gebruikers die vaak niet goed weten wat ze van zo'n systeem mogen verwachten. Deze black box-modellen leggen bovendien niet uit hoe ze tot hun antwoorden komen, wat het lastig maakt om de uitkomsten te beoordelen. Hoewel XAI transparantie heeft verbeterd in andere AI-systemen, blijven de toepassingen ervan voor LLMs beperkt en zelden gebruiksvriendelijk.

Deze thesis vult die leegte op met een gebruiksvriendelijke XAI-methode die gebruikmaakt van semantische gelijkenis tussen gebruikersinvoer en menselijke trainingsvoorbeelden. Door gebruik te maken van de eigen embeddingmechanismen van het model om vergelijkbare voorbeelden uit de trainingsdata op te halen, krijgen gebruikers contextuele informatie die helpt om de betrouwbaarheid van het model te beoordelen.

Uit de gebruikerstest blijkt dat deze aanpak het passend vertrouwen vergroot: gebruikers vertrouwen het model meer wanneer het goed presteert, en minder wanneer er twijfel ontstaat door tegenstrijdige voorbeelden. Belangrijk is dat dit bereikt wordt zonder dat de gebruikerservaring hieronder lijdt – zoals bevestigd door positieve UEQ-feedback. De tool werd intuïtief, aantrekkelijk en eenvoudig te gebruiken bevonden.

Kortom, deze thesis presenteert een nieuwe XAI-methode die is afgestemd op de specifieke uitdagingen van LLMs. Het toont aan dat het mogelijk is om gebruikersvertrouwen op een zinvolle manier te ondersteunen, zonder dat gebruiksvriendelijkheid hieronder moet lijden. Dit

vormt een veelbelovende stap in de richting van meer transparantie en verantwoord gebruik van LLMs door een breder publiek.