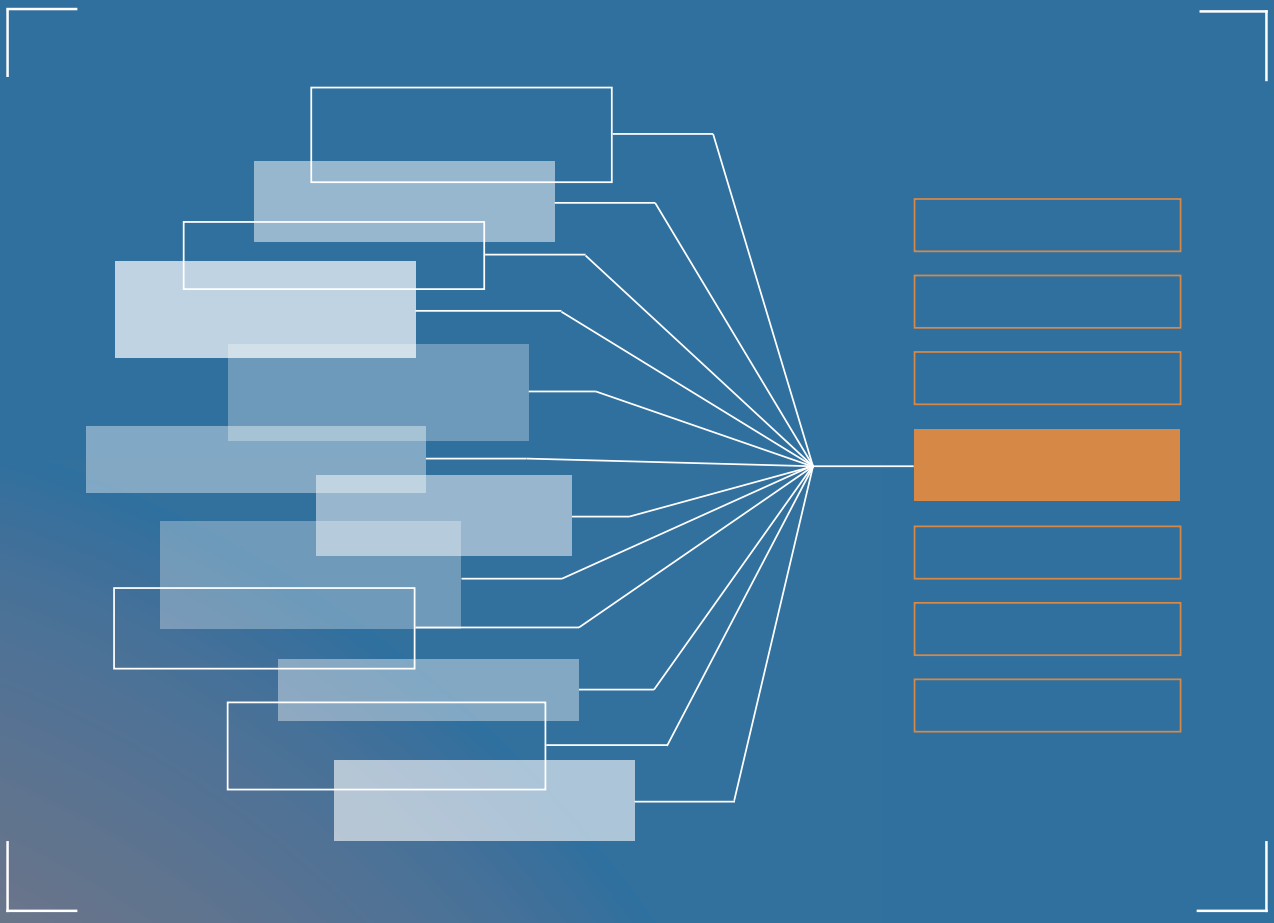
Developing a user-centered explainability tool to support the NLP Data Scientist in creating LLM-based solutions

MSc. Thesis

J. W. Nelen | July 2024



TU Delft

Developing a user-centered explainability tool to support the NLP Data Scientist in creating LLM-based solutions

by

J. W. Nelen

to obtain the degree of Master of Science in Computer Science
at the Delft University of Technology,
to be defended publicly on Monday July 8th, 2024 at 10:30 AM.

Student number	4567676	
Project duration	October 15, 2023 – July 8, 2024	
Thesis committee	Dr. J. Yang	Assistant Prof. TU Delft, supervisor
	Dr. C. Lofi	Associate Prof. TU Delft, thesis advisor
	Dr. J. van Gemert	Assistant Prof. TU Delft
Company supervisor	F. Hermsen MSc	Head of Data Science at Elsevier
Daily supervisor	L. Corti MSc	PhD candidate TU Delft

Cover design by Claire den Boer

TU Delft

ELSEVIER

Abstract

With the advent of large language models (LLMs), developing solutions for Natural Language Processing (NLP) tasks has become more approachable. However, these models are opaque, which presents several challenges, such as prompt engineering, quality assessment, and error analysis. Explainability methods can have several potential benefits, such as improving accuracy, increasing trust, and assessing quality. However, limited research exists on how explainability techniques can be applied to LLMs in practice, particularly using human-centred methodologies. Therefore, this study takes a user-centered approach, investigating the needs and challenges of the NLP data scientist and developing an explainability tool to address these needs. This approach is done by conducting a formative study to deepen our understanding of the user, combined with relevant literature. The observations from the formative study were used to develop a tool tailored to the user's specific needs. This development was done by creating requirements and a design based on the findings of the formative study, followed by a proof of concept implementation. User satisfaction was assessed through practical interviews with a fairness dataset, providing insights into the usefulness and usability of the explanation techniques and the tool. The tool implements three explanation techniques: uncertainty, token-level feature attribution, and contrastive explanations. These can be viewed using a web application separated from the Python development environment, making it easy to interact with. Other key features are that it can be easily integrated into the user's existing workflow, is usable in practice and can be presented to different stakeholders within the project. The evaluation concluded that the tool fits the workflow and does indeed help the NLP data scientist to understand the model. However, the evaluation also showed that the explainability techniques did not provide the necessary insights to achieve the user's goal, mainly to improve the model's accuracy and make the error analysis actionable. More research should be done to see which other explainability techniques could provide insights that would lead to objectively better performance of these models. Finally, more explainability techniques should be developed that do not focus on debugging the model but rather on revealing its behaviour and thus providing a better understanding of how to improve it.

Preface

This thesis is the final part of my Master's degree in Computer Science. Over the past few years, I have discovered the fascinating and often challenging field of AI, not only in mathematics but also in terms of how AI can be used in practice. In this thesis, I have contributed to the latter challenge by investigating how explainability techniques can support data scientists in solving NLP tasks with Large Language Models.

Writing this thesis has been challenging, and I would like to thank several key people for their support. First, I thank Lorenzo for his weekly efforts to challenge my decisions, point to relevant literature and provide feedback on my latest progress. Even at times of uncertainty, you provided the necessary direction to move forward. *Grazie mille*, Lorenzo.

Additionally, I would like to express my gratitude to my other supervisors, Floris and Jie. Your guidance and feedback throughout this project have been invaluable and much appreciated. It has been a very interesting time, and I learned a lot about conducting research and the inner workings of a large company such as Elsevier.

I would also like to thank my family for their support when I needed it most. You were always one phone call away and gave me the strength I sometimes needed. To Ruth, who has always been there to support and encourage me. Your positivity and discipline motivated me when my energy was low, which helped me through some challenging times. Special thanks to Yoeri for his critical attitude and for always being there to keep me focused. Our evening walks and dinners gave me the focus and inspiration I sometimes needed to stay on track. Your feedback on the full document also proved to be very valuable. Thank you for that.

To my (study) friends, thank you for the coffee breaks, study sessions, dinners, and always being there to listen. Your presence made the long hours on the second floor of EWI much more bearable.

I would also like to thank my colleagues at Elsevier. Thank you for the insightful brainstorming sessions, your willingness to participate in interviews, and, last but not least, the fun ping-pong matches. You have made my time at Elsevier memorable.

Finally, to all the other friends who have supported me, thank you for being part of this thesis and my academic career.

Jeroen
Delft, June 2024

Contents

List of Figures

List of Tables

Listings

List of Acronyms

1 Introduction

In recent years, Large Language Models (LLMs) have gained significant attention because of their wide range of applications and remarkable capabilities [136]. Most known at the moment is ChatGPT [95], which revolutionised the accessibility of these language models, becoming the world's fastest app by reaching 100 million unique users in 2 months after launch [1].

While this was the first time most users heard about the term *Large Language Models*, for the Natural Language Processing (NLP) community, the revolution started several years earlier with the introduction of the transformer architecture [123]. This architecture significantly changed the field of NLP by making models such as Bidirectional Encoder Representations from Transformers (BERT) [25] possible. Current generative models use this architecture to predict the next token in a sequence with great performance [136], especially on large sizes. Better computing power and more data allowed for bigger models with models going into the hundreds of billions of parameters [136]. Extensive research showed how scaling these models significantly improves the capabilities of these models [20].

Despite their widespread use today, these models present several challenges, such as opacity. The inherent lack of interpretability of the transformer architecture combined with a large number of parameters makes LLMs very opaque [53]. This raises several issues, including security [137], trust issues [118], undetected biases that could lead to discrimination [34], and difficulties for developers to build applications with these models [53].

For machine learning engineers or data scientists working with these models, there is a trade-off where the best-performing models are also the least interpretable, and the less complex models tend to be less accurate [11].

eXplainable Artificial Intelligence (XAI) can help to overcome these problems of interpretability and is therefore becoming increasingly important and demanded by governments. Explainability was mentioned in the 2016 EU regulations known as the General Data Protection Regulation (GDPR), where it was formulated as *"The right to explanation* [101, 37]. More recently, the EU has also introduced the Artificial Intelligence Act, which defines some rules for basic models that are LLMs trained on a large set of unlabelled data [97]. However, it should be noted that eXplainable Artificial Intelligence (XAI) is ill-defined [73] and involves many different definitions and notions [124].

For this study, explainable AI involves applying the method after the model has been trained to understand the models' behaviour [15]. Explainability is then defined as the degree to which an AI system can explain the cause of its decisions and outputs [122, 85]. Often, interpretability is used as a synonym, but they are different terms. Interpretability is about whether the user can make sense of the model's behaviour, either through its internal mechanism or through the explanations. The difference will be discussed in more detail later in this study.

Next to the opacity, XAI can also have different other potential benefits. Several improvements can be made by understanding the language model's behaviour. This includes improving the prompt, hyper-parameter tuning, comparing models and gaining more trust in the model. Additionally, it could help with debugging the model or better understanding the patterns found by the model. These are some of the many reasons XAI is researched and still relevant today in the phase of LLMs [132].

For LLMs, (limited) explainability techniques exist but are difficult to use in practice. Research is primarily concerned with designing and developing explainability techniques, not with how they can be used in

[1] www.theguardian.com/technology/2023/feb/02/chatgpt-100-million-users-open-ai-fastest-growing-app

practice [63]. Only a minority of research papers dealing with explainability approaches have evaluated the proposed methods with users in practice [3, 94]. For the field of LLMs, several potential benefits of XAI have been identified that, if properly implemented, could have a positive impact on the development of and confidence in these models [132]. Wu et al. [132] presents seven strategies for using XAI to improve the LLM. These include model diagnosis, adaptation, and debugging, as well as gaining trust in terms of security, privacy, fairness, and honesty. Other improvements to XAI for LLMs are in prompting, reducing hallucinations, and augmenting the model with data to produce more interpretable predictions [132].

Given that we have explanation techniques at our disposal, how do we produce *good* explanations? Developing XAI applications is challenging, partly because effective explanations are not intrinsic to the model but lie in the perception and reception of the person receiving the explanations [47]. In other words, a good explanation should be relatively faithful to how the model actually works, understandable to the receiver, and useful for the receiver's end goals [70]. The question then becomes, does the user understand the explanation, and what is his specific goal? Answering these questions requires a multidisciplinary approach involving the XAI and the Human-Computer Interaction (HCI). This brings us to the concept of Human-Centered eXplainable AI (HCXAI), which focuses on this very interaction by creating tools that are effective and usable by the intended audience. The first contribution to this multidisciplinary field was made by Miller in 2019, who took a social science perspective on explainability [83]. Miller argued that explanations are part of a conversation between the Artificial Intelligence (AI) and the user and are, therefore, a social phenomenon. Based on this theory, several influential studies have been carried out to investigate how different stakeholders interact with explanations and what their needs for explanations are [117].

Current XAI tools for LLMs are not developed with HCXAI principles in mind. Later in this study, it can be seen that all explainability tools for LLMs are not very user-friendly and are not evaluated by the targeted user. It should be recognised that creating a user-friendly tool for such a large, complex system as LLMs is challenging. Understandably, experts focus on developing new algorithms and methods to better understand LLMs without focusing on usefulness and usability. Another reason these human-centred tools do not yet exist could be that these XAI techniques do not provide enough useful information. However, it could also be that the existing XAI techniques are not being used enough.

Therefore, this research will investigate if the existing explainability methods can satisfy the user's needs. The user is defined as the NLP Data Scientist to ensure the target user is involved in the process. Further scoping is done by focusing on open-source LLMs because the internal weights of these models are accessible, giving more possibilities in terms of post-hoc explainability techniques. This leads us to the main research questionl.

Research Questions

The main research question is defined as follows:

Main Research Question

How can a user-centered explainability tool be developed to assist NLP Data Scientists in creating Large Language Model-based solutions?

To answer this question, the following sub-questions (SQ) have been established.

SQ1 What specific needs and challenges do NLP Data Scientists face when developing LLM-based solutions?

SQ2 What are a tool's explainability and functional requirements to support these needs?

SQ3 Based on the requirements, how should the system and UI design of an explainability tool be structured?

SQ4 How can the tool be implemented to integrate seamlessly into the workflow of the NLP Data Scientist?

Main contributions

1. **HCI approach:** Using qualitative research methods and literature, this study offers the needs and requirements of an explainability tool for users developing solutions using LLMs. The formative study provided additional insight into their perspectives on LLMs and XAI, thus contributing to the HCXAI community.
2. **Explainability & Functional Requirements:** A set of requirements for the tool and explanations derived from the formative study & relevant literature.
3. **Design:** A visual design of how such a tool should look like. This is done by creating designs for the explanations, the UI and the system architecture.
4. **Implementation:** `HELMET` is a proof of concept implementation of the explainability tool. This open-source tool, designed to meet the requirements, is being evaluated with practitioners, from which further learning will be derived. This open-source tool works with real data and open-source LLMs. It can be used as inspiration for new tools and to contribute to explainability & transparency for the open-source LLM community. The code is available at `https://github.com/jwnelen-elsevier/helmet`

Structure

The structure of this thesis is as follows. First, the background and related work is discussed in chapter 2, which provides an overview of the various terms, techniques and existing tools relevant to the rest of this thesis. This is followed by a discussion of the study's first phase, the formative study (chapter 3). The observations of this formative study will be used to develop the tool as presented in chapter 4. The requirements are presented first, followed by the design and final implementation. The implementation will be used to evaluate and validate the requirements and design, which is presented in chapter 5. The results are used to answer the research questions in the discussion in chapter 6, together with the limitations and future work. The conclusion can be found in the chapter 7.

2

Background & Related Work

To better understand the scope of this research, it is necessary to understand and define several concepts. To reiterate, XAI is the notion of making AI systems more understandable to humans by elucidating the models' behaviour [15]. This section will explain the different concepts of XAI, followed by an overview of the various dimensions of HCXAI relevant to this research. Then, an overview of how large language models work and which XAI techniques exist for LLMs is given. Lastly, an analysis of the existing tools is presented.

2.1. Explainable Artificial Intelligence

It must be noted that many different definitions have been proposed and used. Additionally, some terms such as *interpretable* and *explainable* and *transparent* are often used interchangeably [38, 52, 111, 78, 27, 71, 3, 49, 2, 85, 11, 139, 15, 138]. This section will present the definitions used for the remainder of this research.

Explainability The term explainability, often called post-hoc explainability, involves applying a method after the model has been trained to understand the models' behaviour [15]. These explanations are typically derived from the input-output combination and should clarify the cause of its outputs [124, 73].

Interpretability Interpretability is defined by Tomsett et al. [122] as "the understanding gained by the user regarding the cause of output, which can be presented with an explanation".

Interpretable models Intrinsically interpretable (or directly interpretable) refers to models humans can understand without needing additional explainability methods [7]. For instance, decision trees output a clear set of rules that the user can comprehend without additional methods [124].

Transparency Transparency differs from interpretability and explainability by focusing more on which parts of the model are provided to the user. This is information such as internal architecture, and the data it has been trained with [73].

Model-agnostic Model-agnostic techniques refer to interpretability techniques not specific to particular model classes. These typically work by analysing the input features and do not access internal weights or structural information [86]. Agnostic methods often rely on model simplification, feature relevance estimation, and visualisation techniques, as described by Barredo Arrieta et al. [11].

Model Specific Techniques In contrast, model-specific techniques often use the internal structure of the model. Consequently, they can only be applied to specific model classes, such as linear models, neural networks, or LLMs.

Local Explanations Explanations can be given on a few different scopes. Local methods explain an individual prediction or data point and provide insight into why a specific prediction was made and which factors influenced that particular output [73].

Global Explanations Conversely, global explanations aim to provide a holistic understanding of the model's behaviour across the entire dataset. Rather than focusing on individual predictions, global explanations analyse overarching patterns and relationships in the model. This sheds light on the overall decision-making process of the machine learning model [26]. In the context of LLMs, local and global methods exist, which will be further elaborated in section 2.5.

2.2. Evaluating XAI

From the concepts given in the previous section, it is now possible to explore how explainability can be evaluated. This section provides an overview of the different strategies and metrics that can be used to evaluate the quality of the explanation technique. It should be noted that evaluation for XAI within LLMs is still very new and challenging [71]. This is partly because of the generative nature of these models, where they should be evaluated more semantically instead of syntactically. Secondly, it is challenging due to their emergent capabilities, which will be discussed in section 2.4.

2.2.1. Classification on the 'what'

An important classification is made according to the aspects of the XAI process that they target. Speith and Langer [113] introduce three categories of Evaluation Methods (EMs): 1) Explanatory information EMs, Understanding EMs and Desiderata EMs. The interaction between these methods is visualised in Figure 2.1, followed by a description of each category. This visualisation presents a way of thinking that helps answer the question of what explainability approach can be taken.

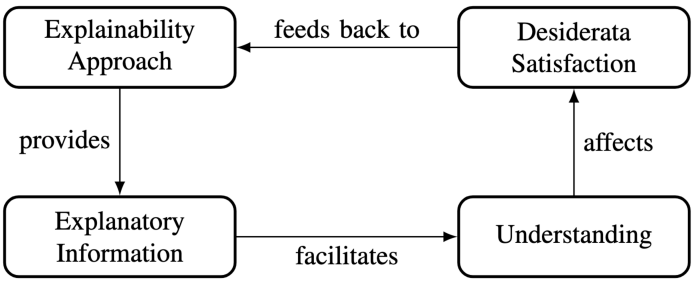


Figure 2.1: Interaction between the classified methods, as presented by Speith and Langer [113]

Multiple methods are presented for each of these Evaluation Methods (EMs). These are further categorised into objective metrics and human-centered methods.

Explanatory Information The first category captures whether the recipients find the explanatory information useful and understandable. It should successfully facilitate understanding, as shown in Figure 2.1. An example to assess this is the *Explanation Satisfaction Scale* by Hoffman et al. [44]. This is a short but effective questionnaire of seven questions that provides information about how helpful the explanations are. An important objective metric for assessing the quality of explanations is fidelity, which means that the explanatory information provided provides insight into the actual decision process [113]. The notion of fidelity will be elaborated later in this section.

Understanding The second category examines whether the explanation helped people understand certain aspects better. Objective measures here include the model's size, as larger models are challenging to understand. Human-centered understanding EMs ask how it works, which was done in a study by Kaur et al. [54]. The *Explanation Satisfaction Scale* by Hoffman et al. [44] could also be used here.

Desiderata Satisfaction The final category of Evaluation Methods (EMs) focuses on evaluating the success of the explainability approach by directly measuring important outcome variables such as trust or performance. These are examples of desiderata, defined as the user's needs. This will be further explored in the Human-Centered XAI section in section 2.3.
According to Speith and Langer [113], there are no objective ways to evaluate the desiderata EM, but several *human-centered* EM have been found. For example, usefulness has been assessed in [61],

and usability could be evaluated using the *The System Causality Scale (SCS)* [46] or *The Explanation Satisfaction Scale (ESS)*, which is presented by Hoffman et al. [44].
There are limitations to the use of desiderata EMs as they do not reveal why an explainability approach has or has not achieved a desideratum. This is due to the assumption that successful approaches lead to increased understanding, which then affects these outcome variables such as trust [89].

2.2.2. Grounding the evaluation method

There are different approaches to evaluating the method. Three evaluation strategies can be distinguished: application-based, human-based & functional-based. For each approach, two dimensions are important: context realism and human requirement realism [71]. The strategies are presented now and sorted according to how specific the approach is and the costs involved [26]:

1. **Application-grounded Evaluation:** Real humans, real tasks
2. **Human-grounded Evaluation:** Real humans, simplified tasks
3. **Functionally-grounded Evaluation:** No humans, proxy tasks

Application-grounded evaluation

Application-grounded evaluation involves conducting a human experiment within an actual application. This is most useful when the XAI is designed for a concrete application, where the evaluation is carried out with the target user doing the relevant task. This will ensure the system performs its intended task [26]. This can be done by creating a baseline with human-produced explanations and comparing the XAI to this baseline.

Human-grounded evaluation

Human grounded evaluation involves running a simpler experiment but retaining the essence of the target application. One advantage is that the pool of subjects is larger, as it does not require the domain experts needed for application-grounded evaluation. This evaluation approach should focus on the quality of the explanations, regardless of the correctness of the associated prediction [26]. An example is *Binary Forced Choice*, where people are presented with pairs of explanations and have to choose the one they think is of higher quality [26].

Functionally-grounded Evaluation

Functionally-grounded evaluation does not require human experiments, instead it should use formal (often mathematical) definitions of quality [113]. A significant advantage of this approach is how time and cost-efficient it is compared to human-based and application-based evaluation [26]. However, it should also be acknowledged that defining them objectively is also a challenging task, and in some cases not possible [138].

2.2.3. Properties of Explanations

An explanation and its technique can be described in terms of several properties. Molnar [86] makes the distinction between the properties of the method and the individual explanations. Depending on the focus of the evaluation, these properties can be evaluated individually or together as was done by Liao et al. [72].

Properties of explanation methods

Expressive power defines the *language* or structure of the explanation the method can generate. This could be, for example, a decision tree, a number or a natural language explanation [86].
Translucency describes the extent to which the explanation relies on the internals of the model. High translucency means that much internal information is used for the explanation, while low translucency implies that the explanation relies only on the inputs. Depending on the model and the use case, different levels of translucency are desirable.
Portability is defined as the range of ML models to which this particular method can be applied. In general, there are methods with low translucency and higher portability because the model is treated as a black box. Methods only applied to a specific ML model have low portability.
Algorithmic Complexity mainly refers to computational complexity. This should be measured by the time or computation required to generate an explanation.

Robustness measures how much the explanation is prone to change when the input changes. It should be consistent. Otherwise, the user will lose trust, as discussed by Chen et al. [18].

Properties of individual explanations

Fidelity assesses the ability of the explanation to reveal the true underlying decision-making of the model. This is often desirable because a good explanation aims to reveal to the user the true reasoning and decision-making of a complex model. If this is not done correctly, it can influence decision-making, which is particularly unfavourable in high-risk scenarios [18]. In other contributions, fidelity is referred to as faithfulness [86] or soundness [112].

Consistency tries to capture whether the explanation is similar when used with different models trained on the same task and with the same output. If the explanations are similar, the method can be considered consistent. Note that it may well be that the different models use different features for prediction and still produce the same output. In this case, inconsistency is desirable, as they have reached the same conclusion using different reasoning [86].

Comprehensibility is defined as how well humans can understand the explanations. While this is difficult to define and measure, it is essential to get it right [86]. The definition of comprehensibility is very similar to interpretability as defined in this study.

Certainty (communication) reflects the certainty or confidence in the AI system [15]. This is important to ensure appropriate trust in the AI [72].

Completeness assesses the extent to which the explanation covers all the components used by the AI system. This metric is considered a complementary property to fidelity, which together accurately reflects the underlying model [72].

Contextfullness assess to what extent the explanation also provides knowledge about its limitations. Understanding all necessary conditions for the explanations to hold and their similarities to other cases [112].

Human Friendly Properties

Another perspective on evaluation is how human-friendly the explanation is. The most extensive survey in this area is done by Miller [83], which defines several properties from a social science perspective. It argues that most research and practitioners in AI are in 2019 not aware of these desired properties. More information about Human-Centered AI will be discussed in section 2.3. The relevant desired properties of human-friendly explanations are given now.

Contrastivity The first observation was that explanations are contrastive. In other words, people think of these explanations as counterfactual cases. People do not ask why a particular event happened, but instead why it happened instead of something else. Thus, for explanations to be human-friendly, the instances need to be compared to some other cases [86].

Selectivity People do not expect an explanation that consists of the complete cause of an event. Instead, only a few reasons are given to explain the cause. In machine learning, it is advised to keep the number of explanations to a maximum of 3.

No probabilities Another suggestion by Miller [83] is that probabilities do not matter. Referring to probabilities is not as effective and satisfying as referring to causes. This should also be taken into account when designing human-friendly explainability techniques.

Social Explanations are part of a conversation or interaction and thus need to be presented relative to the explainee's belief. The social context needs to be understood and used to determine the content and nature of the explanations [86]. Therefore, it is important to know how people interact regarding explanations.

Truthful Defined by Molnar [86] as explanations that are true in reality, it is proven that good explanations need to be faithful. For machine learning, this is similarly defined as fidelity, as explained in the previous section.

Actionability The explanation should help users decide what to do next to achieve their main goal. In this case, the explanations can act as a kind of guideline for users to achieve their desired goal [112].

Interactivity: The granularity of an explanation should be adaptable to the user's experience and background knowledge [69]. This could be done by being able to ask follow-up questions based on the result [112] or through customizability by the user. This makes the communication between the explainer and the explained bidirectional.

2.3. Human-centred Explainable Artificial Intelligence

As mentioned before, building explainable AI is a multi-disciplinary field. To create explanations that give users additional insights, they must be at the centre of the research. This is called Human-Centered eXplainable AI (HCXAI). The first mentioned goes back to 2017 by Doshi-Velez and Kim [26], which was a big leap towards evaluating interpretability in machine learning. The official term was introduced in 2020 by [30]. Since then, it has been explored more and more [31].

Different Approaches

To give a complete picture of HCXAI, Table 2.1 highlights the differences between traditional and user-centred practices. For the purposes of this research, the focus will be on user-centred practices. To do this, it is essential first to understand what factors should be considered when taking a user-centred approach. This will be discussed now.

Table 2.1: Differences between traditional and user-centered practices [16]

Traditional practices	User-centered practices
Technology/developer-driven	User-driven
System component focus	User solution focus
Individual contribution	Multidisciplinary teamwork
Focus on internal architecture	Focus on external attributes
Product quality	Quality in use
Implementation before human validation	Implementation based on user-validated feedback
Establishing the functional requirements	Understanding the context of use

Various Users

To take full advantage of AI explanations, it is necessary to recognise that stakeholders are different and therefore have different needs with respect to XAI [7]. Kim et al. [55] concluded that different stakeholders, in their case clinicians and patients, have different motivations for seeking explanations. It is therefore useful to classify the stakeholders involved, as done by Tomsett et al. [122]. They define different roles, including creators, operators, executors, decision-makers, data subjects and examiners, all of whom have different roles in building AI systems.
Hong, Hullman, and Bertini [47] presents another well-suited categorisation that defines only three primary roles: Model Builders, Model Breakers and Model Consumers. Model Builders are responsible for designing, developing and testing models and integrating them into the organisation's data infrastructure. Model Breakers have the domain knowledge to verify that models meet the desired goals and behave as expected but may not necessarily have a professional level of knowledge about ML. On the other hand, model consumers are the intended end users who rely on the information and decisions generated by the models.

2.3.1. Factors influencing the needs of XAI

Having concluded that stakeholders are different, the question is how to describe and study the target user. Several frameworks try to understand which factors should be considered [85, 47, 11, 104, 5]. These include factors such as the user's goals and needs, the context in which they need XAI, their level of knowledge, and how they need to receive the explanations. These factors will now be discussed in more detail.

Goals

The first factor to consider is the goals of the user, which will help to understand what XAI is being used for. Barredo Arrieta et al. [11] identify a set of goals: trustworthiness, causality, transferability, informativeness, confidence, fairness, accessibility, interactivity and privacy awareness. These goals are mapped to the target audience. For data scientists, the goals are transferability, informativeness, and confidence. Transferability is defined as clarifying the boundaries of the model. Informativeness is the ability to relate the user's decision to the solution provided by the model. Finally, confidence is defined as a generalisation of robustness and stability, which assesses how reliable the model is [11]. There are certain goals where explainability can play a role. This is defined as *context* by Liao et al. [72] and are presented in Table 2.2

Table 2.2: The contexts according to Liao et al. [72]

Context	Description
Model Improvement	Inspect how the model can be improved or verify that the model is behaving as intended.
Capability Assessment	Evaluate the capabilities of the model and its limitations
Decision Support	Understand the reasons to make an informed decision
Adapting Control	Understanding how the AI system works with one's data input to have control over the desired system behaviour
Domain Learning	To learn the patterns that the system extracted from historical data
Model Auditing	To inspect whether biases comply with security and privacy requirements

Desiderata

The next factor that should be included is the *desiderata* of the user, proposed by Langer et al. [63]. This concept combines stakeholders' interests, goals, expectations, needs and requirements for AI systems. Desiderata relate explainability approaches to the satisfaction of each stakeholder. It recognises that explanatory information facilitates stakeholder understanding, which influences satisfaction. This study attempts to understand and satisfy the desiderata of the target user, the NLP data scientist. Langer et al. [63] defined them for the general developer, which is most relevant for the target user in this research. These desideratum are presented in Table 2.3.

When

Next to the persona-related factors, another question that should be investigated is *when is XAI needed?* The development phase of any AI system can be defined in multiple phases or stages, and all have their contributors and corresponding goals and objectives. The phases are described by Suresh et al. [117] and include Development, Deployment, Immediate Usage & Downstream impact. Downstream impact is defined as the impact of the prediction on other system components.
Stages are also defined by Hong, Hullman, and Bertini [47], stating three stages. First, the *Ideation & Conceptualisation* stage; second, the *Building and Validation* stage; and third, the *Deployment, maintenance, and Usage* stage. It is important to recognise these phases for both the XAI needs and the workflow of the NLP data Scientist.

Table 2.3: The desideratum for developers according to Langer et al. [63]

Desideratum	Description
Accuracy	Assess and increase a system's predictive accuracy
Debugability	Identify and fix errors and bugs
Effectiveness	Assess and increase a system's effectiveness; work effectively with a system
Efficiency	Assess and increase a system's efficiency; work efficiently with a system
Performance	Assess and increase the performance of a system
Robustness	Assess and increase a system's robustness (e.g., against adversarial manipulation)
Security	Assess and increase a system's security
Transferability	Make a system's learned model transferable to other contexts
Verification	Be able to evaluate whether the system does what it is supposed to do

Knowledge level

Another factor to consider is the level of expertise of the user. During the development of a tool, discussing the level and type of background knowledge required to comprehend an explanation is crucial [112]. In this research, it is assumed that NLP Data Scientists have a minimal understanding of ML and NLP. However, domain knowledge regarding the solutions they are building might not always exist.

How

Lastly, the factor that needs to be considered in this study is the question of *how does the user want to have the explanations*. This is investigated by discussing the workflow of the user and investigating where in this workflow the best point is to incorporate the explanations.

2.3.2. HCXAI Principles

Previous studies have defined several principles that any HCXAI system how adhere to. One major contribution is from Chromik and Butz [19], which defines four important principles

1. **Complementary Naturalness:** Consider complementing implicit explanations with rationales in natural language.
2. **Responsiveness Through Progressive Disclosure:** Consider offering hierarchical or iterative functionalities that allow follow-ups on initial explanations.
3. **Flexibility Through Multiple Ways to Explain:** Consider offering multiple explanation methods and modalities to enable explainees to triangulate insights.
4. **Sensitivity to the Mind and Context:** Consider offering functionalities to adjust explanations to explainees' mental models and contexts

Other principles are proposed by Cirqueira, Helfert, and Bezbradica [22], which sets out five different principles:

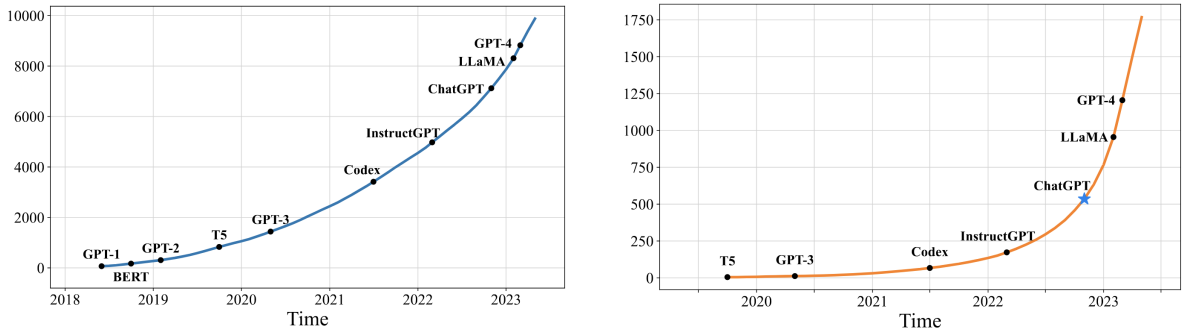
1. The explanation should provide prediction probabilities in order to observe the confidence and limitations of the system.
2. The explanation should provide cases of similar and dissimilar predictions to understand the prediction.
3. The explanation should give a level of importance to data instances to understand the global behaviour of the model.
4. The explanation method should be given in human-readable rules and understandable visualisations.
5. The explanation method should provide the influence of features to grasp the notable attributes within a local prediction quickly.

2.4. Large Language Models

This study focuses on the use of Large Language Models (LLMs) to create solutions to NLP tasks. This section explores the themes associated with these models. This is done by giving a brief background on the model architecture, followed by related topics such as fine-tuning and prompting. The applications are then discussed, followed by the challenges and problems of these LLMs.

2.4.1. History

Language modelling as a way of building NLP applications has been around for a while. In the early days, it was done using statistical models followed by neural language models. Here, the field started using word embeddings and bidirectional LSTM architectures [74]. However, the revolution really began in 2017 with the emergence of the transformer model [123]. This transformer model, which uses a multi-headed self-attention mechanism, was most revolutionary in terms of scalability. It made it possible to create huge models (ranging from 1.5B parameters to 1.2T parameters [136, 53]) and thus increase the performance of the model. From that moment on, it gained more and more traction in the scientific community, as can be seen in Figure 2.2.



(a) Query="Language Model"

(b) Query="Large Language Model"

Figure 2.2: Amount of publications that match the query in the title or abstract, created by Zhao et al. [136]

The traditional model architecture was an encoder-decoder architecture consisting of two stacks; a stack of encoder transformer layers and a stack of decoder transformer layers. For both stacks, each layer contains a multi-head attention mechanism and a feed-forward neural network [74]. The attention heads and feed-forward networks are connected with a normalisation step. A visual representation of a transformer is presented in Figure 2.3. The latest models use so-called decoder-only architectures. Unlike the encoder-decoder architecture, the decoder-only architecture focuses solely on the decoding process [74]. This is done by sequentially generating new tokens using the previous tokens in the sequence to generate the output.

2.4.2. Fine-tuning

Once a model has been pre-trained on a large corpus, it can be fine-tuned. The core concept is to tune the model in a supervised way that improves its performance on a specific task. There are different ways of doing this, such as alignment learning, instruction learning and parameter-efficient tuning. Alignment learning tries to address the safety of the model by tuning it to be helpful, honest and harmless [74].
Instruction learning tunes the model to understand instructions and to respond effectively to user requests. By giving the model an instruction-required-output pair, the model learns how to respond effectively to the prompt and input [93].
Alignment and instruction learning are complete tuning methods because they change all the parameters in the model. In contrast, parameter-efficient tuning uses only a subset of the parameters, leaving all other parameters fixed. This significantly reduces computational and storage costs while maintaining comparable performance improvements [74].

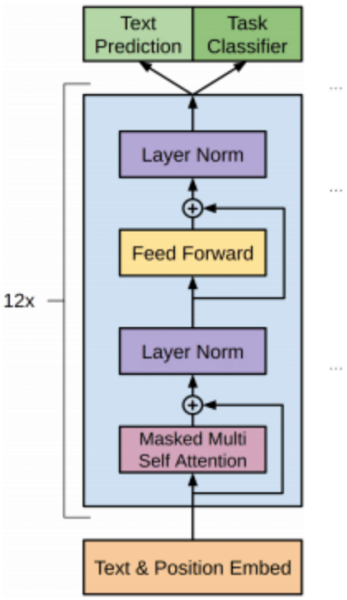


Figure 2.3: Decoder Transformer Architecture [98]

2.4.3. Prompting

Once the model has been fine-tuned, it can generate new answers by issuing a query to the model, called a prompt [93]. One important method here is zero-shot learning, where the model is able to generate an answer to queries that it has not been trained on or added to the prompt. Few-shot learning, also called In-Context Learning (ICL), is a method where multiple input-output examples are given in the prompt to show the model the desired output (format). The last popular method is reasoning, where the model is asked to generate answers to a logical problem by reasoning. An example of this is *Chain-of-Thought (CoT)*, where the model is asked to give its reasoning step by step. These methods can significantly influence the performance of the model if used appropriately [93].
It is worth noting that the explanations can also help in the prompting paradigm. For example, these models have emergent properties where they are able to perform well on tasks on which they have not been trained. Investigating how these emerging abilities arise and what is needed can also be done with explainability [135]

2.4.4. Evaluation

There are several ways in which the output of the LLM can be evaluated. A comprehensive overview is provided by Tikhonov and Yamshchikov [121], but this thesis only provides an overview. There are specific categories in which an LLM can be evaluated. For example, it could be text-specific, such as word order or tokenisation. It could also be skill-specific, such as writing, reasoning, mathematics or coding. Finally, it could be based on output attributes (also called personality traits) such as consistency, readability and correctness. Tikhonov and Yamshchikov [121] notes that all current evaluation approaches are not very effective and do not meet modern requirements, as they lack a precise and formal definition.

2.4.5. Applications

Now that the relevant concepts have been described, it is useful to understand the applications and challenges of these Large Language Models (LLMs). The introduction of generative models introduced a whole new range of applications in various sectors. These are applications like chatbots, assisting with programming, writing creative work and more [53].
In addition to this whole new field, these models also revolutionised the classical NLP tasks. This could

be tasks like sentiment analysis, text classification, fact checking and machine translation, which can now be solved with LLMs [100]. Concrete examples are data extraction, improving data quality by reference checking multiple sources or classifying incoming emails.

2.4.6. Challenges

There are many different challenges associated with LLMs. These include issues such as cost, security, technical challenges and more. The most relevant to the target user are described here.

Lack of Reproducibility When dealing with closed source models; lack of reproducibility of inference due to stochastic API in a black box environment; don't know when changes are made, which model versions are maintained and stochastic outputs even at low temperature [53].

Hallucinations Another challenge is that LLMs suffer from *hallucinations*. This means that it contains inaccurate information that is difficult to detect due to the fluency of the text. This can be partly solved by the technique of retrieval augmentation, where stored information is retrieved based on the prompt and added to the model's input. This is not always possible and, unfortunately, does not solve all hallucinations [53].

Prompt Brittleness It has been found that the syntax of the prompt, such as length, words and instructions, has a significant impact on the performance of the models. It has also been shown that the order of examples within few-shot learning has a significant effect on performance, with some permutations performing close to state-of-the-art and others performing virtually random guesses [75]. Therefore, creating the best possible prompt can be a challenging task.

Fairness Several types of bias can occur in the output of the model, resulting in unfair models. These include social bias, where the model stereotypes, excludes social groups or misrepresents [34]. For example, in a case of stereotype bias, "*He* is a doctor" is much more likely to be generated than "*She* is a doctor" [68]. One study suggests that about 15% to 30% of attention is associated with stereotypes [132]. These biases can arise from the data or during training and can be categorised as intrinsic and extrinsic biases. Intrinsic means that the bias is encoded in the embeddings, whereas extrinsic bias corresponds to the decision bias of the downstream tasks [67].

Privacy Privacy is a major concern. Recent studies have shown that models can leak training data during generation. This means that sensitive data can be exposed and used for harmful purposes [132]. There are two main approaches to improve privacy: 1) prevent the model from storing sensitive data and 2) ensure that it does not leak sensitive information during generation. In addition, to address this challenge, explanatory techniques could be used to confirm whether LLMs have internalised certain knowledge.

2.5. Large Language Model Explainability techniques

This section will give an overview of the different explainability techniques that are currently available for Large Language Models (LLMs). This is done by first exploring the local explanation techniques, followed by global explanations.

2.5.1. Local Explanations

Local explanations for LLMs are often categorised into four types: feature attribution analysis, analysis of the individual components of the transformer like the attention mechanisms, example-based explanation and natural language explanation [135]. The overview is also visualised in Figure 2.4. These will be discussed below.

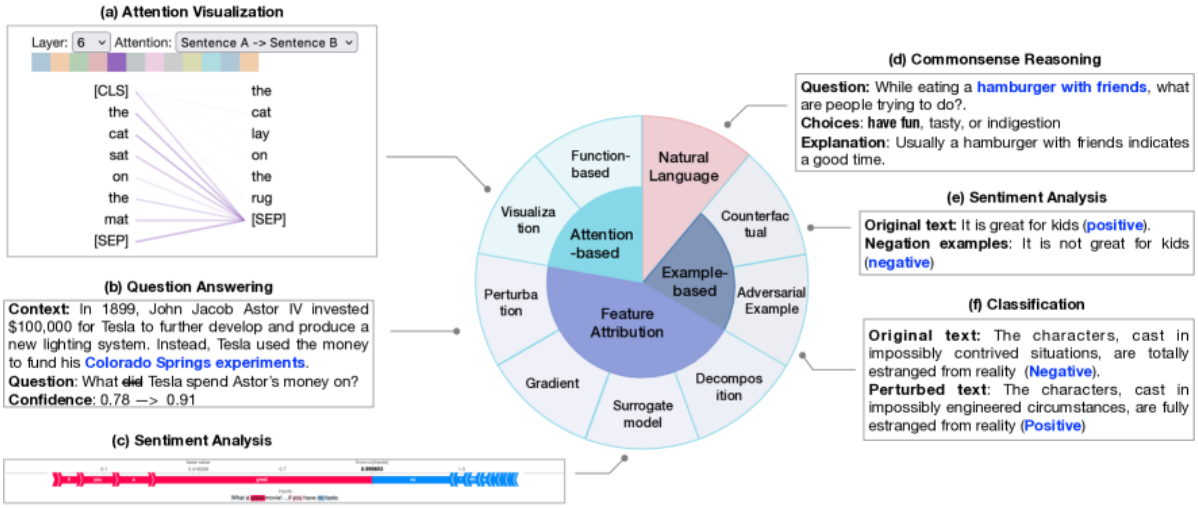


Figure 2.4: Overview of local XAI techniques, visualised by Zhao et al. [135]

Feature Attribution Analysis

Feature attribution focuses on the relevance of each input feature to the model's prediction. While several computation methods exist, they all aim to highlight which tokens had a positive or negative impact on the output. The three computational methods are perturbation-based, gradient-based, and decomposition-based, and they will now be explained.

Perturbation-based Perturbation-based methods perturb the input by removing, masking or altering input features and evaluating how this changes the output. Examples are *leave-one-out* [66], Input Reduction and HotFlip [29]. An additional advantage is that they can measure the robustness of the model [33].
This category includes methods such as LIME [103] and SHAP [76]. These methods also change the input and see how this affects the output. However, they are not inherently usable for LLMs and adaptations are needed. TransSHAP is an example that focuses on adapting SHAP to subword text input and providing sequential visualisation explanations that are well suited to understanding how LLMs make predictions. Note that this is adapted for BERT (encoder-only) models [58].
In addition, perturbation faces some challenges in terms of efficiency and reliability [77]. In reliability due to model overconfidence [33]. For example, models can maintain high confidence predictions even when the reduced inputs are nonsensical. However, this can be solved with regular examples, label smoothing, and fine-tuning of model confidence [33].

Gradient based Gradient-based feature attribution determines the importance of each feature using partial derivatives with respect to each input dimension. The magnitude of this derivative will reflect the sensitivity of the output to changes in the input [135]. Examples are vanilla gradients [125] or integrated

gradients [116]. One challenge with Integrated Gradients (IG) remains the computational overhead required to achieve high-quality integrals [110].

Decomposition-based Decomposition aims to decompose the relevance score into linear contributions from the input. This could be done by assigning relevance scores directly from the output layer to the input [28]. Or it can be done by aggregating relevance scores layer by layer, as is done in layer-wise relevance propagation [87]. These can be used to decompose relevance scores into contributions from different model components, and have also been adopted to work with transformer models [17].

Analysis of Transformer components

In addition to the feature attributions, it is also possible to explain the output by utilising the internal mechanisms of the model. As explained in section 2.4, each transformer block contains a multi-head self-attention sublayer, followed by an Multilayer Perceptron (MLP) sublayer [123]. These sublayers can be used individually to provide information about the prediction [74].

Attention-based The attention-based explanation makes use of the multi-headed attention heads of the LLM. Intuitively, it captures meaningful correlations between intermediate levels that explain the models' prediction [135]. This can be visualised *raw*, where a heatmap can be used to visualise the weight for each layer. However, *raw* is not enough to fully explain the prediction. To solve this, functions such as Grad-SAM [10] or integrated versions of partial gradients [41] can be used. Grad-SAM works by analysing self-attention units and using the attention matrices together with their gradients to produce a ranking over the tokens. This identifies the input elements that best explain the model's prediction. This works better than visualising attention alone.

MLP-based The Multilayer Perceptron (MLP) layer, which is a layer of fully connected neurons with a nonlinear activation function [123], can also be used to provide information about the prediction. This can be done by viewing the token representation as a changing distribution over the vocabulary and the output of each Feed Forward Network layer as an additive update to that distribution. Each update can then be decomposed into sub-updates corresponding to a single vector that can be transformed back into tokens that are interpretable by humans [35].
It should be noted that there is a big debate about whether attention maps can be used for explanation, as they may not be faithful. Some argue that raw attention does not identify the most important features for prediction [108, 50] or during code generation [60]. Others say they do not contribute as much to prediction as assumed [84]. Some technical solutions have been explored but have not settled the debate [136, 12].

Example based

The last local explanation category is Example-based. These explanations illustrate how the output of the model changes with different inputs. This can be done in a number of ways, including adversarial and counterfactual. Adversarial examples are based on changing the less critical components of the input data to show how this changes the output. This is particularly useful for highlighting areas where models fail and where the model can be improved in terms of robustness and accuracy. Counterfactuals are a form of casual explanation where the input is perturbed in such a way that the output should change. This is done in Polyjuice [131], for example, a Python tool that supports multiple permutation types such as deletion, negation and shuffling. This is then used to create realistic counterfactuals [135]. It should be noted, however, that these methods are highly susceptible to hallucination.

2.5.2. Global Explanations

Explanations with a broader scope are called global explanability and aim to understand the LLM as a whole. They are generally more focused on uncovering biases and privacy issues, and therefore more focused on building more trustworthy models. The most studied techniques are probing techniques and mechanistic interpretability. Probing techniques scrutinise model representations, and mechanistic interpretability is a relatively new field that focuses on reverse engineering the inner workings of the MLP.

Probing Probing techniques refer to methods used to understand the knowledge that the LLM has captured and how that knowledge is represented. By probing the model it can discover paths within the model [135, 111]. An example is provided by Clark et al. [23], who demonstrate an attention-based probing technique using a classifier to show that syntactic information is captured within BERT's attention.

Mechanistic interpretability A relatively new approach is called mechanistic interpretability, which examines the neurons of the model. This can be done by categorising or decoding concepts from individual neurons [24, 92]. These can then be grouped together to further understand how different neurons together perform a specific task, which is called circuit discovery [128]. While these methods are difficult to scale to today's largest LLMs, a very recent publication was able to scale them to the Claude 3 LLM [119].

2.5.3. Challenges in Explainability for LLMs

Explaining the behaviour of LLMs poses several challenges. This is due to a number of factors, which have been discussed by Liao and Vaughan [70]:

Complex and uncertain model capabilities and behaviour There is a very wide range of different tasks that an LLM can perform well because of its immense flexibility. However, these emerging capabilities also make it very unpredictable and unreliable. This non-deterministic behaviour makes the response inconsistent and therefore difficult to generalise [70].

Massive & opaque architectures Today's models are massive and complex, making it very difficult to get a full picture of the knowledge reflected in a model or the reasoning used to produce its output. Even when a closer look is taken at the different internal mechanisms, it is very hard to comprehend the full behaviour of the model and can even create misleading explanations as they might be unfaithful [70].

Proprietary Technology A more obvious reason is the inaccessibility of the model parameters. When these models are used, they are often only accessible via API's. As a result, it is impossible to access the inner workings of the model, which is often required to compute most post-hoc explanations. Not only are the parameters not shared, but other details such as size, training data and number of parameters are often hidden, creating even more opaque models [70].

Organisational pressure to move fast The final reason discussed by Liao and Vaughan [70] is the "AI race". As organisations are often under pressure to release products quickly to be the first, responsible AI challenges this fast pace. Companies try to achieve breakthroughs to improve quality, which is incentivised by the market, while transparent AI is not. As organisations are often under pressure to release products quickly in order to be first, responsible AI conflicts with this goal. Companies try to achieve breakthroughs to improve quality, which is incentivised by the market, whereas transparent AI is not.

2.6. Related work

This section explores and examines the implemented tools that use explainability techniques for LLMs. By reviewing existing implementations, a better understanding of the current state will be gained. A brief overview of the tools is given, followed by some initial findings. First, a set of criteria is presented to focus attention on only those tools that are relevant for the current scope.

2.6.1. Selection Criteria for Existing Explainability Tools

In order to narrow down the scope of all tools, several criteria have been established. These will now be presented, together with the reasoning behind them.

- **Uses at least one XAI technique**. Since the focus of this tool is on explainability, it should use at least one technique that gives insight into the behaviour of the model.
- **Focus on decoder-only models**. Given the differences in explainability techniques between different model architectures, the scope is still large if all transformer architectures are considered. Therefore, it was decided to include only auto-regressive models, which are currently becoming increasingly dominant in the NLP field and are relatively unexplored in terms of their internal behaviour. In addition, fewer tools were found that focus specifically on this family of models.
- **Compatibility with Open Source Models**. Most explainability techniques use internal mechanisms to compute an explanation, either via the attention mechanism or the tightly coupled feed-forward layer, so the tool should be able to access these weights.
- **Open Source Implementation**. In order to select only relevant tools, it was decided to consider only open source implementations, as the implementation behind the tool is a relevant factor for this section.

Together, these criteria will give a list of tools that are most relevant to this study and can also inspire the development of a new tool.

2.6.2. Results

A wide range of implementations have been found that present an explainability technique for LLMs. The result is presented in Table 2.4, together with additional information on the tool. For each tool, the table describes if the tool is actively being developed or maintained, what technique is implemented and if the explainability technique is evaluated using at least one evaluation metric. Lastly, it has been investigated for each tool if they are user-evaluated using HCAI & HCXAI principles.

2.6.3. Design Frameworks

Contributions have been made to the design of a usable architecture. Two important contributions are from Mosca et al. [88] and Lee et al. [65]. A pattern noted in these two designs is the use of an external frontend where the visualisations are presented, and the user can interact with them. The Interaction Framework for Artificial and Natural Intelligence (IFAN), proposed by Mosca et al. [88], is presented in Figure 2.5. While this is originally created to satisfy the needs of multiple users, it is a system architecture that will be used as inspiration during this study.

Table 2.4: List of other XAI implementations

Name	Year	Active dev	Techniques	XAI evaluated	HCI evaluated
LLMCheckup [129]	2024	No	Feature Attribution Free-text rationalization Semantic Similarity Counterfactual Generation	Fluency Consistency	No
InterroLang [32]	2023	No	Feature Attribution Counterfactuals Rationalization	Correctness helpfulness satisfaction	Yes
Inseq [105]	2023	Yes	Feature Attribution	No	No
Ferret [9]	2023	No	Feature Attribution Shap Lime	Faithfulness Plausibility	No
LM-Debugger [36]	2022	No	FFN updates	No	No
PolyJuice [131]	2021	No	Counterfactuals	No	Yes
Ecco [6]	2021	No	Feature attribution Neuron Activation	No	No
Transformer Lens [91]	2021	Yes	Neuron Activations	No	No
Transformers-Interpret [99]	2021	No	Feature Attribution	No	No
LIT [120, 8]	2020	Yes	Feature Attribution Attention maps Counterfactuals	No	Yes
Captum [59, 82]	2020	Yes	Feature Attribution	Robustness	No
ExBert [48]	2020	No	Attention maps Contextual representation	No	No

Year defines the year it was published, *Active dev* assesses if the github is still in active development. *Techniques* describes the implemented explanation techniques and XAI evaluation describes whether the technique was evaluated and more specifically if it was evaluated with humans

2.6.4. Insights

Given the overview given in Table 2.4, several observations can be made. First, it can be noted that all implementations were primarily developed for research purposes. Their focus was on presenting one or multiple novel explainability techniques rather than building an industry-ready tool. Consequently, it is reasonable that these were not evaluated using human-grounded evaluation metrics and user studies. There are some exceptions to this observation, including Polyjuice [131] and LIT [120]

Next, most use visualisation techniques over natural language. The visualisation tools commonly used saliency maps of the tokens, either by highlighting tokens within the text or as a matrix where the input is positioned on the Y-axis and the output on the X-axis. Furthermore, attention mechanisms are often visualised using two columns of text and lines connecting the words to highlight the attention between the connected words.

However, when looking at rationalisation, which is the technique of giving an explanation in text, the most notable is InterroLang, by Feldhus et al. [32]. A comprehensive overview of other techniques can be found in the survey by Gurrapu et al. [40].

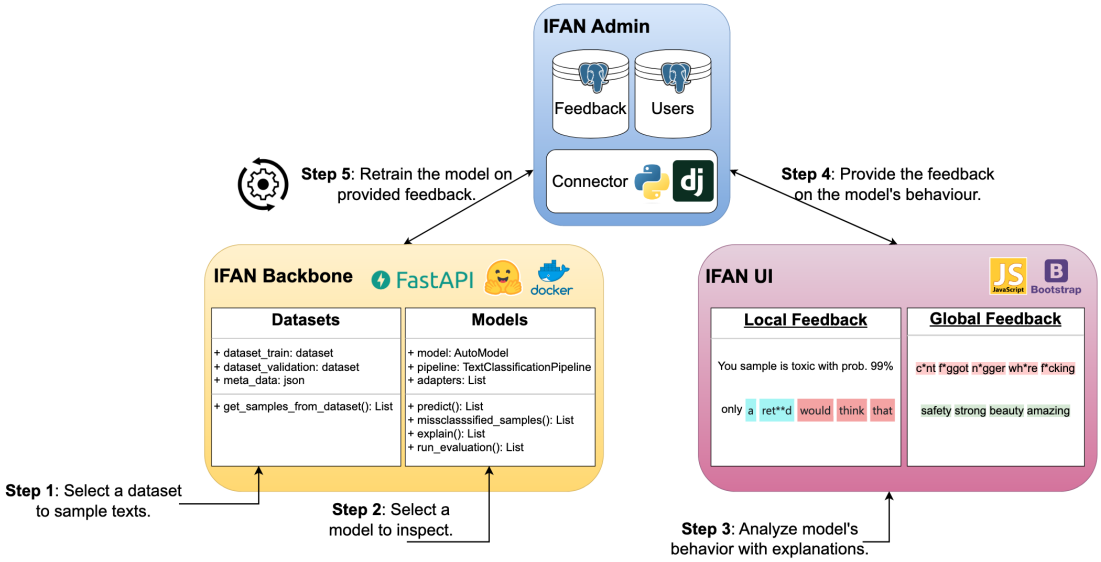


Figure 2.5: Proposed architecture from IFAN [88]

3
Formative Study

This section describes the first phase of the research, a formative study. Before this could be done, the target user of the tool needed to be decided. The target user was selected based on conversations with employees within Elsevier. In total, 14 conversations were conducted to determine the final target user for interviews and tools. The employees held positions such as data scientist, responsible AI expert, data engineer, head of data science or manager. The conversations revealed that different roles within the company have different definitions, knowledge levels, and perspectives on XAI. It also provided a better overview of a suitable user for the tool. All these interviews were taken into account when selecting the most appropriate target user.

Target user of this thesis: NLP Data Scientist

It was concluded that the NLP Data Scientist was the most appropriate target user. The NLP Data Scientist is defined as a general data scientist with domain specific NLP knowledge and projects. This decision was influenced by the fact that they were actively building LLM-based products and recognised the benefits of a human-based explainability tool, thus indicating its potential.

The formative study followed, by interviewing the the NLP Data Scientist. This formative study provided insights into the workflow and user needs required to gather the right requirements and implement the human-centred tool. The outcome of this phase is a set of observations, which follows a similar approach to other contributions, including a study by Gu et al. [39]. The method is described first, followed by the results.

3.1. Method

A formative study was chosen to further deepen the knowledge of the NLP Data Scientist. This was done by interviewing participants and discussing topics related to LLMs and explainability. The interviews with the participants were divided into three parts: 1) What are the current challenges and desiderata of the NLP Data Scientist, 2) What is the current workflow when developing an LLM-based product, and 3) How do they use XAI? The questions can be found in Appendix A. It was chosen to not only talk about explainability but also to gain knowledge about the general needs and challenges of the target user. This ensured that no challenges were overlooked when selecting and building the explainability tool. Each interview lasted approximately 45 minutes and was recorded in audio and video format. The audio was automatically transcribed and then edited for clarity and accuracy to improve the transcript processing.

Participants

Six NLP data scientists participated in the interviews using a self-selected convenience sample. They were selected from three teams; all had at least two years of experience in data science and had completed at least one project in the NLP domain. The interviews were voluntary, with their informed consent and their manager's approval. The participants are given in Table 3.1

Table 3.1: Overview of the participants

Participant	Position
Participant 1	Senior Data Scientist
Participant 2	Principal Data Scientist
Participant 3	Data Scientist
Participant 4	Senior Data Scientist
Participant 5	Manager Data Science
Participant 6	Data Scientist

Thematic Analysis

Based on the interview transcriptions, several steps were taken to distil the knowledge. These are based on the steps described by Naeem et al. [90]. These steps include the creation of *codes* and their collation with supporting data. The codes can then be grouped into *themes*. These themes and codes can be revised if necessary. How the codes and themes were created is described in more detail below.

Coding Strategy

There are several ways to code transcripts for data analysis. A clear distinction is made between inductive and deductive coding [13]. Inductive coding looks for patterns in the data to guide coding and theme development. Deductive coding, on the other hand, examines data based on preconceived frameworks and ideas. It is therefore, less flexible and more focused on what is already known to the researcher.

A hybrid of the two approaches can also be used, as suggested by Naeem et al. [90]. We take this approach because this research is based on literature, which is presented in chapter 2. This helps to create the initial codes; however, certain patterns and considerations could not be included in any codes. Therefore, these codes were created inductively by analysing the patterns between the transcripts. This hybrid method allowed for a flexible set of codes that provided a well-defined basis for the requirements of the tool.

Codes

In total, 312 quotations were highlighted, which created a total of 124 codes. The codes combine pre-defined codes and newly found codes during the analysis. A sample of the codes created can be found in Appendix B

Themes

Once the codes had been analysed, it was possible to create themes. A theme should represent a patterned meaning within the data that informs the research questions [90]. These themes can be created by grouping and categorising the different codes. This categorisation is also based on the ideas and aspects presented in the background of this research. Sub-themes have been introduced because each theme is still a broad range of topics. The defined theme and sub-themes are presented in Table 3.2

Table 3.2: Identified themes and sub-themes

Themes	Sub-themes
Persona	Characteristics Goals Desiderata
Challenges	Persona Challenges Technical Challenges
LLMs development in practice	Utilisation Factors on model
Perspectives on XAI	Current view XAI Potential benefits of XAI Usage in practice
Workflow	Steps Evaluation Strategies Business practices Technical Stack Workflow desires

Distilling Observations supplemented by literature

Processing the themes allowed for the distillation of observations, which helped to create this tool in a human-based way. In this research, observations are defined as conclusions drawn from the themes with additional literature for comparison. This comparative approach provided more knowledge on the current status and how it differs from the findings of this research. It should be noted that the observations are high level, i.e. they provide direction. However, they do not provide insight into how the observations can be solved in practice.

3.2. Results

This section presents the results of the formative study. First, an overview of all observations is given, which can be found in Table 3.3. Here, the sub-themes, and observation conclusions are given, providing an overview of the results that will be discussed. This table is followed by a section for each of the themes, together with the relevant literature.

Table 3.3: The themes and the list of observations for each sub-theme

Theme	Sub-theme	Observation
Persona	Characteristics	OB 1.1: NLP Data Scientists are versatile
		OB 1.2: Responsible for working with multiple stakeholders
	Goals	OB 1.3: Model Visualisation & inspection is less important
		OB 1.4: Model tuning & selection keeps being important
		OB 1.5: Building Proof-of-Concepts is an important goal
	Desiderata	OB 1.6: Desiderata consistent with the general data scientist
		OB 1.7: Trust is not a desideratum
Challenges	Persona	OB 2.1: Staying up to date is difficult
		OB 2.2: Performing an actionable error analysis is challenging
		OB 2.3: Creating reproducible results is challenging
		OB 2.4: Prompting is difficult
	Technical	OB 2.5: Setting up models is difficult
		OB 2.6: Comparing different models is challenging
LLM Utilisation	Utilisation	OB 3.1: Compared to pre-LLM era
		OB 3.2: Generative use cases
		OB 3.3: Closed and open-source models are used
	Factors	OB 3.4: Cost vs Performance is a trade-off
		OB 3.5: Usability is an important factor
		OB 3.6: Controllability is an important factor
		OB 3.7: Format alignment is important
Perspectives on XAI	View	OB 4.1: Different Definitions exist
		OB 4.2: Differences between classical ML & LLMs
	Potential benefits	OB 4.3: Understanding model behaviour
		OB 4.4: Better prompt engineering
		OB 4.5: Error Analysis
		OB 4.6: Improving Safety & Trust
		OB 4.7: Quality Assessment
	Usage in practice	OB 4.8: Unawareness as a reason for lack of adaptation
		OB 4.9: Perceived usefulness as a reason for lack of adaptation
		OB 4.10: Fitting metrics as a reason for lack of adaptation
Workflow	Steps	OB 5.1: Individual iterative steps
	Evaluation Strategies	OB 5.2: Performance Metrics
		OB 5.3: Subject Experts
	Business practices	OB 5.4: Company practices
		OB 5.5: Industry Metrics
	Technical Stack	OB 5.6: Used Tools
	Workflow desires	OB 5.7: Tracking experimental results
		OB 5.8: Verifying hypothesis
		OB 5.9: Easy collaboration

3.2.1. Persona

The first topic concerns the characteristics of the target user. These observations will help create a tool that supports this user's specific needs. In order to digest further findings, several observations will be compared with the literature, as discussed in the method section of this chapter.
The following sub-themes are discussed: the characteristics of the NLP data scientist, the goals and needs of this user, and the questions the user has during his work.

Characteristics & Responsibilities

The first sub-theme concerns the characteristics of the NLP Data Scientist and the responsibilities of the role.

Literature The literature survey on this topic revealed a significant gap since little research was found on the characteristics and responsibilities of the NLP Data Scientist. However, in the general Data Scientist, several contributions have been found. For example, Kim et al. [56] investigated into the roles and types of Data Scientists within large tech companies. They found four topics involving a Data Scientist: user engagement, software productivity and quality, domain-specific problems (such as NLP), and business intelligence. Kim et al. [56] identified the responsibilities of a Data Scientist and observed a significant difference across teams and domains. The roles included querying the data, preparing the data, analysing the data and digesting insights. It sometimes included building a data platform or machine learning solution.
This research focuses on domain-specific problems within Natural Language Processing (NLP), which creates a new set of characteristics and responsibilities. When considering the roles defined by Tomsett et al. [122], NLP data Scientists can be described as the 'operator' of the project, as it often interacts with the input & output. Additionally, they can be seen as the 'creators' when they are fine-tuning the models and processing the outputs.

Observations

Now that we have an understanding of the Data Scientist, we can compare the interview results to it.

Observation 1.1: NLP Data Scientists are versatile Compared to the traditional Data Scientist, it can be concluded that NLP Data Scientists have a similar range of responsibilities. Participant 6 acknowledged that their role is quite versatile and can include many aspects of a project. They might include being the data engineer, as mentioned by P4, by setting up the data processing pipeline to ingest the data, even when their primary responsibility is doing a text classification task. Another task included in this role is setting up the infrastructure necessary to fine-tune a model or set up a deployment environment, as mentioned by P6.

Observation 1.2: Responsible for working with multiple stakeholders Various stakeholders are involved in an NLP project. Understanding the business requirements and creating a solution that fits the goals of the stakeholders is a significant part of their responsibility, as participants 2 and 4 mention. However, it could also work the other way around, where the stakeholder is solving a part of the problem of the Data Scientist. Business requirements often come from clients, who can be from inside or outside the company and are very important to the NLP Data Scientist. The client will define the problem, the task, and the business value and might even have a say in the performance metrics. It is difficult for the data scientist to narrow the scope so that a Proof of Concept (PoC) can be built, which also satisfies the stakeholders.

Goals

The second sub-theme in this theme is the goals of the NLP Data Scientist. As described in section 2.3, it is important to understand the persona's goals before building the Explainability Tool. First, literature is presented to create the correct background knowledge to contextualise the observations.

Literature One scientific contribution investigating the goals of a Data Scientist is by Mohseni, Zarei, and Ragan [85]. Two main goals were identified: 1) model visualisation & inspection and 2) model tuning and selection. In this context, this would be visualising the attention heads, for example. These will be compared to the insights from the formative interview.

Observations

Observation 1.3: Model Visualisation & inspection is less important Model visualisation and inspection were not mentioned during the interviews. This is also more challenging when dealing with LLMs, as they are large, complex and challenging to visualise or inspect. While methods exist for visualising attention heads, for example, these seem not useful for them. Additionally, the inspection might be complex when using closed-source models, as no internal mechanisms can be inspected.

Observation 1.4: Model tuning & selection keeps being important In contrast, model tuning and selection continue to be relevant. LLMs are often compared, primarily when a new model is published. NLP Data Scientists frequently perform model tweaks, such as fine-tuning, to optimise model performance for specific tasks or datasets. While the context and application differ, they are similar to how a traditional data scientist operates.

Observation 1.5: Building Proof-of-Concepts is an important goal Another observation from the interviews was the goal of building proof of concepts. This was often mentioned as a first step to get to the first version of the product quickly. The aim of the PoC was to get feedback quickly and see where improvements could be made. This was often done with a small amount of data and a smaller version of the model.

Desiderata

As outlined in chapter 2, desiderata for XAI are the expectations, needs and demands combined. This sub-theme revisits literature, followed by the observations extracted from the interviews.

Literature Langer et al. [63] describes the desiderata for XAI of the developer. These are the following nine: Accuracy (1), Debugability (2), Effectiveness (3), Efficiency (4), Performance (5), Robustness (6), Security (7), Transferability (8), Verification (9). These can now be compared to the findings of the interviews.

Observations

Observation 1.6: Desiderata are consistent with the general data scientist First, all desiderata remain valid for the NLP Data Scientist, though their importance may differ. In this section, all desiderata will be placed into the context of the NLP Data Scientist. Additionally, each desideratum will be assessed based on its importance.

In the context of model capabilities, improving accuracy (1) and assessing performance (5) remain important to the NLP data scientist. 50% of respondents emphasised improving the accuracy and performance of several aspects of the solution.This could be prompt engineering to evaluate which model is best for the specific use case (P1).

Debuggability, while still important, appears to be less so given the focus on refining model performance using pre-configured libraries and APIs. These pre-configured libraries from the model owner (e.g. OpenAI) are preferred, as mentioned by Participant 1, who notes that tools like Langchain are helpful but difficult to debug when an error occurs. Langchain will be further discussed in the workflow theme, and the challenges will be discussed in more detail in subsection 3.2.2.

Effectiveness (3) is defined in this context as the degree to which they achieve their intended goal. This is a valid desideratum of the revised persona. Similarly, several respondents highlighted effectiveness (5) - finding solutions quickly and evaluating results - as still a primary goal for NLP data scientists.

Robustness (6) is not mentioned in the interviews and may be less relevant to them. This may be true when building a model that interacts directly with users. However, when it is used to improve data quality or to perform non-generative tasks, it may not be the most relevant. More research should be done to further confirm or falsify this desideratum.

Security (7) is crucial for them, especially in terms of data security and privacy. When using closed-source models, the data is often stored at the hosting company. During the interviews, the use of the model focused on the company's own data, which should not be stored elsewhere, as Participant 1 mentioned.

A possible solution is for the hosting company to offer a private version of the model hosting, as OpenAI does, without storing the company's data.

Transferability (8) in the current context is about whether the LLM can be transferred to other contexts. This desideratum seems less critical because the solution is often tuned for one task at a time.
Finally, verification is defined as checking that the system does what it is supposed to do. This can be seen as evaluation, which is an important part of the persona's role.

Observation 1.7: Trust is not a desideratum Within the list presented by Langer et al. [63], trust is not mentioned as a concern for the developer. This is in line with the NLP Data Scientist, who is often focused on performance and less on trust. This was discussed in the interviews. To quote P5: *"Trust is not a big concern for me: as long as they [OpenAI] do not use our data to train new models on"*. This quote refers to trust in the companies behind closed-source models and is consistent with the opinions of other participants. The contract with OpenAI, which states that the company will not use their data, creates enough trust in the use of closed-source models. In addition, the low stakes of the projects may be a reason why trust is not an issue.

3.2.2. Challenges

To ensure we solve the right problem, we need to understand the user's challenges. This is split into two sub-themes: the challenges of an NLP Data Scientist and the technical challenges.

Challenges faced by NLP Data Scientist

First, the challenges related to the persona are presented. The relevant literature is presented first, followed by the observations.

Literature Previous research has explored the challenges faced by the General Data Scientist. For example, by Kim et al. [56], which identifies six problems faced by the general data scientist:

- Poor data quality
- Data availability (missing values, delayed or incomplete data)
- Data preparation (putting it all together)
- Scale (it can take a long time to run the batch jobs)
- Machine learning related tasks (e.g. attribute mapping)
- Getting stakeholders on the same page

A number of other studies have been carried out that highlight the need for reproducibility in data science projects as a challenge. It has been shown that data, packages, documentation and intermediate results need to be preserved to address this reproducibility issue [80].
In addition, coordination, collaboration & communication are challenges faced by data scientists. Martinez, Viles, and G. Olaizola [80] divides them into:

1. Team management (collaboration, lack of transparent communication)
2. Project management (timelines, uncertain business goals)
3. Data & Information Management (lack of reproducibility, poor quality data, accumulation of knowledge)

Observations

Now that the literature has been briefly explored, the observations from the interviews are presented and compared with the literature where appropriate.

Observation 2.1: Staying up to date is difficult A challenge that was often mentioned was staying up with the latest technology. The field of research on LLMs is developing rapidly, as P2 mentions, making it difficult to be fully informed. This was also mentioned as a reason for not using XAI, as newer XAI methods are often left aside despite their potential benefits.

Observation 2.2: Performing an actionable error analysis is challenging In addition, several participants acknowledged that error analysis is a difficult workflow phase and often consumes a lot of time. This challenge can be attributed to many factors, including the observation that closed-source models are often used, which do not provide additional information about how the model arrived at its output. In addition, scoring metrics for NLP tasks require a lot of work to define. As described earlier, this could be solved by asking the Subject Matter Expert (SME). While the SMEs can provide insights, it is not a mathematically defined metric. This also makes it harder to move from erroneous outputs to actionable insights on what needs to be changed. Diving into where the model went wrong is challenging and often requires domain knowledge of the specific use case.

Observation 2.3: Creating reproducible results is challenging The challenge of reproducibility was also confirmed during the formative study. Several factors make reproducibility difficult. One reason is the structure of a (Jupyter) notebook, which is not a linear document. The non-linearity can make it difficult to trace the sequences of code execution, which is not a challenge associated with the NLP data scientist. Secondly, the randomness in the behaviour of the LLM during the experiment creates additional difficulties in terms of reproducibility. Another reason is the random nature of these models. The exact different prompts can give different outputs, making it very difficult to reproduce results. Adjusting the temperature of the model is one way to minimise this.

Observation 2.4: Prompting is difficult Only NLP data scientists face the particular challenge of prompting. P2 notes how difficult it can be to create the best possible prompt. P4 confirms this by pointing out how small changes in the input can produce very different outputs. This sensitivity makes prompt engineering a particularly challenging task. But it can also be challenging because of how specific the prompt needs to be.

Technical Challenges

During the interviews, several technical difficulties specific to the NLP Data Scientist were mentioned that complicate their workflow. These are now described.

Observation 2.5: Setting up models is difficult Setting up open source LLMs is difficult as it requires in-depth knowledge of cloud infrastructure, model architectures, hosting platforms and other prerequisites. P1 states that while several solutions address this, such as the Huggingface Platform [1] , it still takes a considerable amount of time to load and configure the model correctly.

Observation 2.6: Comparing different models is challenging It can be difficult to decide objectively which model is better than the other. There are several reasons for this. Firstly, there are no well-defined metrics for comparing two text-generated outputs and deciding which is better. P2 confirmed this, noting that all outputs were correct, but still needed to decide which model performed better.
Further complicating this task, Participant 6 observed that the behaviour of different versions of the same ChatGPT snapshot was significantly different, making it even more difficult to decide which model and version to use.

3.2.3. LLM utilisation in industry

The third theme identified is the use of LLMs. To deepen our knowledge on how XAI can be used, it is important to understand how LLMs are currently being utilised in industry. This is divided into two sections; 1) what tasks are being solved and 2) what models are being used and what influences this choice.

LLM Utilisation

Observation 3.1: Compared to pre-LLM era There is a clear impact of Large Language Models (LLMs). Before these powerful models existed, some NLP data scientists used BERT models or **LSTM!**s. However, a few years ago, companies started to explore the latest LLMs, as mentioned by P4. In addition, while some BERT models are still in use, P3 mentioned that some of these older models are being replaced by LLMs. As well as replacing models, LLMs are now being used for traditional tasks such as text summarization, entity extraction and classification.

[1] https://huggingface.co/

Observation 3.2: Generative Use Cases In addition to replacing older language models, generative models have introduced several new NLP tasks, such as chatbots, Retrieval-Augmented Generation (RAG) and content generation. Despite these new use cases, most of the usage is still coming from traditional NLP cases. NLP can be used to improve high-quality data, especially in a company that focuses on it.

Observation 3.3: Closed and open-source models are used It can be observed that the most popular models currently available are used by the NLP data scientist. Only the newest and well-tested models are considered, such as Mistral [51], Llama [4], and Phi [1]. On closed source, ChatGPT, GPT-4 [96] & Claude were the most mentioned at the time of these interviews.

Factors for deciding which model to use

As mentioned above, both open-source and closed-source models are used. However, which one is actually used depends on several factors. This sub-section makes some observations about which factors are most important.

Observation 3.4: Cost vs Performance is a trade-off One of the most important trade-offs during the experimentation phase is cost. Experimentation can be costly when using APIs from closed-source models but usually results in better performance. For the time being, LLMs are being used internally to improve data quality.
However, there is a specific point at which it is cheaper to host a large open-source model in-house. Especially when fine-tuning the model, as P5 mentioned: *"If the project is very large, with millions of documents, it might be advantageous to use open source models."*

Observation 3.5: Usability is an important factor Libraries such as those offered by OpenAI and Anthropic make it very easy to set up a model and start experimenting. These libraries provide instant access to their state-of-the-art models, making them extremely fast to use. This is in contrast to open source models, where the user has to build an infrastructure themselves. While there are platforms that make this a minimal effort, it still requires more engineering knowledge than closed-source libraries.

Observation 3.6: Controllability is an important factor A disadvantage pointed out by one of the participants was the opacity of updates and versions of closed-source models. These LLMs are constantly updated, but the changes made are not fully disclosed. P6 mentioned: *"[ChatGPT] is getting lazy over time"* and also noticed a big difference between the performance of different snapshots. So it could be that the performance was good and changed significantly after an update. This is something that cannot be controlled.
In relation to opacity, it has been noted that some closed-source models use pre-processing methods to guide the model and apply post-processing to the output. While this generally produces a better output, it is another part that is opaque and uncontrollable.

Observation 3.7: Format alignment is important One advantage of this post-processing is that the output is formatted correctly. Currently, models like ChatGPT and GPT-4 easily output **JSON!**, which is only sometimes the case for some open-source models.

All the advantages and disadvantages play, to a certain extent, a role in the decision of the NLP Data Scientist. This can be summarised into a table, as is presented in Table 3.4

3.2.4. Perspectives on XAI

To further develop our understanding of NLP data scientists, this topic is related to perspectives on XAI. It is relevant to examine their current perspectives within the field. This section discusses these perspectives in terms of their current view and the potential benefits identified.

Table 3.4: Advantages and disadvantages for closed & open-source LLMs

	Advantages	Disadvantages
Closed Source	Often better performance Easy to setup	No control over updates and changes Limited customisation No ability to audit the performance
Open Source	Have explainability More transparency More customisation	Difficult to set up Knowledge about infrastructure is needed Often more latency

Practitioner's current view of AI

The first sub-theme describes the participant's current view of the explainability of AI. Here, it was found that most participants had different definitions of explainability. Secondly, it was interesting to note a difference in the perspective of explainability in traditional Machine Learning (ML) and LLMs. This sub-theme is based on the formative interviews only.

Observation 4.1: Different definitions exist When participants were asked about explainability, some mentioned opaque model architectures and the inability to edit system prompts. They also noted that closed-source models do not disclose the data on which they were trained. These points are more related to transparency and less to explainability. However, more experienced staff were able to identify them correctly. In this regard, P6 mentioned that *"Experienced data scientists know how these models were built. And as a by-product, they know how they will behave."*

Observation 4.2: Differences between classical ML & LLMs One interesting observation that can be made from this is the significant difference in the way LLMs view XAI as opposed to traditional AI. In the context of LLMs, they seem to accept the opaque nature of these models, whereas in the conventional domain, they found XAI helpful. Participant 2 mentioned that feature importance was used to present the model & results to stakeholders but has never used XAI when using LLMs. SHAP values were also used in the traditional ML domain. The use of XAI correlated with experience, with the more experienced data scientists having used more XAI in the past. However, they never used it in their development of LLM-based solutions.

Potential benefits of having XAI

While most participants indicated that they did not see a direct use for explanations, they did recognise potential benefits when asked about the hypothetical scenario of having explainable LLMs. One participant emphasised that explanations should be implemented appropriately in order to be helpful, which is in line with the focus of this research. This paper first reviews recent research in this area and then discusses the potential benefits that emerged from the interviews.

Literature Previous studies have investigated why an NLP data scientist seeks interpretability. A study by Hohman et al. [45] identified four reasons: 1) data understanding, 2) hypothesis generation, 3) model building/improvement & 4) communication. One additional reason is given by Molnar [86], stating that explanations can help understand the errors, as they often focus on the abnormal.
More specifically for Large Language Models, Sun et al. [115] suggests that data scientists could use explainability to improve their prompts, have a better understanding of which prompts can and cannot produce certain outputs, and how to produce more desirable outputs. The potential benefits have also been recently explored by Wu et al. [132], who identify seven reasons why explainability can help. The seven strategies are

1. **LLM diagnosis using attribution methods:** Feature attribution methods could help detect model errors and serve as indicators of hallucinations.
2. **Enhancement through model component interpretation:** The self-attention and feed-forward layers could be used to better understand different parts of the model.

3. **Debugging with pattern-based explanation:** This is defined as methods that aim to trace responses generated by the LLM back to specific training examples. This could be useful for increasing confidence or debugging errors.
4. **Credibility and alignment:** Several explainability methods extract patterns in the model that eliminate potential weaknesses in the model. This could be used to assess security. Furthermore, privacy could be improved by preventing the model from storing sensitive data. Finally, by looking at the attention heads, it may be possible to uncover biases in the model, for example because some of them inherit stereotypes.
5. **Explainable prompts:** The use of Chain-of-Thought (CoT) prompts can reduce errors in reasoning and provide adjustable interim steps.
6. **Knowledge-based prompting:** The use of RAG can also improve hallucination.
7. **Data Augmentation Training:** When data augmentation is used, explanations can be used to delineate desired model behaviours or to identify existing deficiencies.

For most of the techniques mentioned above, it must be noted that while they may provide an explanation, it can still be very difficult to make that explanation understandable and interpretable to humans. In addition, some of these methods still lack high fidelity, making them misleading at times [132].
The potential benefits are also discussed in the LLM Explainability survey by Luo and Specia [77]. This states that explainability can be used for the following purposes:

- **Model editing:** Changing the knowledge or behaviour of the LLM, such as locate-then-edit [81].
- **Enhance model capability:** By understanding the underlying mechanisms of the model, conclusions can be drawn to improve the model. This can improve the use of long text and help to improve the performance of ICL.
- **Explainable generation:** The use of explainability can also help to reduce hallucination during inference and ethical bias during training.

Observations

This can now be compared with the findings of the formative study. Explainability was cited as an advantage over opaque, closed-source models. They acknowledge that it could be useful to them if implemented and used correctly. Several even recognised the potential for customers and other stakeholders. This recognises the potential of XAI in the area of LLM. When asked what the benefits of explainability would be, the following four reasons were given.

Observation 4.3: Understanding model behaviour First it was mentioned that it could help to better understand the model's behaviour. Participant 5 wonders why the model gives a certain answer. In addition, P1 & P3 also questioned the model's behaviour. It could be useful to understand better how the model works in order to use it better or to improve it.

Observation 4.4: Better prompt design Two participants mentioned better prompt engineering as a potential benefit, especially if it can be used to see which parts are most important or to correlate changes from input to output. This is in line with strategies one and five from the list above from Wu et al. [132].

Observation 4.5: Fault Analysis Regarding the scenario in which it can be useful, it has been mentioned that failure analysis is a potential stage. At this stage, it can be challenging to gain actionable insights, as discussed in the theme on challenges in subsection 3.2.2. P3 stated that in this case, understanding the underlying reasoning would help to investigate why the failure occurs.

Observation 4.6: Improving security & trust Another benefit described was improving the security and trust of the models. This could reduce bias, as mentioned by P3, or try to detect hallucinations. This observation is in line with point 4 of the list of usable XAI for LLMs mentioned above.

Observation 4.7: Quality assessment The final potential benefit of XAI is to assist in the quality assessment of the model. This involves assessing the capabilities of the model. As discussed in the background, LLMs may have emergent capabilities that are difficult to find and evaluate. Explainability should help the NLP data scientist to identify the strengths and weaknesses of the model. This will help to ensure that it is used for the right applications and performs as expected for unseen data points.

XAI used with LLMs in practice

This sub-theme describes the use of explainability techniques in the industry. Firstly, the literature on this topic will be presented, and then the results of the formative assessment will be discussed.

Literature A study by Kaur et al. [54] investigated how interpretability tools are used by data scientists implementing machine learning solutions. Several conclusions were drawn from this study. First, they showed that there is a mismatch between the interpretability tools that are intended to be used and the data scientists, resulting in over- or under-use of the tool. Second, it showed that participants underused the tool because they were not satisfied with its usefulness. Thirdly, it was found that the visualisations can be misleading and confusing and do not follow usability guidelines. Finally, it was concluded that XAI helped them to assess the readiness of a solution.

Observations

Although the participants recognised the potential benefits of XAI, the interviews revealed that explainability was not being used in their current work. Several reasons were given for this, which will now be discussed in more detail.

Observation 4.8: Unawareness as a reason for lack of adaptation A contributing factor to the lack of adaptation of explainability techniques may be the lack of awareness of data scientists. It was noted that participants were not fully aware of all current techniques, which is understandable given the fast pace of research. P2 mentioned that it is difficult to keep up to date with techniques in this area.

Observation 4.9: Perceived usefulness as a reason for lack of adaptation Another contributing factor may be the lack of valuable explanations. Several respondents commented that they did not see a good use case for explainable LLMs. This does not mean that they did not see the potential, but they did say that most of the current challenges would not be fully solved by explanations.

Observation 4.10: Fitting metrics as a reason for lack of adaptation In addition to unawareness and usefulness, explainability was also mentioned as a secondary metric. Participants mentioned only looking at performance metrics such as F1 or precision, or business metrics such as cost or Customer Satisfaction (CSAT). The interpretability or explainability of the model is a secondary concern, especially when the stakes are not very high. As one participant explains: ”There is no real demand from stakeholders; everyone seems to accept that they are black boxes, complex and opaque, as long as the performance is good”.

3.2.5. Workflow

This section examines the workflow of the NLP Data Scientist. This will be done by providing an overview of the different steps involved in the workflow and investigating evaluation strategies and industry requirements. As a result, a tool can be built that is more in line with this workflow and the tools used in this role. Firstly, the literature is reviewed, and then all the observations are given, after which a diagram is shown to give a complete overview. Finally, an overview of the tools currently in use is discussed to give an insight into the tools used in an industrial setting.

Steps

To further understand the workflow, it is necessary to examine the individual steps performed by a data scientist. The literature will also complement this section and is presented first.

Literature The steps are divided by Wang et al. [127] into three distinct phases. The first phase is the *preparation step*, which includes data acquisition, cleaning, labelling and feature engineering. The second phase is *modelling*, which includes a model selection step, a hyperparameter optimisation step, an ensembling step and a model validation step. The final step is *deployment*, which involves deploying the model, monitoring and finally improving the model if necessary. It is visualised in Figure 3.1.

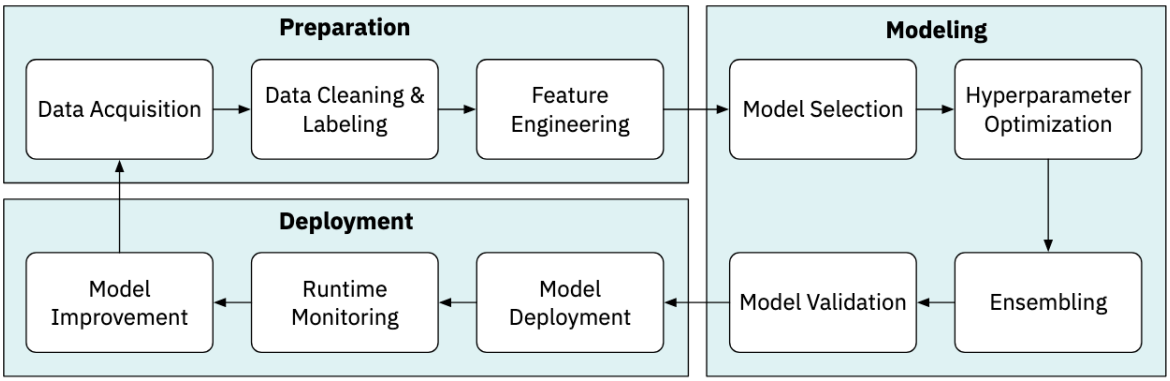


Figure 3.1: Workflow of the Data Scientist as presented by Wang et al. [127]

Observations

Observation 5.1: Individual iterative steps The steps involved in an NLP data science project can be divided into team collaboration and individual efforts. The team, which typically includes several data scientists, mainly revolves around the groundwork of the project. These steps include considering different solutions, evaluating metrics, and setting up a common project structure. The steps are then executed, often iteratively, to refine the performance.
When compared to the steps described by Wang et al. [127] and shown in Figure 3.1, it can be concluded that they are similar when put in the right context. One notable difference is feature engineering, which is a complex task for *standard* Data Science tasks but is straightforward for NLP tasks because the tokenisers and embeddings are provided with the model.
There is also a similarity in the choice of models. In NLP tasks, different (open/closed-source) models are tested, configured and possibly trained, similar to *standard* Data Scientists.
The use of LLMs is at a very early stage in the company. Not many NLP Data Scientist solutions are in production yet. Therefore, this part has been left out of the visualisation. The final workflow of a development cycle is shown in Figure 3.2, created from the formative interviews.

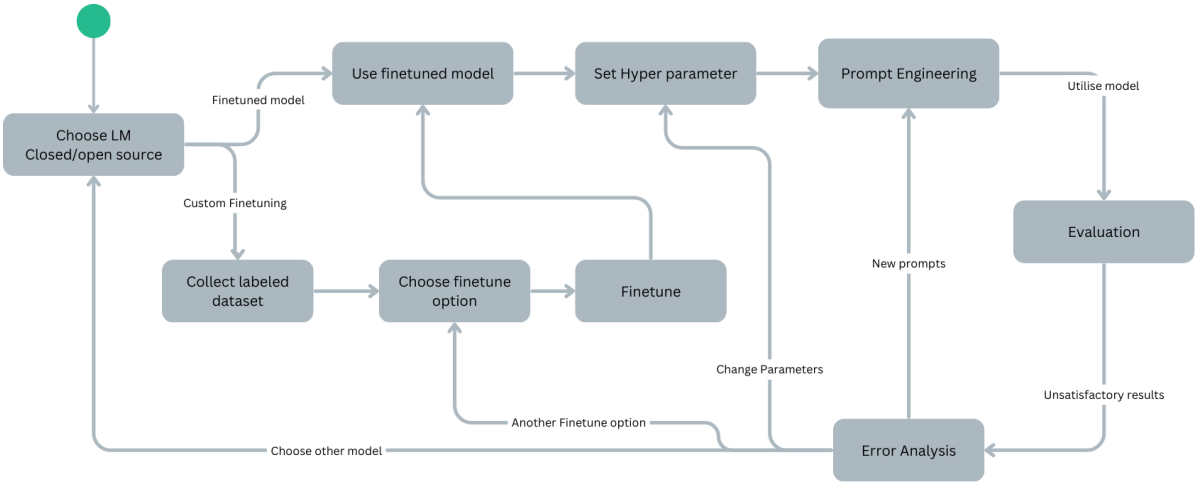


Figure 3.2: Workflow of the NLP Data Scientist

Evaluation

During the interviews, the evaluation strategies used by data scientists were discussed. Evaluating outputs can be difficult, mainly because it is often impossible to quantify what a good answer entails or which is better. Here are some observations on evaluation metrics & strategies.

Observation 5.2: Performance metrics Performance metrics are highly dependent on the task at hand. When the problem is classification, metrics such as precision, recall and F1 scores are used. However, with the latest NLP problems, such as generative tasks, it is more difficult to use these performance metrics. BLEU or ROUGE are used in these tasks, but participants acknowledged the challenge of defining a good performance measure.

Observation 5.3: Subject Experts In practice, performance is assessed using several metrics. This is partly done with the help of Subject Matter Experts (SMEs) who are (as the name suggests) experts in the specific field of the project. These SMEs give feedback to the Data Scientist on the quality and what parts of the solution can be improved.

Business practices

Observation 5.4: Company practices It is interesting to have a high-level overview of the data process at Elsevier. Within the company, several phases are defined: the design phase, the experimentation and development phase, the productionisation phase and the maintenance & monitoring phase.
For each of these phases, the developer is asked specific questions. These include questions about the decisions made in the project and whether these decisions are responsible and well-considered. For example, data scientists might be asked if they are considering using interpretable AI or explaining the results. It is interesting to note that from a business perspective, there is an incentive to think about the explainability of the solution.

Observation 5.5: Industry metrics In industry, projects are initiated to improve business operations in various ways. A business value needs to be defined within the project. Therefore, much weight is given to what the stakeholders think is important, and the data scientist needs to understand the business problem as part of the early stages of the project. The financial aspects of a project are also related. It can take time to create a cost-effective solution.

Technical Stack

The fourth sub-theme identified during the analysis is the technical stack. It is important to understand what tools the NLP Data Scientist uses in order to create a tool that will work in concert with the other tools.

Observation 5.6: Tools used Several tools were mentioned in the interviews. As P6 also pointed out, all users use different tools, so it is not set in stone what to use when. P6 also pointed out that data scientists can often use any tool they want. However, some tools were mentioned several times, making them more likely than others.
Several tools were used to load the LLM. For the closed-source models, most participants liked to use the libraries provided by the company, such as OpenAI & Anthropic. Huggingface, a popular platform for loading and publishing open-source models, was used for the open-source model.
For experimentation, Jupyter notebooks were often used locally or in AWS Sagemaker. These are popular text block-based tools for exploratory data analysis.
Finally, other tools mentioned were MLFlow & Langchain. These are tools that help with querying and building products around LLMs. MLFlow was often used to compare prompts and parameters, which were logged and stored in a database. Langchain was most often used to create RAG-based applications or to chain prompts.

Workflow desires

The final sub-theme relating to working practices is a set of additional wishes within a project. Several observations were made about the desire for a tool within a project.

Observation 5.7: Tracking experimental results We can conclude that they want their experiments to be tracked. Tools like MLFlow are used to track the experiments and give a good overview of the different prompts and outputs. These are very useful for debugging and comparing different solutions and models.

Observation 5.8: Verifying Hypotheses Most NLP data scientists start with a few hypotheses. These formulate different solutions to the problem. If the hypothesis is wrong, debugging tools can be used to understand what went wrong. Therefore, a tool should ideally present these insights.

Observation 5.9: Easy collaboration The final observation is the desire for easy collaboration. There is a lot of collaboration between people, sharing approaches and results. This could be other data scientists, but also other staff and managers. This part of the workflow is important to remember when developing a tool, not specifically for explainability.

3.3. Conclusions

Several conclusions can be drawn from this formative study. First, the NLP Data Scientist has several use cases where they now use LLMs for classic NLP tasks. In these tasks, they have difficulties with error analysis, prompting and uncontrolled output. There are several factors that determine which model the NLP data scientist uses. Opacity is one of them, but more important is the performance and usability of the model. Because the closed-source models are easily accessible through an API or Python library, this is often the preferred model.
XAI is considered useful for prompt engineering, model evaluation, and quality assessment. However, no explainability methods are currently used. This is due to low business incentives, uncertainty about the usefulness of the explainability method, or a lack of awareness of the potential methods. The lack of awareness is understandable because of the fast pace of research in this area.
For a tool to be useful, it should work with the Huggingface platform and have a similar workflow to MLFlow. Collaboration is often used, so the tool should support this. Lastly, it should be able to reproduce the results, as this is one of the challenges.

4 Development

This section describes the tool's development process. This is done by first creating the requirements, which are presented in section 4.1. These explainability and tool requirements can be used to create a design, as discussed in section 4.2, followed by a Proof of Concept (PoC) that was developed (section 4.3) and used for evaluation. The evaluation of the tool is discussed in chapter 5.

4.1. Requirements

This section presents the requirements derived from the observations. First, the requirements' definition is explained, followed by the final requirements.

Method

When defining software requirements, a common distinction is made between functional and non-functional requirements. Chung et al. [21] defines functional requirements as a function that the software (component) must be able to perform. On the other hand, non-functional requirements describe *how* the software will do that. This can include performance requirements, how adaptable the software is, or the usability of the system. High quality requirements have been created using the standards described by Mall [79]. This standard states that requirements must be complete, consistent, specific, extensible and traceable. In this study, it has been decided to create a separate section for the XAI requirements, as this is the basis for this research. This will be a mapping between the observations and the available explainability techniques, supplemented with relevant literature. These explainability requirements will be described first, followed by the functional and non-functional requirements.
These requirements will be used to design the tool, and its implementation can then be used to evaluate it. It should be noted that it is not feasible to assess all requirements properly. However, they will still be defined as they are derived from the observations.

4.1.1. Explainability Requirements

Traditionally, explainability is used as a non-functional requirement [57, 16]. However, as the research focuses on using the best XAI techniques currently available, it is presented separately. This is done using the observations and supplementary literature. Several papers have discussed desirable properties, including Ras, Gerven, and Haselager [102], Liao et al. [72], and Hohman et al. [45]. Liao et al. [72] evaluates the metrics based on context and user, which for this research are chosen to be Model Auditing, Overall and Capability Assessment.
It should be noted that each accountability requirement can potentially create tensions with others. An ML engineer might find one property more important than a layperson [72]. Then, a trade-off should be made between the two properties [18]. The properties are now listed as Explainability Requirements (XRs).

XR 1: High Fidelity Fidelity is the degree to which the explanation matches the input-output mapping of the model. It is a synonym for fidelity, and it is argued that it is the most important property of an explanation because it cannot generate valid explanations without being faithful. This is confirmed by Liao et al. [72], which shows that fidelity is most important for model improvement. Finally, this is also shown by Sokol and Flach [112], which also shows the importance from the explainee's point of view.

XR 2: High Interpretability The extent to which the user can actually gain insight into actionable results is sometimes, therefore, called actionability [112]. It has been divided into two sub-properties: 1) high clarity, meaning that the resulting explanation should be unambiguous, and 2) high parsimony, meaning that the explanation is simple, selective and concise [83, 112]. Note that optimal parsimony depends on the user's skills.

XR 3: high explanatory power For the method to be effective, it should be able to explain many different phenomena. This can also be expressed in terms of how many questions it answers for the user [102].

XR 4: Interactivity Interactivity is also an important feature. As shown in Hohman et al. [45], users could not fully comprehend static explanations but were able to understand them better when they could interact with the explanations. This is also supported by Miller [83] and [112].

XR 5: Contextfullness Ensuring explainability can also provide an understanding of the limitations of one's own explanation. This will ensure that the user will better understand the conditions under which the explanation is true [112].

4.1.2. Functional Requirements

This section presents the Functional Requirements (FRs) of the user-centric tool.

FR 1: Load an open-source LLM As mentioned earlier, open source models are required to be able to apply explainability methods. Based on observations 3.3 and 5.6 regarding the models and tools used, we can conclude that these models are primarily hosted and loaded via Huggingface. Therefore, this tool should be able to load models from the Huggingface platform.

FR 2: Open to model configuration When the loaded models are used, practitioners configure them in their own way. Therefore, the configurations must be configurable by the user. This is in line with observation 5.1, which presents the individual steps. Here, users mentioned that they configure the model according to their needs for each specific use case. Ensuring that this is possible also helps to improve the usability of the model, which is a factor why users tend to use closed source models, as discussed in observation 3.5.

FR 3: Scalable to large models If the infrastructure allows it (e.g. large GPUs), it should be possible to use large models in this tool. This is because these are the models that are actually used in an industrial environment. This is also in line with company practices (observation 5.4) and their industry metrics (observation 5.5).

FR 4: Input-Output analysis It should be possible to see the input-output pairs of the model. Users may have different hypotheses that they would like to test and see how the model responds. This is one of the most important goals of the NLP Data Scientist, which this tool should support.

FR 5: Compare outputs Next to investigating input-output combinations, it should be possible to compare different input and output pairs to get a better understanding of which inputs work and which do not. This can help with the error analysis, as discussed in observation 2.2. Additionally, having a different model and comparing them is observed in 2.6 as a difficult task, therefore comparing the outputs of two models should also be possible.

FR 6: Define own metrics As discussed in observation 5.5, all tasks and projects have their own set of metrics to evaluate the model. Therefore, it should be possible to let the user define its own metrics. This can be done by returning the model's output, which the user can then create their own metrics on.

FR 7: Result tracking Another requirement discussed in observation 5.7 relates to tracking results. By tracking results, the user can go back later and see their own results. This helps with the challenge of reproducibility because you can show the different experiments that were done after the fact. This is in line with the reproducibility challenge discussed in observation 2.2.

FR 8: Different organisational levels for results From observation 5.9 regarding the ease of collaboration, it would be beneficial to have different organisational levels to organise the different outcomes. This could be a project where each user has their own project. This would make it easier to see only the user's result, but still share it with colleagues.

FR 9: Presentable data As mentioned in 5.3, the NLP Data Scientist often works with SMEs and other stakeholders such as the Responsible AI Expect. Therefore, the data generated by the model and the explanations should also be presentable to these external parties. This means that it should also be usable by other stakeholders who may want to investigate the results.

4.1.3. Non-Functional Requirements

This section presents the Non-Functional Requirement (NFR) of the tool. These are based on observations and software engineering principles.

NFR 1: Extensibility In order to be fully usable, the tool must be extensible in several ways. First, as this tool could be used to replace older models, it is useful to be able to compare different models, as stated in observation 3.1. It should therefore be possible to add other (older) models and even other model architectures. In addition, new XAIs may be introduced in the future, which may be better than those currently decided. Therefore, the tool should be able to add additional explainability techniques as needed. Finally, in terms of extensibility, the front end of the tool should be set up correctly to handle different visualisations. Practitioners may need other visualisations, which they should be able to add themselves.

NFR 2: Easy to learn Learning how to use a new tool can impose a cognitive load on the user. This should be avoided by making the tool as easy to learn as possible. This means intuitive command names, consistency across the platform, and a component-based interface [79].

NFR 3: Low costs According to observation 2.4 on cost vs. performance, the tool should be as cheap as possible. This is also in line with observation 5.5 on industry metrics. This means using open-source libraries and minimising resources such as cloud computing or databases.

NFR 4: Performance In line with observation 3.4 on cost versus performance, this tool should not have a negative impact on performance metrics.

NFR 5: Speed It is required that the tool does not hinder development too much. It should be recognised that calculating these explanations can be computationally slow, but it should not make development too inefficient. This is discussed in observation 4.5 on industry metrics and, more importantly, a user desideratum as discussed in 1.6.

NFR 6: Improved Documentation Well-written resources are an important requirement. This will help the data scientist in several ways. First, it can help them understand what kind of techniques are being implemented and how to interpret the explanations. User guides and online help are also recognised by Mall [79] as useful.
Secondly, observation 2.1 noted that it is difficult to keep up to date with the latest technology. Having resources will also help to give them information about the current state and where to find the latest information. This could also help them with ignorance (observation 4.8).

NFR 7: Easy installation As it should fit into the user's workflow, it is important that it is compatible with Python. It should also be easy to install and use. As the most commonly used tool according to observation 5.6 is Jupyter Notebooks, it would make the most sense to implement it there. Additionally, Pypi could be used to publish the package.

NFR 8: Visibility of functionalities and state Another non-functional requirement shown by Tomsett et al. [122] is the visibility of all functionalities. This study showed that this is useful for the user and gives an overview of what the system can do. In addition, Tomsett et al. [122] also mentioned that the user should always be informed about the state of the system. This could be idle, loading, ready, etc and should be visible from within the tool.

Conclusion

This section gave an overview of the system's requirements. It first presented the explanations that should be implemented, after which the tool requirements were described. Together, they give a good overview of what the tool should be able to do to satisfy the needs of the NLP Data Scientist. These requirements can now be used to create a useful to support the target user.

4.2. Design

A design will guide the implementation phase of this study. Two designs have been created: an architectural design and a User Interface (UI) design. The architecture gives an insight into the inner workings of the tool and how the different components interact. The UI is a visual representation of what the front end of the tool will look like. The design is based on several sources. First, the requirements were used as the main starting point. Then, various design principles presented in the literature were used, which are discussed in this chapter. Finally, other implementations were used as a reference point for what was possible and what was not. Together, these sources gave a complete picture of what the tool should look like and what components should be included.
Regarding the tool's design, the first and most important topic concerns the explanations. The decisions regarding the techniques are presented together with their visualisation design. Then, the architectural system of the components is described. Finally, the UI views and components are presented.

4.2.1. Explainability Techniques

There are many different Explainability Techniques (XTs) and implementations, so a selection is made based on requirements. The considerations are described first, and then the final three techniques are presented.

Considerations

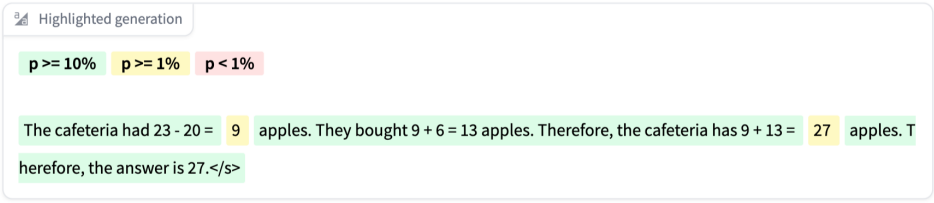
Several explanation techniques are potentially helpful to support the needs of the NLP user. Several decisions could be made based on the formative study and the requirements.
Firstly, the visualisation of attention heads could be considered, but is not chosen at this time. This is because there is a lot of debate about the fidelity of attention heads [136, 12]. Some argue that raw attention does not identify the most important features for prediction [108, 50] or during code generation [60]. Others say that attention mechanisms do not contribute as much to prediction as thought [84]. Since fidelity is one of the primary desired properties according to XR 1, these techniques are the most appropriate. In addition, according to observation 1.3, visualisation of the model is less important to the NLP data scientist. Therefore, it was decided not to include the visualisation of the attention mechanism.
The fidelity is also too low for immediate explanations like Chain-of-Thought (CoT). Experiments show that CoT reasoning can contradict the output of the model, making it unfaithful. This was shown by Bubeck et al. [14] and Wu et al. [132]. However, they concluded that performance was still improved. It is therefore recommended that data scientists keep this method in mind, even though it is not implemented in this tool.
Finally, it should be noted that most data scientists in the evaluation phase examine data at a local level, so the explanations should also work at a local level. As XR 2 states, explanations should also be interpretable. Global explanations might be difficult to interpret and, therefore, not the most appropriate technique.

Final Explainability Techniques

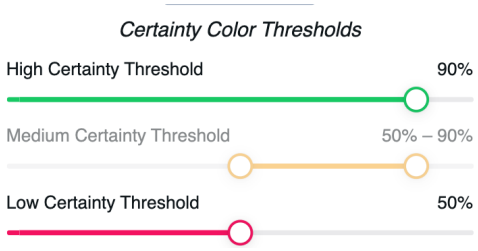
Based on the above considerations, three techniques have been selected for this tool. The techniques are visualisation of uncertainty, attribution of features at the token level & contrastive explanations. They will now be described in more detail, after which the visualisations of these techniques will be presented.

XT 1: Model uncertainty Displaying the confidence of the model is the first XT. It will highlight parts of the output where the model was uncertain, or give a general percentage of certainty. This is also recognised by Cirqueira, Helfert, and Bezbradica [22], which notes that uncertainty can highlight the limitations of the model. Furthermore, understanding the sources of uncertainty can provide insight into biases and hallucinations in the model and will provide insight into the capabilities and limitations of the model. This was also explored by Sun et al. [115], who concluded that uncertainty was useful, but that uncertainty alone was not sufficient to satisfy their need for explanatory power. It should be noted that there are concerns about the interpretability of *uncertainty* as it may not match people's intuition about what it means for a model to be uncertain [70].

Given XT 1, the uncertainty of the model should be visualised. This could be visualised by highlighting tokens and colouring them according to uncertainty. To make this interpretable, certainty is divided into three categories: High, medium, and low uncertainty, which are green, orange, and red, respectively. It is left to the user to define their thresholds and ranges, giving them more flexibility based on the model and use case. An example is shown in Figure 4.1, as demonstrated in a Huggingface Space [1].



(a) Certainty Highlighting



(b) Configuring the different thresholds

Figure 4.1: Displaying Uncertainty

XT 2: Token-level Feature Importance The second technique is token level feature importance, which highlights which tokens contributed positively and negatively to the predicted token. It can be interpreted as where the model looked and which parts of the prompt contributed most to the predicted token. This technique was chosen to help the NLP data scientists improve their prompts, which was a challenge as described in observation 2.4

Based on XR 2, we need a way to visualise the feature importance of each token. This will allow us to answer how much each token input contributed to the output produced. The visualisation of feature importance is often done by using a saliency map over the different tokens. An example is shown in Figure 4.2.

[1] Source: https://huggingface.co/spaces/joaogante/color-coded-text-generation

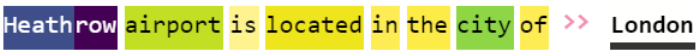


Figure 4.2: Feature Attribution (by Ecco Alammar [6])

Since token importance is calculated for each generated token individually, there should be a way to interact with the tokens to update the highlighting. Only then will it be easier to interpret, as it is now more interactive which align with the requirements. This will be done by updating the colouring depending on which token the mouse is hovering over. In addition, clicking on a token will freeze the highlighting, making it easier to focus once an observation has been made.

XT 3: Contrastive Explanations The last required explanation should answer the question *"Why not Y instead of X?"*, as this was mentioned as part of the error analysis in observation 2.2. This can be answered with so-called contrastive explanations. Although this has a lower explanatory power (it only answers one question), it is a question that was observed several times during the formative study. It has also been shown by [134] that this is a useful explanatory style and an important question to ask. Therefore, this explanation technique was chosen for implementation.

For XR 3, a similar visualisation can be used as for the feature importance. Figure 4.3 shows what the contrastive explanations could look like. Here you have to give an additional word and it will calculate the tokens that reflect why it did not produce the alternative token. In Figure 4.3, 1 is the feature attribution and 2 is the contrastive explanations.

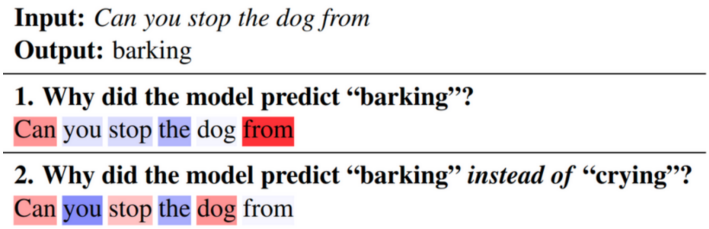


Figure 4.3: Contrastive explanations (from interpret-lm [134])

4.2.2. Architectural Design

The next section presents the tool's architectural design. This design is similar to [88], but then more specific to LLMs. There are two main components. The Python package should be included in its LLM development, often in a Jupyter Notebook environment. The second component is a Application Programming Interface (API) responsible for storing and retrieving the model's logs. This also includes a frontend that can load the logs and provide more insight into the model and its output. An overview is given in Figure 4.4. After that, each component is described individually.
The system architecture takes into account NFR 1 (extensibility) and NFR 7 (industry practices). By creating a modular system, it is easier to extend with new functionalities and it is easier to comply with industry practices, thus fulfilling these two Non-Functional Requirements (NFRs).

4.2.3. Component 1: Python Package

The Python package should have the following features. First, the package should be able to load Huggingface models as described by FR 1. This is achieved by creating a wrapper around the Transformer package. This will also make it possible to add custom configurations and scaling options, as described by FR 2 & 3. The Python package is responsible for calculating the explanations. This should be done in a standardised fashion. In addition, the models should be used in the Python environment, using user-defined metrics, as presented in FR 6. When the model is used, the output should be logged and sent to the API to ensure that it can be retrieved and viewed later (FR 7).

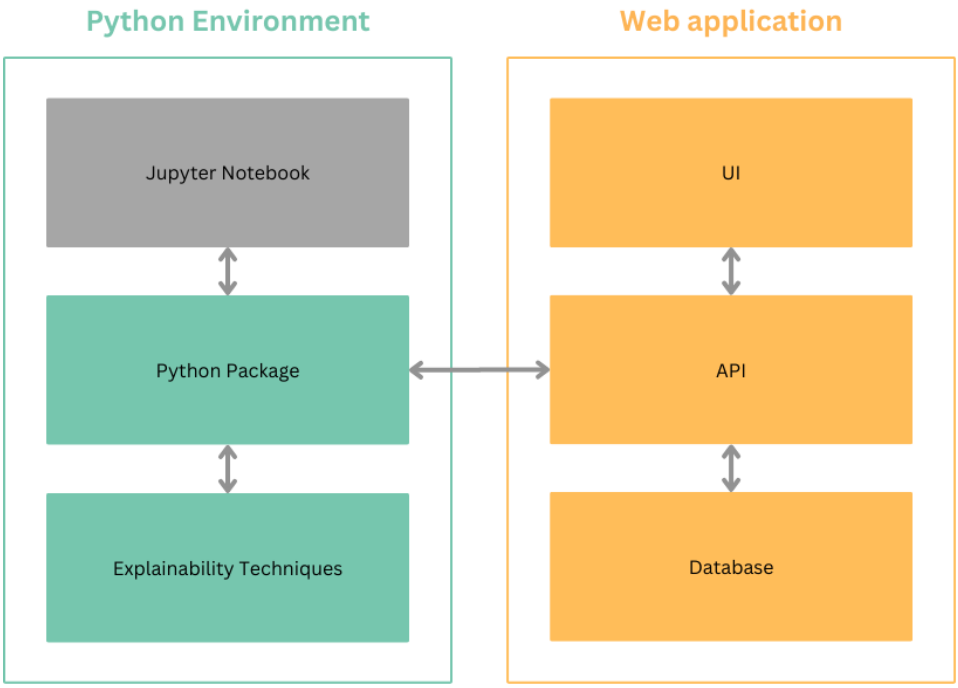


Figure 4.4: Designed architecture

Finally, the Python package should be installable, making it easy to set up, as stated in FR 7. This could be done by publishing it to a package manager and making it fully open source.

4.2.4. Component 2: Web application

The web application will have two separate sub-components. Firstly an API and secondly a frontend. The API will act as a communicator between the Python package, the database and the frontend. The Python package sends the logs to the API, which should store them in a database that satisfies functional requirement 7. The API is also responsible for retrieving the logs to send to the Python package and the web application frontend.
The UI is responsible for interaction with the explanations, which satisfies functional requirement four regarding input/output analysis. It should have an overview of all the experiments that have been explored and provide the appropriate information about the experiment. In addition, the UI should visualise the explanations and provide the possibility to interact with them. The visual design of the UI will now be discussed.

UI design

The User Interface (UI), also called the frontend, should be able to display the runs and the details of the explanations. It has been decided to use a conventional layout that is easily recognisable. This means a navigation bar at the top of the UI with clear titles and a clickable tool title that takes you to the home page, which should help for NFR 2, meaning it should be easy to learn. In addition, the runs are presented as a table, which gives additional recognition as this is similar to other tools such as MLflow.

As part of the non-functional requirements, it was desired to visualise all the functionalities of the tool (NFR 8). This is done using Direct Manipulation Interfaces, i.e., icons, objects, and widgets such as buttons [79]. This is done for all actions as well as the Compare button. This requirement also states that the state of the system should be visible. As NFR 8 also specifies, the functionalities and states of the system should be visible. Two types of states should be visible: 1) whether it is connected to the API and whether it is loading or idle, and 2) whether it is connected to the backend. The footer will display the state of the API, giving information about whether it is connected or not. A loading spinner should be used for the components that take longer to load. The final widgets, states and icons can be found in the implementation section (4.3).

The final design decision made is the use of projects. Projects will be a collection of runs, making it easier to find the results for each user and project. This will help to organise the different runs and experiments as defined in FR 8.
A total of six distinct pages have been developed. The visual design of these views is shown in Figure 4.5

(a) **Home Page** The home page, the first page the user sees. Links to all other pages are displayed here.
(b) **Projects page** Where the various existing projects are displayed and where a new project can be created. (FR 8)
(c) **All Runs** A list of runs within a specific project in a presentable format (FR 9)
(d) **Detailed view** The page where explanations are displayed and the user can interact with them. It displays all the designs presented in the previous section.
(e) **A Comparison View** The page where multiple runs can be compared in a side-by-side view, satisfying FR 5.
(f) **Resources page** The page where the user can read more information about the implemented XAI techniques, satisfying FR 6 on improved documentation of XAI techniques.

4.3. Implementation Details HELMET

This section introduces the implementation, which is called Human-Evaluated large Language Model Explainability Tool (HELMET). There are four separate entities in the final implementation, a Python package, a API, a UI and a database.

The Python package and the UI are able to communicate with the API, which is a NodeJS server that is able to save, load, update and delete runs. In addition, projects can be created from both the UI and the code. The API has access to the database, which is a non-relational database in MongoDB. More information on this will be provided later. The final folder structure can be found in the Appendix C. In addition, some features of the `HELMET` Python package and frontend are highlighted by a series of code examples. The implemented methods and libraries used are shown in Figure 4.6.
The implementation is structured as a monolith repository, were the Python package and web application are separated. The folder structure is given in section C.1

4.3.1. Python Package

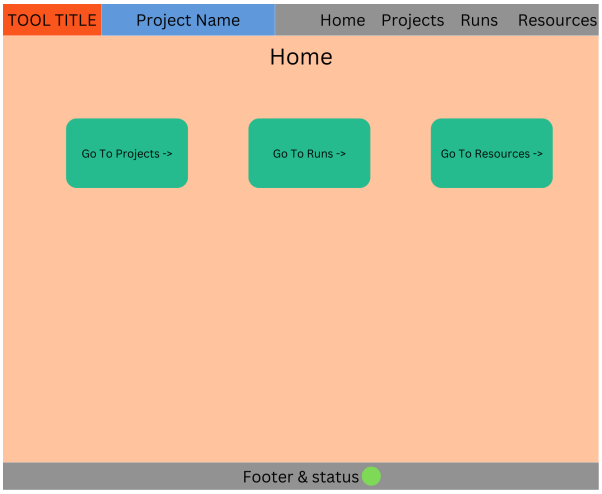
The most important part of the implementation is the Python package. This is a wrapper around the transformers library that is used to load the models. This is where the explainability techniques are implemented, so these are discussed first. Other details will then be presented.

Implementation Explainability Techniques

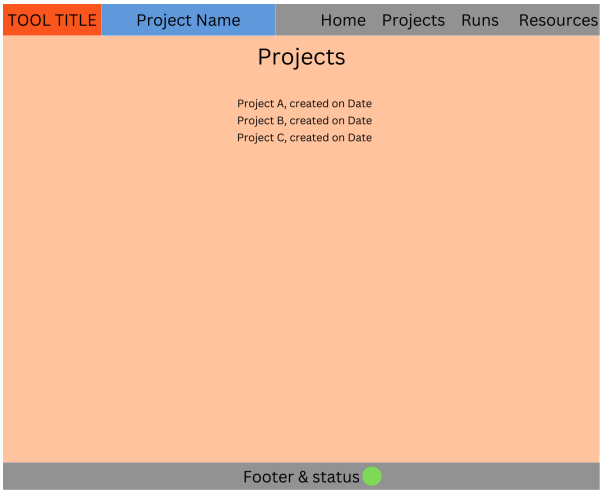
For each of the three explainability techniques presented in the design, the implementation details are now given.

Uncertainty The uncertainty is calculated from the transition scores using the `compute_transition_scores` function in Huggingface. These are the log probabilities of this token conditioned on the log softmax of the previously generated token, followed by a normalisation step. This is a convenient way to quickly get the scores of the selected tokens at generation time, as mentioned in the library documentation. This can be interpreted as the confidence in the model at each generation step. This method is always calculated during the generation of the output, as this is the only time when this can be done efficiently without saving all the logits.

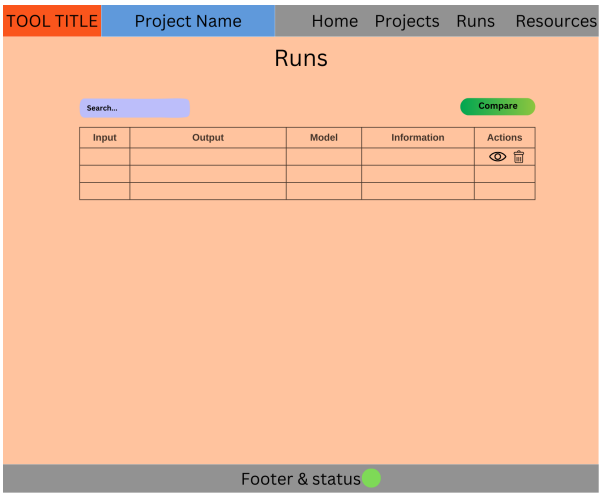
Token level feature attribution It was decided to use the Input x Gradient Feature Attribution technique, originally proposed by Sundararajan, Taly, and Yan [116], as this was the most computationally feasible. This is done by calculating the gradients and comparing them to a baseline to see which tokens are different [109].



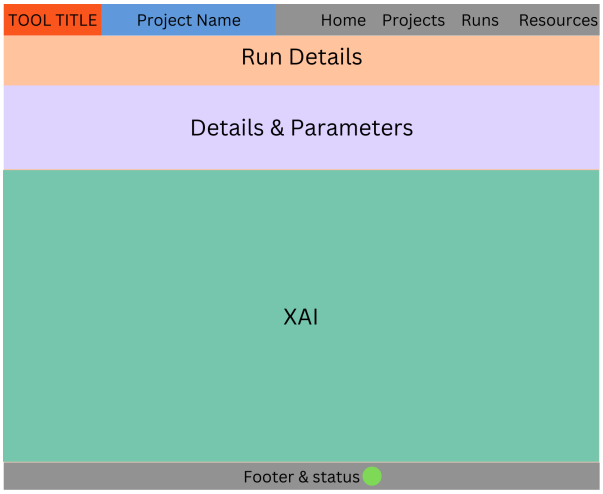
(a) Home Page



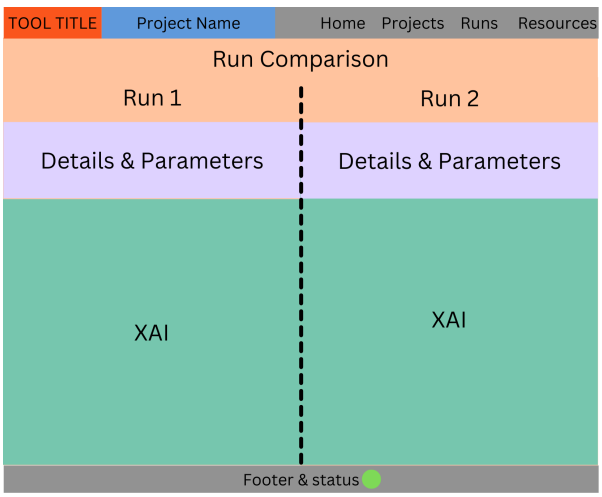
(b) Projects Page



(c) Runs Overview



(d) Details single run



(e) Comparison view

(f) Resources page

Figure 4.5: The six views designed for the UI

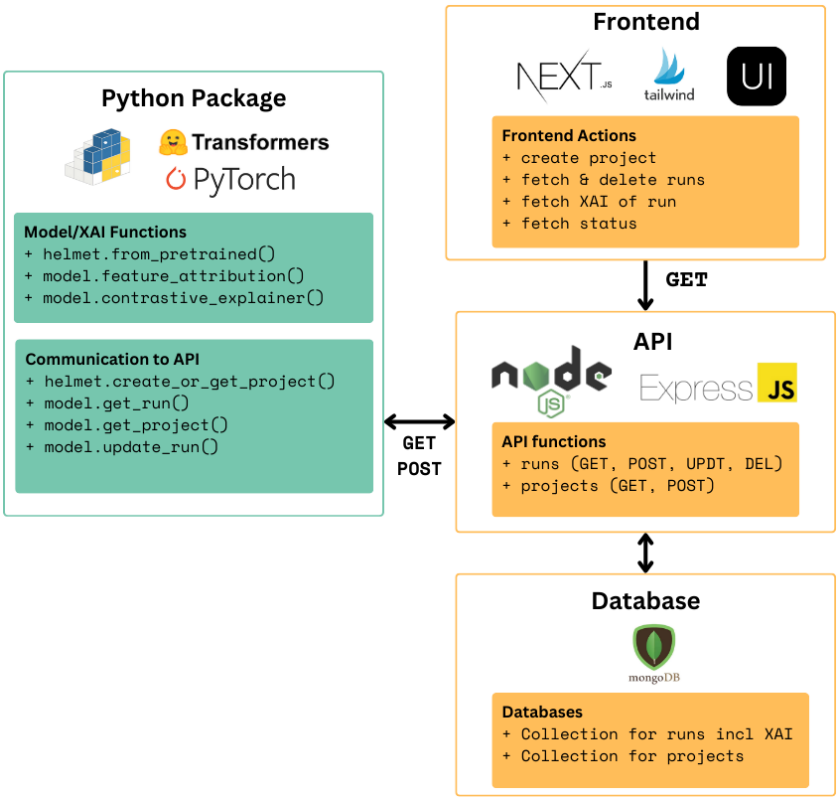


Figure 4.6: Final Implementation architecture

Contrastive explanation Contrastive is computing the feature attribution and additionally computing the gradients for another token. Then, the difference can taken to get an answer to the original question of why not B instead of A. This technique is proposed by Yin and Neubig [134], from which the implementation is taken as a reference.

Generate Explanations When a prediction is made, the output of the model is returned to the user along with an ID. This ID can be used to compute additional explanations, as mentioned above. The Listing 4.1 gives an example of code that queries the model and computes the explanations *post hoc*.

```
output, response_id = model.predict(prompt, generation_arguments)

# The uncertainty is always computed
attributions = model.feature_attribution(response_id)
constrastive_explanation = model.contrastive_explainer(response_id, "<new token>")
```

Listing 4.1: Computing Explanations

Features

Currently, all decoder-only architectures are implemented. Encoder and encoder-decoder architectures are implemented in a shallow way to give an idea of how HELMET can be extended. This section highlights the main features. Code examples are provided in Appendix C

Installing HELMET One of the requirements was that it should be easy to integrate into the workflow of the NLP data scientist. The first step the user should take is to install the HELMET package. Fortunately, this is easily done using Pypi [2], as it is published at https://pypi.org/project/helmet/. This is done with the command `pip install helmet`. Poetry [3] is used as a dependency resolver to ensure that the correct dependencies are installed.

[2] https://pypi.org/
[3] https://python-poetry.org/

Loading a project The first thing you do when starting a project is to create a project or load an existing one. This can be done from the UI, but also from the Python page. This is based on the title you provide, which returns a project ID from either an existing project or a newly created one. This is shown in the Listing 4.2

```
from helmet import get_or_create_project

project_name = "Project Name"                # Name of the project
platform_url = "https://api.example.nl"      # URL to the platform
task = "text_generation"                     # The NLP task
project_id = get_or_create_project(platform_url, project_name, task)
```

Listing 4.2: Creating or getting project

Loading a model Any model can be loaded using the model checkpoint name, which is the standard way of referencing Huggingface models. A model can be loaded using `helmet.from_pretrained(args)`. This has similar arguments to the parallel Huggingface method, making it easy to load models. All generative decoder models are currently supported. When the device is set to `CUDA`, the models are automatically quantized. An example of how to load a model is given in the Listing 4.3

```
from helmet import from_pretrained
model = from_pretrained(checkpoint, model_type, embeddings, project_id,
    device, platform_url, model_args)
```

Listing 4.3: Loading the Huggingface model

Using the model Prompting the model is as simple as running `helmet.predict(prompt)`. Additional arguments can be given here, such as the temperature or the maximum number of tokens. These additional arguments are also stored in the run, making it easier to go back and see which arguments performed best. A coding example is given in Listing C.2

Configuration of the model Several configurations can be set during inference or when loading the model. These are needed to ensure that the inference techniques can use the embedding layer of the model, for example. In addition, the type of model and the device are required. An example is given in the Listing C.1

Data Classes Several data classes have been implemented. These are used to standardise parts of the communication, creating a more type-safe communication between the different parts of the application. The input and output is standardised within the Python package. This is to ensure that all input is parsed correctly in the prompt, but also in the frontend. In addition, each run is stored in a class, which is then stored in the database. This allows the class to be recreated when the run is reloaded from the database.

Extensibility A major advantage of the current modular implementation is its extensibility, as desired by NFR 1. The tool could be extended in many ways, such as new models or even new model architectures. It could also potentially be extended to include closed source models. In this case, the runs would still be stored. However, the degree of explicability is limited. It has been shown that some techniques do not require internal mechanisms, so it is still possible to include certain explanations [114].
In addition to new models, it would be easy to add new explainability techniques and corresponding visualisation techniques. With knowledge of both Python & JavaScript, it should also be easy to add new features.

4.3.2. API

The second component of the finished application is the API. This is there to provide smooth communication between the Python package, the frontend and the database. It is a stateless NodeJS Express server that is lightweight, fast and easy to deploy.

Its main function is to process requests from the Python package and store them in the database. All Create, read, update and delete (CRUD) operations are implemented for all runs. The API can be run using NPM [4], with the command `node server.js`.
Two routes are defined, one for the projects and one for the runs; `/project` and `/runs`. These two routes both use the `GET`, `POST`, `PUT` and `DELETE` conventions to retrieve, update, create and delete the various documents in that particular collection.

4.3.3. Database

The database is a non-relational database that stores the projects and runs. A non-relational database is chosen because different runs and explanations have different schemas. Therefore, a more standard relational database would be complex. This database can be used locally or in the cloud. The API makes it easy to connect to the database and store data.
The current implementation supports the storage of runs and projects, both in a separate collection. The runs can be retrieved using the project ID, ensuring that only the runs of the corresponding project are retrieved. An example of how it would look like in the MongoDB is given in Listing 4.4


```
{
    {
        "_id": "663b9357cf5a52ab5c96f128"
        "date": "2024-05-08T14:59:35.954Z"
        "model_checkpoint": "gpt2",
        "tokenizer": "gpt2",
        "model_type": "dec",
        "input": {
            "prompt": "Can you stop the dog from ",
            "input_tokens": [tokens]
        },
        "output": {
            "output_str": "urch",
            "tokens": [
              "urch"
            ]
        },
        "explanations": [
            {
              "explanation_method": "certainty",
              "certainties": [
                0.20173077285289764
              ]
            }
        ],
        "project_id": "66349e3bc969e501ab987f07",
        "execution_time_in_sec": 0.2844071388244629,
        "custom_args": {
            "max_new_tokens": 1
        }
    }
}
```


Listing 4.4: Example of how a run is stored in de database

4.3.4. Frontend

The final component of the implementation is the User Interface (UI). This is a web application developed in NextJS [5], a React-based web framework that focuses on easy development and deployment. This section will highlight the main features. Screenshots from the UI are can be found in Appendix D

[4] https://www.npmjs.com/
[5] https://nextjs.org/

The main feature is that all runs can be viewed. They can be viewed in a list, giving an overview of all the prompts that have been run. In addition, each run can be viewed in more detail, including the explanations. Runs are searchable and can be deleted individually or all at once. When the detail page is visited, all input, output and the custom arguments can be viewed as well, as shown in Figure 4.7

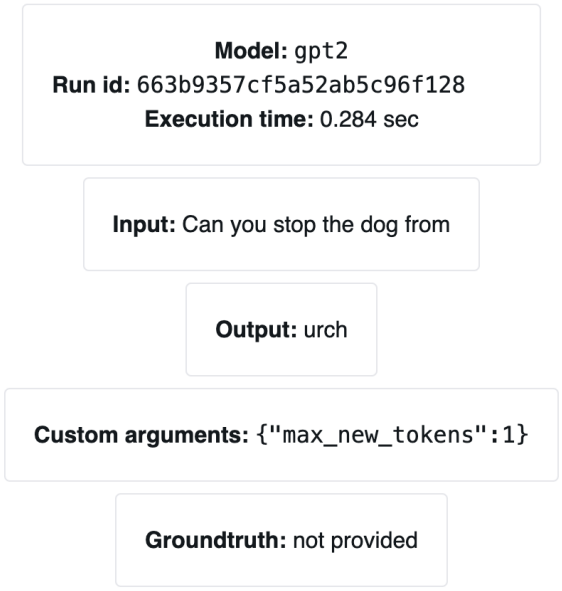


Figure 4.7: All custom arguments can be viewed

The explanations are interactive. The confidence thresholds can be edited, dynamically highlighting where the model was confident and where it was not. Also, the feature attributions are dynamically coloured depending on which token is hovered over. If the explanations are not yet computed, the code to compute them is suggested. This code can be easily copied and run from the Jupyter notebook where the model is loaded, as shown in Figure 4.8

Figure 4.8: The code suggestion when the explanation is not computed yet

Resources Another feature is the documentation of the implemented explainability technique. This page is located on the frontend, so it gives the user more information about the method if needed. This should help with the interpretability of the methods as well as make them more aware of the possibilities of XAI within LLM.

Familiar Elements The last of these are the familiar design elements. Some of this has already been discussed in the design section, in section 4.2. Some notable features are the tooltips that appear when the mouse is over a button, giving additional information about what to expect when you press it. The icons have also been chosen to be familiar to most users. They are shown in Figure 4.9

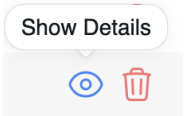


Figure 4.9: The tooltip and icons

4.3.5. UI Examples

To give a better overview of how the tool looks in practice, several UI examples are presented. These screenshots give the basic views that are implemented. These are:

- Runs page (Figure 4.10)
- Comparison page (Figure 4.12)
- Detailed Run with XAI (Figure 4.11)

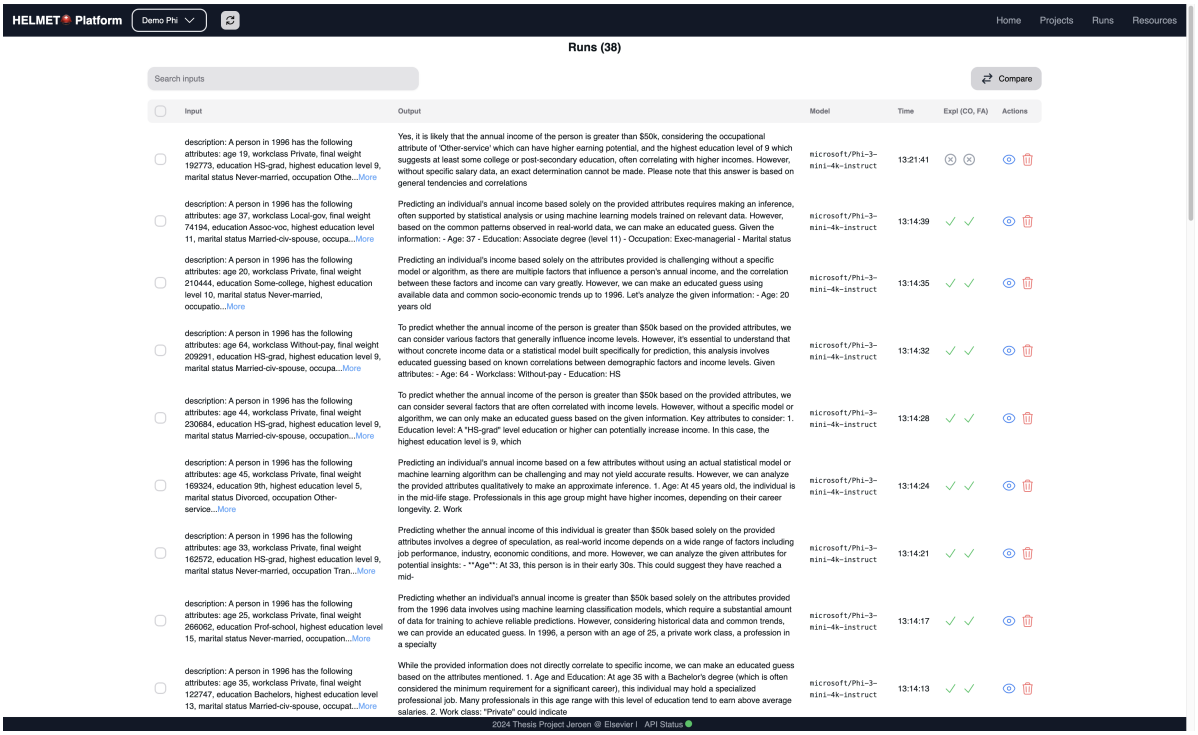


Figure 4.10: Runs page

The second column from the left shows if any explanations have been calculated for that specific run; Contrastive (CO) and Feature Attribution (FA)

4.3.6. Implementation Challenges

Implementing a tool like HELMET is quite a challenge. It should be recognised that several difficulties are associated with the task of building such a tool. Firstly, it requires extensive knowledge of LLMs and their internal mechanisms. Without this knowledge, it is difficult to know what to configure for the tool and how to load the model correctly. This is also due to the fact that the `transformers` has a wide range of arguments that can be provided. This steep learning curve should be taken into account.
Secondly, the computing power required to produce results in a reasonable time is high. Even with tiny models loaded on the GPU, debugging is extremely difficult and will take a while, especially if no GPU is available locally.
Finally, there is the implementation of the XAI techniques. As mentioned several times during the research, other tools exist. Tools like Captum [59] could be used, but the experience from this study is that they are really difficult to use. The input has to be in a very specific input and the output of the calculation is very limited. Therefore it was decided to implement all XAI techniques within HELMET again. For training and inference in a more mature environment, several other tools could be considered, including DeepSpeed, Megatron-LM, JAX, Colossal-AI, FastMoE and BMTrain.

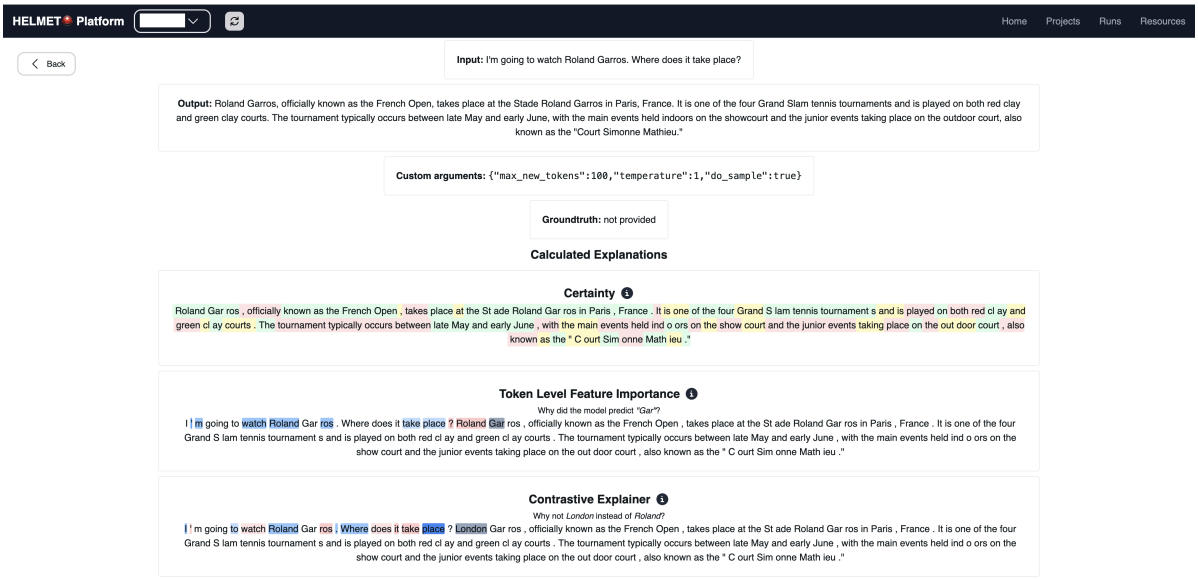


Figure 4.11: Detailed page with XAI

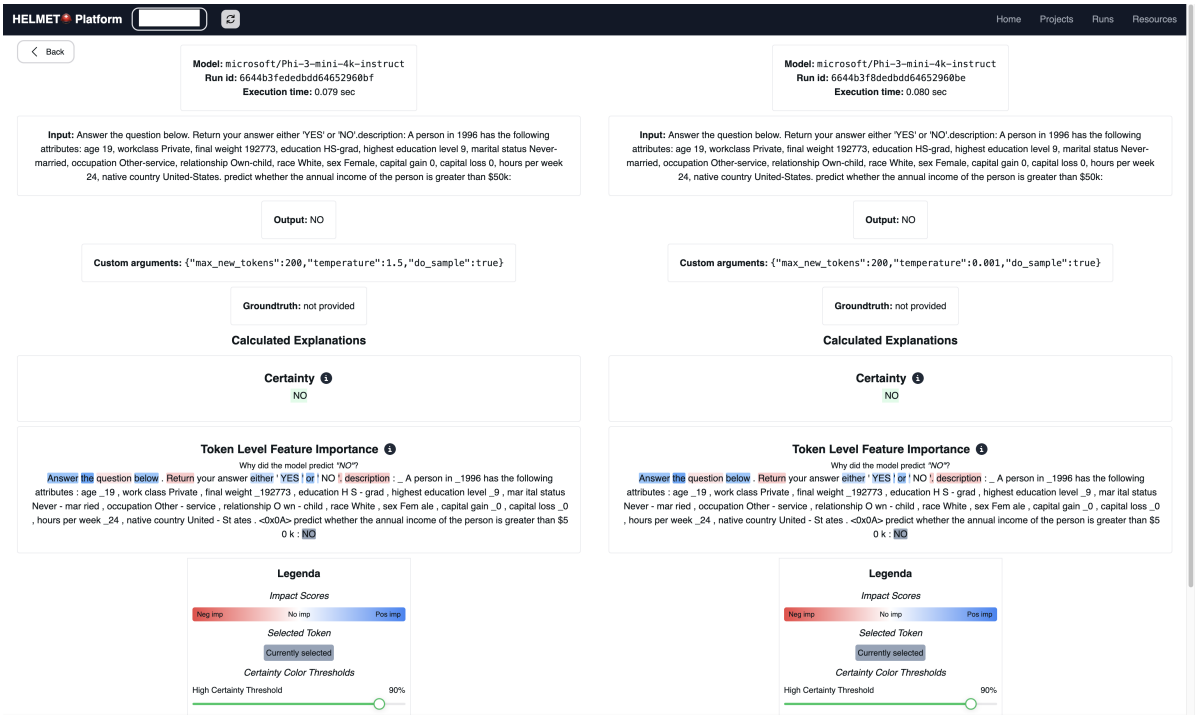


Figure 4.12: Comparison

5 Evaluation

The tool should be evaluated to validate the requirements, design, and implementation. This is done using a hands-on demo and survey. First, the evaluation method is described, followed by the results of the individual parts of the evaluation.

5.1. Method

This evaluation phase captures the extent to which the explanations are helpful to the NLP data scientist and whether the tool fits into the current workflow. By doing this, the requirements can also be validated. This is done in a human-grounded way, i.e. with real people and a simplified task as described in the background.
The process combines a qualitative evaluation and an exploratory analysis using open questions that will be further elaborated. Where to focus our attention is based on the HCAI framework presented by Xu, Gao, and Dainoff [133]. This framework describes, among other things, which primary design goals are important in the HCAI implementation approach. For example, the main goal of Explainable AI should be to create usable AI, while UX design should focus on the usability and usability of the tool. The goal will be elaborated first, and then the approach will be explained.
Three main questions should be answered during the evaluation:

1. Does explainability give practical insights to an NLP Data Scientist?
2. Does the current implementation fit the NLP Data Scientist's workflow?
3. Is the tool easy and intuitive to use?

The complete questionnaires can be found in the section E.1, section E.2 and section E.3 respectively. The procedure will now be explained, after which the three parts will be discussed in more detail.

It was decided not to evaluate existing tools. This is because all the existing tools described in the Table 2.4 are intended for research purposes. Their focus was not on implementing a user-friendly tool but on eliciting a new technique. Therefore, the existing tools were only used as inspiration for the current implementation where possible.

Procedure

A video was made to introduce the tool and its functionalities, which can still be found on Youtube [1]. This helped the participants to familiarise themselves with its features and setup. This was followed by a hands-on session in which the participants worked through an example with the researcher. Here, the participant was asked to complete a number of steps, such as creating a project, loading a model, prompting the model and interacting with the explanations. A dataset was loaded to give the participants more data to interact with. The whole session took about 25 minutes. After the session, the participants were asked to fill in the questionnaires, which can be found in the Appendix E.
The task given to the user was to judge the fairness of the model based on the explanations. This is a subset of the decodingTrust dataset, from Wang et al. [126]. The Fairness dataset was chosen because

[1] https://www.youtube.com/watch?v=z2zAmB6L7WU

it was assumed that the explanations would give an indication of how biased the model actually is by highlighting parts of the input that are biased. This is a simpler task and not related to the tasks that data scientists solve in practice. However, the focus was on interaction with the explanations and user satisfaction, not on solving the task.

Participants

Nine NLP data scientists took part in this evaluation. They were selected from three teams within the organisation. They ranged from junior to senior NLP Data Scientists, but all had experience with LLMs. It should be noted that although their official title is Data Scientist, their main work currently involves the use of LLMs. The overview is given in Table 5.1

Table 5.1: Overview of the participants

Participant	Position
Participant 1	Intern
Participant 2	Senior Data Scientist
Participant 3	Senior Data Scientist
Participant 4	Data Scientist
Participant 5	Principal Data Scientist
Participant 6	Senior Data Scientist
Participant 7	Data Scientist
Participant 8	Data Scientist
Participant 9	Data Scientist

Evaluation Questionnaires

As mentioned before, the questionnaires were used to evaluate the tool. A suitable questionnaire is chosen for each of the three questions described above. Here the three forms are explained in more detail.

Part 1: Usefulness of the explanations The first part assesses whether the explanations add value to the NLP data scientist. The Explanation Satisfaction Scale (ESS) presented by Hoffman et al. [44] is used for this. More information about this scale can be found in the background, section 2.2. This is a suitable scale to get a better understanding of how well the explainability techniques help the user achieve their goals [43]. It has also been used in several other prominent contributions, such as [106]. As the focus is on human satisfaction and not on the quality of explanations, the ESS is preferred to the System Causability Scale (SCS) by Holzinger, Carrington, and Müller [46].
In the evaluation, we must be clear to the participant that this is an isolated questionnaire and should only evaluate the usefulness of the techniques, not the tool itself.
Because the tool implements multiple explanation methods, the participants are asked to answer these questions for each method they use. Thus, this result will also include whether one method is more valuable.

Part 2: XAI in practice The second part of the evaluation questions will be a series of open-ended questions that will help us to better understand whether the current solution fits their workflow. In addition, observations suggest that the error analysis step of the workflow is most promising for useful XAI, so this will be explicitly asked about in this questionnaire. These open-ended questions will provide additional insight into the described workflow. The questions are the following:

1. Do you think the setup of having an external platform next to the Jupyter Notebook/Sagemaker is more or less usable?
2. Can this tool be easily integrated into your workflow when creating LLM-based products? Why yes/no?

3. Consider the task of error analysis and improving the prompt. Would this tool be helpful in that phase?
4. Does this tool change your opinion on XAI for LLMs? If yes, in what way?

Part 3: Tool usability We use the User Experience Questionnaire (UEQ), a well established evaluation survey by [64]. It is often used to test the usability of a tool and to determine how satisfactory the user experience is, making it a good tool for assessing the usability of the implemented tool. An additional advantage is that it is quick and gives a comprehensive impression of the usability.
The questionnaire consists of a set of 26 questions where the participant is asked to give their opinion on a contrastive 7-point scale. The 26 questions are all related to attractiveness, but consist of 6 final scales, of which attractiveness is one. The other five are divided into two qualities:

1. **Pragmatic Quality**, which means goal-oriented. This category includes efficiency, clarity and reliability.
2. **Hedonic Quality**, meaning not goal-directed. This category includes Stimulation and Novelty

The analysis of the questionnaire data follows the standard procedure presented in [107]. In order to digest the findings, the results are often compared with a benchmark. The UEQ offers such a benchmark, which contains data from 452 product evaluations with the UEQ (with a total of 20190 participants in all evaluations). The benchmark is updated once a year, [107]. A tool can be considered user-friendly if 10% of the results in the benchmark dataset are better than the tool and 75% of the results are worse than the tool.

It should be noted that data may be excluded in certain instances. This may occur when a participant has three or more critical inconsistencies, which indicates that multiple answers contradict each other. Another reason for excluding data in this context is when a participant repeatedly answers with the same number. This could be indicative of a lack of sufficient engagement with the questionnaire.

5.2. Results

This section is divided into three parts, corresponding to the three sub-questions above. Section 5.2.1 discusses the results on the usefulness of the implemented explainability techniques. Then subsection 5.2.2 presents the results on the quality of the integration of the tool into the workflow. Finally, the evaluation of the tool's UI is covered in subsection 5.2.3.

5.2.1. Usefulness XAI for LLMs

The three techniques evaluated are certainty, token-level feature importance & and contrastive explanations. For each of them, we asked the following questions:

1. From the explanation, I understand how the LLM works
2. This explanation of how the LLM works is satisfying
3. This explanation of how the LLM works has sufficient detail
4. This explanation of how the LLM works seems complete.
5. This explanation of how the LLM works tells me how to use it.
6. This explanation of how the LLM works is useful to my goals.
7. This explanation of the LLM shows me how accurate the LLM is.
8. This explanation lets me judge when I should trust and not trust the LLM

Explanation satisfaction is defined here as the degree to which users feel they sufficiently understand the AI system or process being explained to them [43]. This scale includes understandability, satisfaction, detail preferences, completeness, usefulness, accuracy and trustworthiness. The results for each technique are presented individually, followed by several general findings.

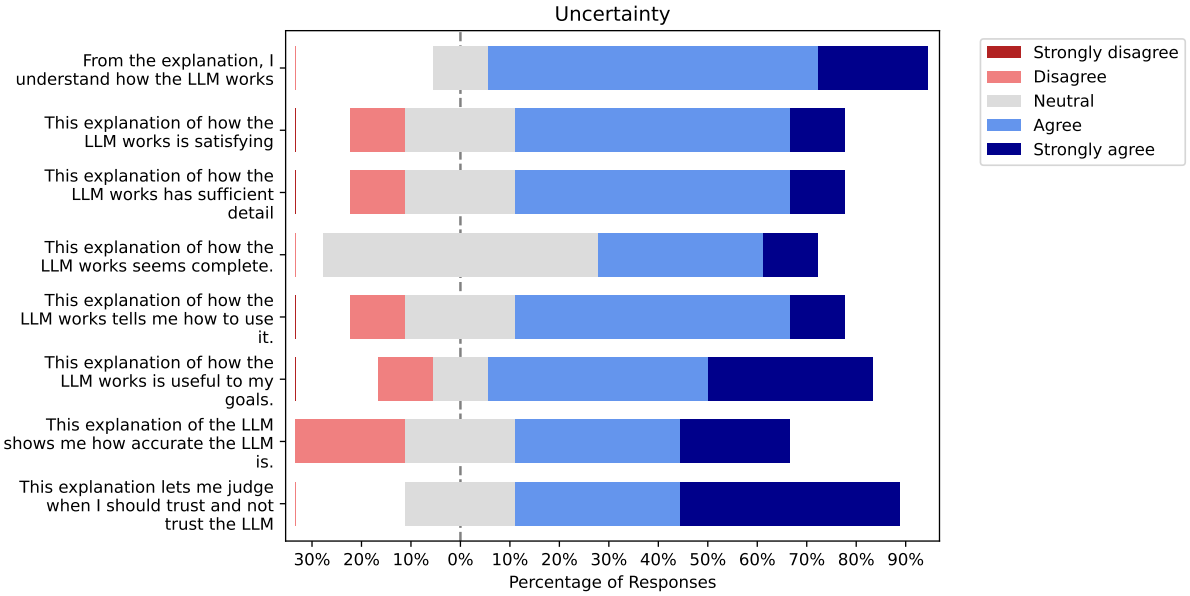


Figure 5.1: Results from the XAI Satisfaction Questionnaire for the uncertainty technique

Uncertainty

The first technique to be evaluated is **Explanation Technique 1: Model Uncertainty**. Figure 5.1 shows the results of the questions related to this technique. For each question, it shows the percentage of responses for each of the options. This is followed by an analysis of the results.

The overall sentiment of this technique is positive. The answers generally reflect a favourable view of this technique, especially regarding understanding the LLM and trust. This is understandable for trust, as it is easier to trust a model with a higher confidence. However, it can also be seen that the score on *sufficient detail* is relatively low, which was also mentioned often during the hands-on session. Here, it was mentioned several times it would be beneficial to show the percentages, instead of the colouring as it is currently.

Additionally, a lower score on completeness can be seen. This could be due to the fact it is computed on a token level, which was evaluated as not being very insightful. Participant 4 said: *"One value would be more intuitive. Suggested is an average over the individual uncertainties on output level or on sentence level."*

Another relatively negative score was on the accuracy assessment. This is also reasonable, as the uncertainty does not give any insights into the accuracy. In this light, one participant was not satisfied with this technique, saying *"It would be good to know what the problem is that it can solve."*, which is a valid concern.

Token Level Feature Importance

The second technique that will be assessed is **Explanation Technique 2: Token Level Feature Attribution**. Figure 5.2 presents the results from the questions regarding this technique.

This technique is also rated positively, especially for comprehensibility, concluding that it helps to better understand how the LLM works. This could be due to an alignment between the user's mental model and the explanation. That is, the way the method explains the reasoning of the model is consistent with the way users reason.

However, the feature importance technique seems to be incomplete. One participant mentioned here that *"Single value would have been nice. Also it would be beneficial to remove stop words before computing the value"*, concluding that some sort of aggregation should be done to make it at sentence or prompt level. This is a valid comment which could be a good further improvement to this technique.

In addition, one acknowledged that it can be useful for prompt engineering, as it provides insight into which parts are most relevant to the prompt. Here, Participant 8 mentioned: *"I think it helps identify which parts of the question influence the model's answer and which tokes attribute to it."*

However, the participant noted that this is highly dependent on the use case and the problem at hand.

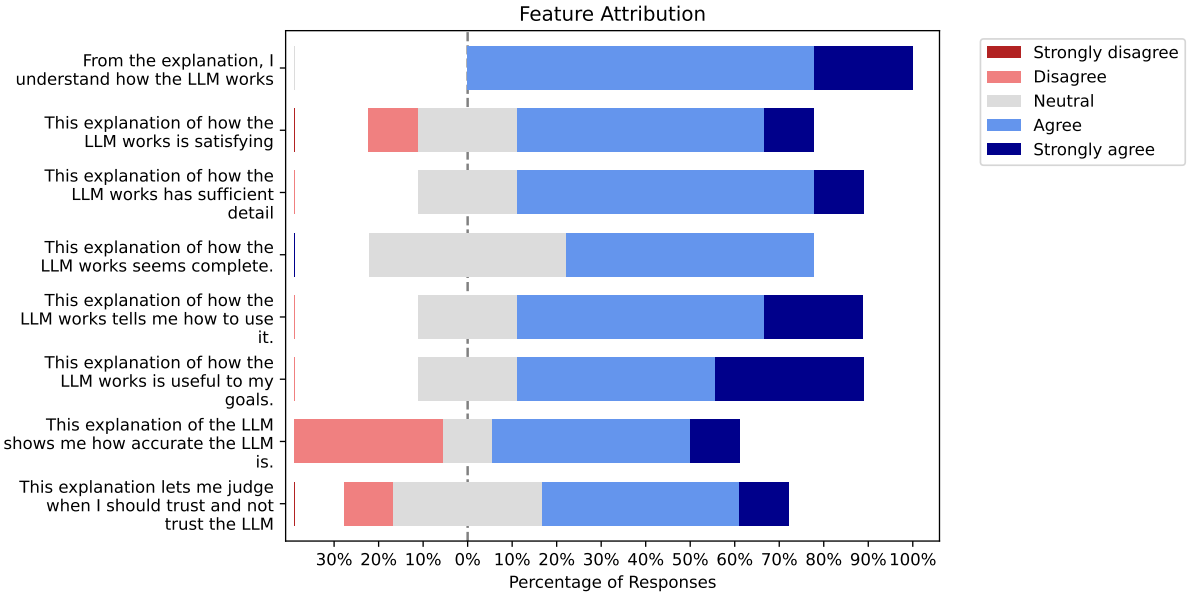


Figure 5.2: Results from the XAI Satisfaction Questionnaire for the Feature Attribution

More research could be done on the specific cases where this particular method is most useful. Finally, this technique is not useful for assessing the accuracy of the model and improving confidence. This is reasonable as it does not provide metrics to support these goals.

Contrastive explanations

The last technique in this part of the evaluation is **Explanation Technique 3: Contrastive Explanations**. Figure 5.3 presents the results from the questions regarding this technique.

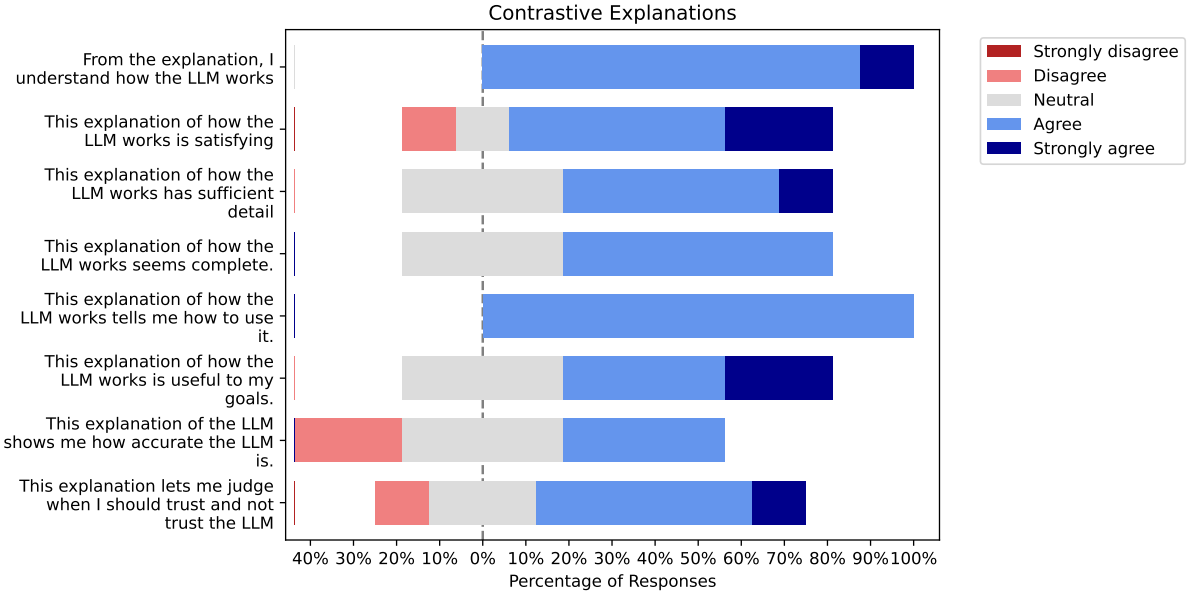


Figure 5.3: Results from the XAI Satisfaction Questionnaire for the Contrastive Explanations

The contrastive explanations were also rated positively, especially those explaining how the LLM works and how to use it. During the evaluation sessions, most participants were also generally positive about this explanation, as it is currently only implemented for the first generated token. This was often not the token on which the participant wanted to compute the contrastive explanation. This could also be the reason for the lower satisfaction score. They confirmed the potential of the technique if it could be done

for other tokens as well.
In the current implementation, the contrastive token has to be given manually. However, it was mentioned that it would be beneficial to let the model decide the token, for example by calculating it for the top_k alternatives.
Finally, Figure 5.3 notes that this explanation did not contribute to the assessment of the model's accuracy.

Comparative Analysis

Some general observations can be concluded from the data and hands-on guided demos that have been performed. These are regarding accuracy, understanding, level of details and dependency on the use case.

Accuracy assessment One conclusion from the results presented above is that all three explanations do not help the NLP data scientist assess the accuracy of the model. The results were relatively negative for all three explanations, as shown in Figure 5.4. This observation is unfortunate, as one of the challenges identified in subsection 3.2.2 was the evaluation of the LLM. Here the assessment of accuracy plays an important role. It could be concluded that explanations or quantitative metrics are needed to fully assess the accuracy and performance of the LLM.

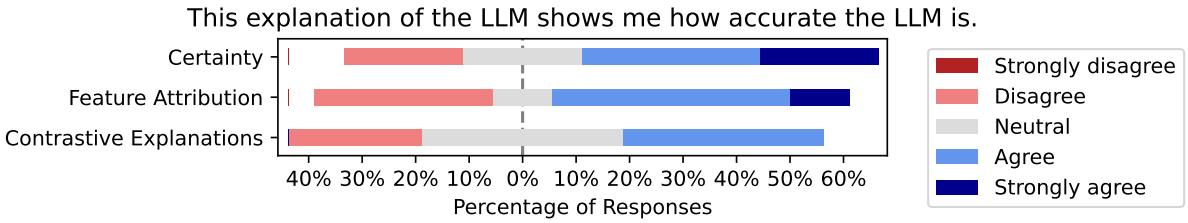


Figure 5.4: Comparison on the question on accuracy assessment

Useful for understanding how the LLM works All three explanations were rated positively in terms of the informativeness of the LLM work, as shown in Figure 5.5. This is interesting because it does not show internal mechanisms such as attentional mechanisms. Obviously, this is unnecessary and post-hoc calculated explanations are useful for a better understanding of the LLM.

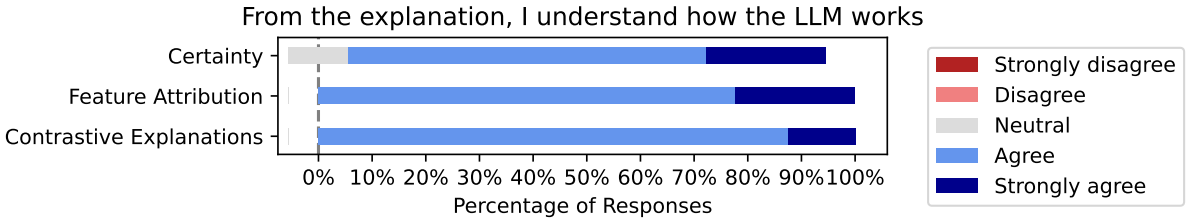


Figure 5.5: Comparison on the question on understanding

Satisfaction One of the important parts of this evaluation is the users' satisfaction. The data presented in Figure 5.6. While a few participants are neutral or disagreeing, most are agreeing that working with the explanations was satisfying. This could be because of the interactive features implemented, it indeed gave additional insights or have other reasons.

Level of detail For the uncertainty and feature attribution, it was often mentioned that the level of detail was too low. Especially with the certainty, showing the percentages was often requested. Additionally, it was requested what the feature attribution numbers were, which was also not shown in the UI. The data presented in Figure 5.7, which softly confirms this.

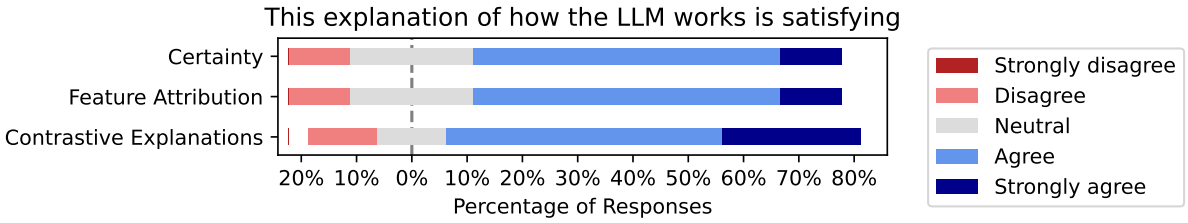


Figure 5.6: Comparison on the question of sufficient details

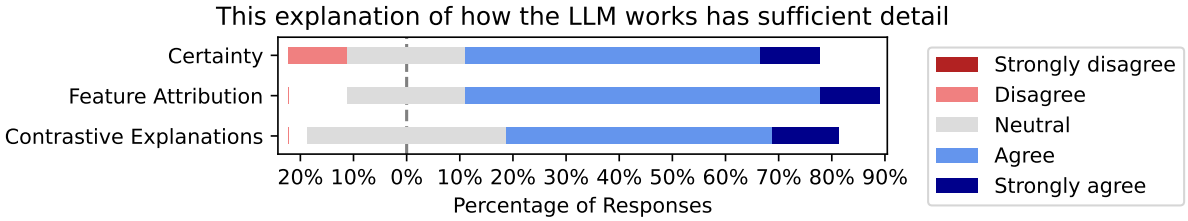


Figure 5.7: Comparison on the question of sufficient details

Use-case dependency Several participants mentioned that the usefulness of the XAI was highly dependent on the use case. For the fairness assessment use case, such as the demo dataset, it was confirmed to be useful in several ways. For example, one participant stated that it would also be useful for classification and other classic NLP tasks. However, it cannot be concluded that these explanations are helpful in other use cases. More research needs to be done on other generative tasks, such as chatbots or RAG-based systems.

Conclusions on user satisfaction

It can be concluded that the explanations satisfy the user in understanding how the model works. All three provided insights helpful for that particular purpose. However, for understanding the accuracy of the model or evaluating the model, all explanations were not sufficient. Here, the use-case, level of detail and visualisation technique play a role.
In general, Token-level Feature attribution has the most potential, as it was proven to be useful towards the goal of the user.

5.2.2. Integrating HELMET into the existing workflow

This section presents the results from the second part of the questionnaire and answers the question whether the current implementation fits the workflow of the user, which is part of SQ4.
The overall response to the HELMET tool was very positive. During the hands-on demonstration, all participants expressed satisfaction with the setup and were able to rapidly comprehend the workflow. 100% of the participants agreed that the tool integrated seamlessly with their existing workflow and was straightforward to set up. Furthermore, 78% recognised that it facilitated troubleshooting and rapid engineering.

External platform setup

Having an external web application was rated very positively. It was acknowledged that the modular setting helped them to keep the Jupyter Notebook clean and still have all the visualisations they needed. To quote P1: *"The web application helps a lot with the analysis and gives more freedom than doing the visualisations inside the notebook"*. In addition, it was nice because the web application could be easily shared with non-technical colleagues and still give them a good overview of the runs without having to set up a Jupyter notebook. This was mentioned by P5 who said: *"It is very useful and it helps to interact more freely with the results without the clutter of the code, and it makes it easier and "cleaner" to provide explanations to project partners."*
In addition, it was felt that the capabilities of the Python package and the web application were clear. Most found it intuitive to use both parts of HELMET and to switch between them.

Integration

The integration was also evaluated positively, were 78% was positive. One big advantage mentioned was how easy it was to install the HELMET Python package. Furthermore, participants commented on the choice of creating a wrapper around the Huggingface platform as a good choice, as this is often used and will make it easier to use and configure LLMs from Huggingface. To cite P7: *"The connection between the model invocation and webapp is straightforward to set up. The webapp is intuitive to use."*

Error Analysis

Error analysis was identified as one of the challenges of the NLP Data Scientist and was therefore specifically asked about in the evaluation questionnaire. 8 of the 9 participants confirmed that it is helpful to have a platform to dive into the different explanations. In this context, participant 2 mentioned *"for the trial and error phase when the user optimises his prompt"*. Only one participant was not convinced of the usefulness of these features in the tool for error analysis and prompt improvement.

Change the opinion on explainability

There were mixed feelings about how XAI could benefit them. Some participants admitted that they were now more aware of the possibilities of integrating XAI into their LLM-based solutions. In contrast, others mentioned that they were not convinced that this tool would help them significantly. They mentioned that the tool is a nice package, but their opinion did not change after using it. Finally, one participant mentioned that HELMET makes the explainability clearer and more defined, but also shows how much work still needs to be done in the area of XAI to better understand LLMs.

Improvements

The final question on this evaluation form was about future improvements. Several were mentioned. One interesting suggestion was to create *"insights & suggestions"* in the details of the input-output analysis. This could present the user with an LLM-generated passage with some useful suggestions based on the interpretation of the explanation.
Secondly, support for multi-modal was an interesting suggestion. It could be very interesting to have this implemented as well and to see how a saliency map over an image could be integrated into this tool.
Finally, there was a request for some level of organisation. This means that runs could be grouped into experiments, which could be grouped into a project. This would give more structure to the long list of runs that are currently implemented.

5.2.3. User Experience

This section presents the findings from the User Experience Questionnaire (UEQ). The UEQ is meant as a measure of user-friendliness through six scales and 26 statements. The six scales are *attractiveness*, *perspicuity*, *efficiency*, *dependability*, *stimulation* & *novelty*, where attractiveness is its own dimension. The other five are categorised in one of two aspects, which are *pragmatic* & *hedonic*. *Pragmatic Quality*, focussing on goal-directed attributes includes efficiency, perspicuity & dependability. Secondly, *Hedonic Quality*, meaning not goal-directed, includes stimulation & novelty. The results will now be presented.

One participant had three critical inconsistencies. Moreover, this participant was close to the critical limit of repeated answers. Thus, this data point is excluded from the results, leaving eight participants left in this analysis.

Consistency in scales

Before analysing the results, it is necessary to assess the reliability of the scales. This can be achieved by computing the Cronbach's Alpha-Coefficient, which is a measure of the consistency of a scale. There is no universally accepted rule regarding the target value, however a reasonable target for this coefficient is >0.65, which will be used for this study. Table 1 presents the scales and associated values. It can be concluded from this that all scales are reliable, with the exception of the dependability scale. Consequently, this scale may be interpreted incorrectly by multiple participants, a point that will be discussed in the results section.

Results

Figure 5.8 presents the overall results based on the six scales. The black diamonds present the mean of each scale. The bars are the benchmarks which the results can be compared with. Based on this, an impression of the attractiveness of the UI can be concluded.

Table 5.2: The Alpha-Coefficient for each scale

Scale	Alpha
Attractiveness	0.92
Perspicuity	0.70
Efficiency	0.91
Dependability	0.03
Stimulation	0.92
Novelty	0.89

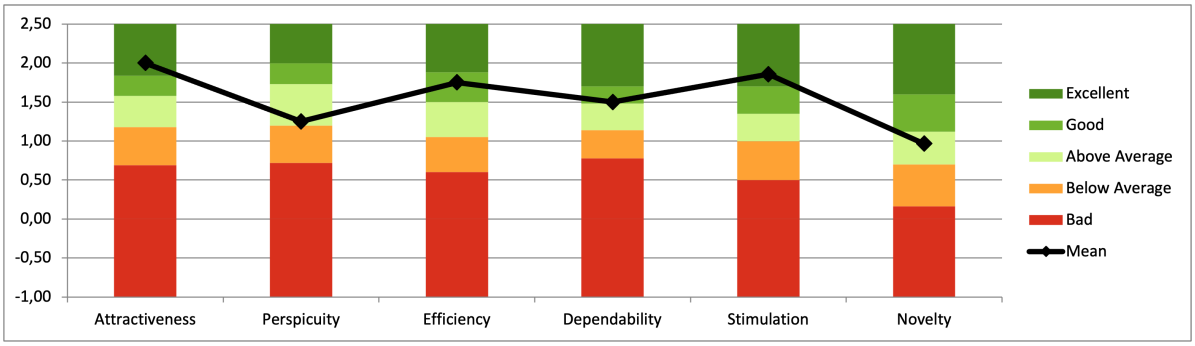


Figure 5.8: Results of UEQ compared to the benchmark

Analysis

From Figure 5.8, it can be concluded that the overall score is quite good. Often, a new product is defined as sufficiently user-friendly when it scores at least in the *good* category in all scales. From the results, it can be concluded that it scored good or excellent in *attractiveness*, *efficiency* and *stimulation*
There is room left for improvement in three categories. *Perspicuity* and *Novelty* have the lowest score, being *above average*. Furthermore, *dependability* does not score high in the *Good*, which could, therefore, also be improved. More investigation was conducted into the data to see potential reasons for these outcomes.

Attractiveness To start with the positive scales, attractiveness has been evaluated as excellent. This aligns with the participants' initial responses during the evaluation interviews, which suggest that the users found the interface visually appealing. Looking at the individual parts that make up the attractiveness, it could be seen that the actual attractiveness could be improved but that is scored well on friendliness and pleasantness dimension.

Perspicuity A lower rating was assigned to perspicuity, which is defined as the difficulty level in becoming familiar with the product. Upon further examination of the responses, it became evident that the lowest score was attributed to the statement "difficult to learn." Additionally, during the hands-on evaluation, it was mentioned on multiple occasions that the tool had a steep learning curve.
Unfortunately, it is not possible to determine which aspect of the tool is causing the learning difficulties. Since the workflow and integration are evaluated positively, as mentioned in subsection 5.2.2, it could be the UI is not intuitive at first sight. This could be because interpreting the XAI is complex or the app is challenging to learn. In all cases, further improvements should be made to improve the perspicuity.

Novelty In terms of novelty, the tool is scored high on being conventional. This can be expected because parts of the user interface were inspired by other web applications. Furthermore, the focus of this tool was not to be novel but to make it usable for new users. Therefore, its conventionality can be seen as a positive attribute.

Dependability First, it should be noted that this scale's reliability did not score high. Therefore, the results might not be representative of the participants' opinions.
Diving into the results, it was noticed that the UI performed relatively well on the how supportive the tool was. Additionally, it was meeting the expectations of the participants. It score a bit less on the predictable side, which could mean it was not always clear what steps are the tool could perform.

5.3. Conclusions

It can be concluded that HELMET definitely has potential. **The tool provides multiple explanation techniques to the NLP Data Scientist,** mainly to gain insights into how the Language Model works and how to use it.
Nevertheless, it remains to be proven that these explanations are indeed useful. The results indicate that they are not as effective as desired in terms of assessing accuracy and usefulness in achieving their intended goal. In order for the explanations to be fully useful, they must be more helpful in assessing the model's accuracy. **The explanations provided insights into the model's behaviour**, which could lead to a more accurate model.
When better explanations exist and are implemented, HELMET has proven to be a usable tool. The tool can be easily integrated into the workflow because of its modular architecture and easy installation procedure using Pip. **Using the two parts of the tool was evaluated as giving them better structure within the Jupyter Notebook without sacrificing visualisation possibilities**. While more details could be given within the explanations, the separation of the Python package and web application was positively reflected.
The UI, in general, was user-friendly considering the results of the User Experience Questionnaire (UEQ). However, it should be acknowledged that the tool has a steep learning curve, which makes it less useful for novice users, particularly those with a less technical background. It is recommended that greater attention be given to making HELMET less difficult to learn, by making the UI more intuitive and providing more information on how to interpret the explainability techniques.

6 Discussion

This chapter will reflect on the key findings from the observations and evaluation. From this, implications can be derived for different research areas, including eXplainable Artificial Intelligence (XAI), Human-Centered eXplainable AI (HCXAI) and XAI techniques for LLMs. The findings integrate the observations, requirements, design, implementation, and evaluation of the tool to better understand XAI's practical utility for the NLP Data Scientist.
First, the findings and implications are presented. This is followed by the limitations, which provide a critical assessment of the methodology and results. From this, future research is recommended, highlighting areas that can be further investigated.

6.1. Findings & Implications

Based on the observations and evaluation results, several important findings can be derived. These will be presented here by answering the research question and some implications that can be derived from it.

SQ1: What specific needs and challenges do NLP data scientists face when developing LLM-based solutions?

The formative study identified several general points. In practice, there is a significant difference between the data scientist and the NLP data scientist working with Large Language Models (LLMs). Their workflow is quite different and involves different steps. Feature engineering is less important, while the task of prompt engineering becomes quite essential.
LLMs are used for many applications, including entity/word extraction, summarisation and RAG applications. Here accuracy is the primary goal for the user.
In addition to accuracy, another requirement is efficiency. This is a significant desideratum, especially in model selection, prompt engineering and model tuning. This could be addressed by well-designed tools to help set up different models and metrics to help compare input-output combinations.
In addition, the importance of trust is lower than found in the literature: Trust is not a big issue for the NLP data scientist creating solutions to classical NLP tasks. However, trust plays a role in generative tasks such as chatbots.

Challenges Several challenges have been identified in this study. The first is actionable error analysis, where the user needs to understand why a particular error was made by the model. This can be challenging due to the complexity and black-box nature of these models.
What has also been discussed is prompting. This is timely as it is often difficult to know why the model is failing at that particular prompt. In addition, a small change in the input can have a large impact on the output, making it difficult to predict the output.
Another challenge was evaluation. There are many different evaluation techniques available within NLP. However, these metrics and strategies are not always consistent with the business case.
Finally, it was discussed that it is quite difficult to communicate these results to stakeholders. Reproducing the results and presenting them correctly to stakeholders was mentioned as a need.

Potential of explainability From the formative study and evaluation, it was found that explainability has potential benefits. Explanations can help better understand the model and, therefore, help with prompt engineering or error analysis. Although these benefits were recognised in this study, the current explainability techniques did not meet the above mentioned needs. Explanations should provide more insight into the limitations of the model and how parts of the pipeline should be improved.

Implications These findings can be used in a number of ways. Firstly, new tools using open-source models should consider these needs and challenges when creating an LLM-based tool, not just when using explainability techniques. Addressing these needs will greatly benefit the tool's usefulness and, therefore, user satisfaction.
Second, vendors of closed-source models should adapt their tools to support the challenges. As found in the formative study, there is a trade-off between open-source and closed-source models, with controllability and opacity being a major disadvantage for closed-source models. However, these models score higher in terms of performance and usability. Therefore, model vendors should include explainability techniques and other supporting techniques in their output. This will solve part of the trade-off and help the model user with guidance and error analysis.

SQ2: What are the explainability and functional requirements for a tool to support these needs?

In terms of explainability needs, there is a need to elucidate model behaviour in order to assess performance. Although explainability is not currently used by the practitioners interviewed in this study, there is potential.
The most important explainability requirements are interpretability and explanatory power. Interpretability will help to gain insight into how the prompt should be changed and how the model arrived at its current output. Secondly, explanatory power is important because it was found in this study that the low level of detail was rated negatively. This study found that the more complete the explanations, the better.
From the chosen explanation techniques, it could be seen that the participants were positive about all three: uncertainty, feature importance and contrastive explanations. However, this research did not prove that it was worth adding this extra layer of complexity by hosting an open source model and using tools such as `HELMET`.
All of the available techniques, such as perturbation, CoT prompting, global explanations or mechanistic interpretability, were not fully suited to the user's needs. More research should be done on how to make Transformer models more explainable, especially for the needs of the NLP data scientist.
Finally, fidelity was considered important, but it is unclear whether this was helpful. More research should be done to confirm this particular need.

Tool Requirements For any new tool, several requirements will improve the usability of the tool. Firstly, ensuring that it can work with all open source models was considered positive. It should be easy to configure which model to use and to add custom parameters such as temperature. The tool should also be able to compare and evaluate input-output combinations based on user-defined metrics.
Second, the tool should be modular. The combination of a Python environment was found to be consistent with the NLP data scientist's working environment, but not the most appropriate for the visualisations. An external platform where the results can be examined is recommended here.
Third, it should be clear how the tool works. This should apply to the features, by providing good documentation and examples. But it should also be clear how to interpret the explanations. This could be done in the UI of the tool, as is done in `HELMET`, but it could also be done in other places.
Another important requirement is to solve the reproducibility problem. Reproducing results within LLMs is considered difficult, so a tool should include the ability to save the logs and parameters to be able to come back to these results later. This was also rated very positively.
Finally, the usability of the tool is very important. As well as being easy to set up with custom models, this should also mean that the user interface is easy to use. This can be achieved by ensuring that the UI is clear and that the state of the system is visible. Tooltips can also be helpful here.

SQ3: Based on the requirements, how should the explainability tool look like?

Visualising explainability techniques is difficult, mainly because the explanation may still be difficult to interpret. For example, token highlighting can be considered difficult for humans to interpret. The evaluation showed that it helped a little to understand the model, but it still raised a lot of questions about what it then actually meant. This was also the case for the certainty visualisation, which raised questions about what it means when the model is uncertain. Creating interpretable explanations is therefore still an unresolved area.

Tool design implications The tool should incorporate the modular structure described above. This could be done by creating a separate web application to interact with the outputs and explanations.
For the model usage, this could be done by creating a wrapper around the `Transformers` library. This has been very well received and will give a lot of implementation freedom, making it a great option for calculating the explanations. `HELMET` and other tools use the features provided by this library, so it is a good candidate for such a tool.
The web application should be able to be hosted online or locally. This was seen as positive as it made it easier to collaborate and share results with others. Multiple levels of organisation will make it easier to group users, experiment, and run them together.
It is recommended that logs of most actions be kept. This will greatly improve the reproducibility of results and help with the interactivity of the UI and the details page for each run.
A conventional layout is recommended for the User Interface (UI), making it easier to understand all the functionalities and to navigate through the application. Another factor is the use of the Direct Manipulation Interface, where it is very clear what each button does and what can be expected after the action is performed.

SQ4: How can the tool be implemented to integrate seamlessly into the workflow of the NLP Data Scientist?

The current workflow of the NLP Data Scientist has been described in this research. After the use case and data are available, the experimentation phase starts, where several hypotheses are created. Based on this, solutions are explored and an evaluation pipeline is created. This will help evaluate the different solutions using the same metrics. The most likely solutions are then implemented. This includes the selection of the model, possible fine-tuning of the model, setting the hyper-parameters and immediate engineering. Finally, the model is evaluated, from which new actions should be derived.

Transformers Library Most open-source tools are accessed through Huggingface using the Transformers library. For any LLM-based tool to integrate well, it should therefore be based on this library. In addition, the Python Jupyter notebooks are widely used, making the transformers library an appropriate choice. Finally, having a Python package that can be installed via Pypi will also make the tool more useful.

Python & JavaScript combination Creating a separate application for model use and explanation interactions made it easier to implement and use. It created two distinct focuses, depending on the stage of implementation. When models were being used, HELMET provided the tools needed to configure, use and evaluate the model. When error analysis began, the web application could be set up to dive deeper into potential errors and solutions.

Use `HELMET` as inspiration It can be concluded that the setup of the proof-of-concept created in this study is fitting the NLP Data Scientists' workflow. While limited explainability techniques can be computed in the current implementation, the tool's architecture does make it possible to use most of the features of the web application. New visualisation techniques and features in the web application could be implemented, making it an excellent tool for developing new techniques. It can be integrated with other tools and resources, making it attractive to use.

6.2. Limitations

It is challenging to research explainability for LLMs in a practical setting. While several measures have been taken to create reliable and insightful results, limitations still exist. This section will state several limitations in the methodology and findings of this thesis.

6.2.1. Methodological Limitations

While much time was invested in creating a well-designed methodology, the current approach has several limitations. This section enumerates these limitations and presents how they were minimised or mitigated as much as possible.

Biased evaluation Because of the way the evaluation was set up, multiple biases could arise. First, using the Likert scale is often susceptible to positive response biases by social desirability or acquiescent responding [62]. This can compromise parts of this research in terms of fairness and validity. Additionally, it should be acknowledged that all participants from the evaluation were aware of the background of the thesis and thus might have given the preferred answer instead of the honest answer. This is avoided as much as possible by stressing that they should answer genuinely. In addition, all participants were deliberately not updated on some parts of the development progress to further limit the risk of telling only what the researcher wanted to hear.

Quantitative measures for evaluation The evaluation utilised qualitative measures, providing valuable insights, but also introduced a limitation regarding the evidence. The evaluation proved explainability could be useful. However, this was based on self-evaluation, not using quantitative measures. More investigation is needed to get quantitative measures of the tool's effectiveness, such as improved model performance or efficiently getting the correct result. Future evaluations should compare the quantitative measures against a baseline to understand the possible gain of explainability fully.

Limitations of UEQ The User Experience Questionnaire (UEQ) provides insights into which aspects of the User Interface (UI) could be improved. However, it does not give a very clear direction on *what* to solve concretely and *how* to achieve this. This thesis did not ask for an important measure, such as the Importance-Performance Analysis (IPA) [42]. It might have been interesting to include one to get a clearer picture of which scales are most important to improve.

6.2.2. Research limitations

The results of this study must be seen in the light of some limitations, which will now be presented.

Assumption of fidelity One of the desired properties of explainability, described in section 4.1, describes the importance of fidelity. One might ask how far this applies to LLMs. For example, Sun et al. [115] showed how the attention mechanism and CoT prompting were useful to the user in the context of writing code. Repeating the research without this assumption might yield different and possibly better results.

Generalisability In the formative study, only a limited number of practitioners were interviewed. This limits the generalisability of this research in terms of its observations and conclusions, especially given that NLP data scientists all work quite differently. This is acknowledged and taken into account by supplementing the observations with research and literature by others.In addition, the in-depth interview for the observations provided saturated insights as it was.
It should also be noted that this thesis was conducted within a single company, mostly located in North Europe. While this provided a controlled environment for the experiment, it may limit its applicability to other companies. Data science practices are highly context dependent and can vary significantly between organisations. It also depends on factors such as the size of the organisation and the scope of the NLP data scientist's responsibilities.

6.3. Future work

This section outlines several potential future research directions, including explainability techniques and recommended improvements to the current implementation of HELMET.

New explainability technique to improve model accuracy An interesting avenue of research could be to develop a technique that provides insight into model accuracy beyond the current evaluation metrics. Current explainability techniques do not provide enough information to improve model performance and do not help to assess accuracy. Such new explanation techniques should focus on creating causal relationships, which makes it easier to understand how the input and output are related. Inspiration can be drawn from Ras, Gerven, and Haselager [102]

XAI better with better performing models Heuristically, it was found that better performing models also made the XAI more meaningful. Within this thesis, the largest model used was 7B parameters. It would be very interesting to see what 70B or even larger would do. If the XAI would be more useful because it makes more sense to us/is closer to our reasoning.

Quantify the increased performance Another interesting research topic would be whether the current implemented explainability techniques or other techniques can help increase the model's performance. This should be done by first setting a baseline and then asking participants to improve the quality of the output using XAI. By comparing the results to the baseline, a quantitative measure of the performance increase can be obtained.

Use-case dependent XAI *"It depends"* was mentioned often during the evaluation. The potential of explainability is acknowledged; the techniques were fairly useful, but they are still very much dependent on the use case. More research on XAI for LLMs in different NLP tasks should be done, just like the scenario-based approach by Wolf [130]. Acknowledged that it is situation-specific is done by Chazette and Schneider [16]

Mental Model Reasoning To deepen one's understanding of the persona, one can investigate the user's mental model, sometimes called the theory of mind [83]. Understanding mental reasoning makes it easier to create explanations that fit their reasoning. By studying their reasoning in more detail, better explanations could be added to tools like HELMET.

Extending the implementation with closed source models To some extent, the tool could be extended to work with closed-source models. An architecture is proposed in Figure 6.1, which would make it easy to extend the tool. This would make the tool even more useful, especially when making comparisons between particular models. Of course, the vendor should implement the explanations in order to use them.

Implementation improvements Multiple improvements could be made to the implementation of the tool. Here, a couple of suggestions are given, drawn from the evaluation.
First, an improvement for the contrastive explanations. This uses a reference token, which should now be chosen manually. Some more information could be presented here, or even further, the token should be suggested based on alternative words or AI suggestions. Regarding the other explanations, it would also be better to have a single value for uncertainty and feature attribution. This would improve the usefulness of the explanation.
Secondly, more built-in metrics should be implemented. This will make it easier to use the tool to evaluate the output of the model.
Regarding the web applications, some additional improvements are suggested now. First, it would be helpful to customise the level of detail that is presented. This will help to give the user to be able to dive deeper into the explanation when needed. Additionally, it would be beneficial to be able to compare more than two runs at the same time, as was suggested during the evaluation.

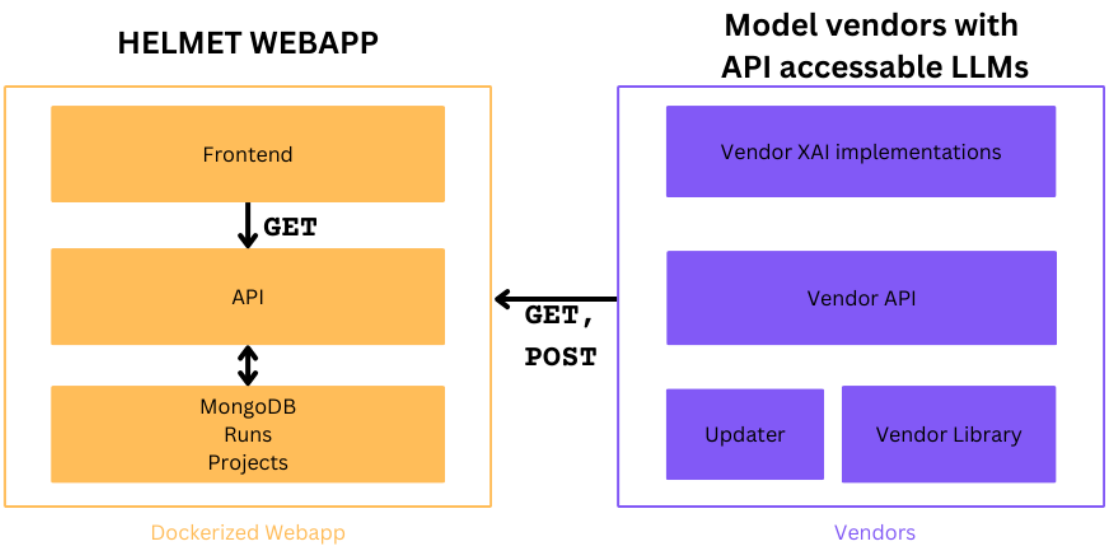


Figure 6.1: Proposed Architecture with extension of Closed-Source models

Lastly, the steep learning curve is one part of `HELMET` that needs further improvement. During the evaluation, it was concluded that the tool is difficult to learn, making it score low on the functional requirements. Improvements should be made in this area to make it easier to start using the tool.

7 Conclusion

This thesis aimed to develop and evaluate an explainability tool to assist users' needs. This was done by conducting a formative study followed by building requirements and a design. Then, a proof of concept was built, which was used to evaluate multiple parts of the tool, giving additional insights into the NLP Data Scientist and its need for explainability.

The first step was to narrow down the user, which was decided to be the NLP data scientist working in industry with Large Language Models (LLMs). A formative study was carried out to investigate their needs, revealing user desiderata, their workflow and different perspectives on XAI. It was concluded that NLP Data Scientists overlap with general Data Scientists; however, more focus was placed on prompt engineers and less on feature engineers. In addition, the NLP Data Scientist did not use explainability, which contrasts with some uses in other Data Scientist domains.

In addition, the formative study identified several areas where explainability could help with parts of the development of LLM-based solutions. The potential benefits were to help with prompt engineering or error analysis or do a better quality assessment. If explainability could help in these areas, it could be seen as a major advantage over the closed-source models that are currently more widely used. This decision is currently based on performance and cost, but explainability could be added as an additional factor if the techniques provide sufficient insight.

Based on the observations and literature from the formative study, requirements could be defined. These were divided into explainability requirements, formulated as desired characteristics, and tool requirements. The requirements helped to create a usable tool that was aligned with the user's workflow, while the explainability requirements were aligned with the challenges. The most important explainability requirements were interactivity, explanatory power and interpretability. Fidelity (also called faithfulness) was considered as a desired property, but it turned out not to be an important requirement for the user.

A design was then created, including an architectural design and a UI. Based on the desired properties of explainability, it was decided which explainability techniques should be implemented. Three techniques were implemented: uncertainty, token-level feature attribution and contrastive explanations. Uncertainty was visualised by creating three categories into which a given token falls, highlighted by different colours. In addition, the architectural design proved to be successful, as the modular system of the Python package and the web application were positively evaluated.

The design was then implemented as a proof of concept, which could be used to evaluate the requirements and design decisions. The proof of concept, branded as a Human-Evaluated large Language Model Explainability Tool (HELMET), gave more insights into the needs of the NLP data Scientist and how they work.

`HELMET` was evaluated using guided hands-on sessions, after which participants reflected on the usefulness of the explanations, the degree of potential integration into their workflow, and the usability of the interface. The results showed that the implemented explainability techniques were useful in some cases, but did not fully satisfy their primary need for improved accuracy. It was also concluded that the explanations were difficult for the user to interpret as they did not mean much to them. The tool was found to fit well into their workflow due to its modular design and easy to install Python package.

The contribution of this thesis is threefold: A human-centred approach is taken to define and understand the NLP data scientist, thereby contributing to the HCI community. Secondly, a well-considered list of requirements and design has been made, including desired properties for explainability. Finally,

evaluation through creation has been used, resulting in an implemented open source tool for others to use, as well as insightful data on what needs to be improved in further research.
Future directions include new explainability techniques that follow user needs and adhere to the desired properties of explainability. Secondly, using the helpful feedback from the evaluation, future improvements to the implementation could make HELMET an even better product.
In conclusion, HELMET demonstrates the potential of explainability techniques for LLMs and provides valuable insights and tools for NLP data scientists. While there are areas for improvement, the foundation laid by this research provides a robust design for developing more effective and user-friendly explainability tools in the future.

Bibliography

[1] Marah Abdin et al. *Phi-3 technical report: a highly capable language model locally on your phone*. en. arXiv:2404.14219 [cs]. Apr. 2024. DOI: 10.48550/arXiv.2404.14219. URL: http://arxiv.org/abs/2404.14219 (visited on 05/14/2024).

[2] Ashraf Abdul, Jo Vermeulen, Danding Wang, Brian Y. Lim, and Mohan Kankanhalli. "Trends and Trajectories for Explainable, Accountable and Intelligible Systems: An HCI Research Agenda". en. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. Montreal QC Canada: ACM, Apr. 2018, pp. 1–18. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3174156. URL: https://dl.acm.org/doi/10.1145/3173574.3174156 (visited on 09/28/2023).

[3] Amina Adadi and Mohammed Berrada. "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)". In: *IEEE Access* 6 (2018). Conference Name: IEEE Access, pp. 52138–52160. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2870052. URL: https://ieeexplore.ieee.org/document/8466590 (visited on 09/28/2023).

[4] AI@Meta. "Llama 3 model card". In: (2024). URL: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

[5] Qurat Ul Ain, Mohamed Anime Chati, Mouadh Guesmi, and Shoeb Joarder. "A multi-dimensional conceptualization framework for personalized explanations in recommender systems 11-23". In: *IUI workshops*. 2022. URL: https://api.semanticscholar.org/CorpusID:248301731.

[6] J Alammar. "Ecco: An Open Source Library for the Explainability of Transformer Language Models". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*. Ed. by Heng Ji, Jong C. Park, and Rui Xia. Online: Association for Computational Linguistics, Aug. 2021, pp. 249–257. DOI: 10.18653/v1/2021.acl-demo.30. URL: https://aclanthology.org/2021.acl-demo.30 (visited on 11/02/2023).

[7] Vijay Arya et al. "One Explanation Does Not Fit All: A Toolkit And Taxonomy Of AI Explainability Techniques". en-US. In: Oct. 2021. URL: https://research.ibm.com/publications/one-explanation-does-not-fit-all-a-toolkit-and-taxonomy-of-ai-explainability-techniques (visited on 11/22/2023).

[8] Narges Ashtari et al. "From Discovery to Adoption: Understanding the ML Practitioners' Interpretability Journey". In: *Proceedings of the 2023 ACM Designing Interactive Systems Conference*. DIS '23. New York, NY, USA: Association for Computing Machinery, July 2023, pp. 2304–2325. ISBN: 978-1-4503-9893-0. DOI: 10.1145/3563657.3596046. URL: https://dl.acm.org/doi/10.1145/3563657.3596046 (visited on 02/22/2024).

[9] Giuseppe Attanasio, Eliana Pastor, Chiara Di Bonaventura, and Debora Nozza. "ferret: a Framework for Benchmarking Explainers on Transformers". In: *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. Ed. by Danilo Croce and Luca Soldaini. Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 256–266. DOI: 10.18653/v1/2023.eacl-demo.29. URL: https://aclanthology.org/2023.eacl-demo.29 (visited on 01/02/2024).

[10] Oren Barkan et al. "Grad-SAM: Explaining Transformers via Gradient Self-Attention Maps". In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. CIKM '21. New York, NY, USA: Association for Computing Machinery, Oct. 2021, pp. 2882–2887. ISBN: 978-1-4503-8446-9. DOI: 10.1145/3459637.3482126. URL: https://doi.org/10.1145/3459637.3482126 (visited on 10/10/2023).

[11] Alejandro Barredo Arrieta et al. “Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI”. In: *Information Fusion* 58 (June 2020), pp. 82–115. ISSN: 1566-2535. DOI: 10.1016/j.inffus.2019.12.012. URL: https://www.sciencedirect.com/science/article/pii/S1566253519308103 (visited on 11/06/2023).

[12] Jasmijn Bastings and Katja Filippova. “The elephant in the interpretability room: Why use attention as explanation when we have saliency methods?” In: *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. Online: Association for Computational Linguistics, Nov. 2020, pp. 149–155. DOI: 10.18653/v1/2020.blackboxnlp-1.14. URL: https://aclanthology.org/2020.blackboxnlp-1.14 (visited on 10/04/2023).

[13] Virginia Braun and Victoria Clarke. *Thematic analysis: a practical guide*. OCLC: on1247204005. London ; Thousand Oaks, California: SAGE, 2022. ISBN: 978-1-4739-5324-6.

[14] Sébastien Bubeck et al. *Sparks of artificial general intelligence: early experiments with GPT-4*. en. arXiv:2303.12712 [cs]. Apr. 2023. DOI: 10.48550/arXiv.2303.12712. URL: http://arxiv.org/abs/2303.12712 (visited on 11/02/2023).

[15] Diogo V. Carvalho, Eduardo M. Pereira, and Jaime S. Cardoso. “Machine Learning Interpretability: A Survey on Methods and Metrics”. en. In: *Electronics* 8.8 (Aug. 2019). Number: 8 Publisher: Multidisciplinary Digital Publishing Institute, p. 832. ISSN: 2079-9292. DOI: 10.3390/electronics8080832. URL: https://www.mdpi.com/2079-9292/8/8/832 (visited on 11/21/2023).

[16] Larissa Chazette and Kurt Schneider. “Explainability as a non-functional requirement: challenges and recommendations”. en. In: *Requirements Engineering* 25.4 (Dec. 2020), pp. 493–514. ISSN: 1432-010X. DOI: 10.1007/s00766-020-00333-1. URL: https://doi.org/10.1007/s00766-020-00333-1 (visited on 11/20/2023).

[17] Hila Chefer, Shir Gur, and Lior Wolf. “Transformer Interpretability Beyond Attention Visualization”. en. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Nashville, TN, USA: IEEE, June 2021, pp. 782–791. ISBN: 978-1-66544-509-2. DOI: 10.1109/CVPR46437.2021.00084. URL: https://ieeexplore.ieee.org/document/9577970/ (visited on 09/19/2023).

[18] Zixi Chen, Varshini Subhash, Marton Havasi, Weiwei Pan, and F. Doshi-Velez. “What Makes a Good Explanation?: A Harmonized View of Properties of Explanations”. In: Nov. 2022. URL: https://www.semanticscholar.org/paper/What-Makes-a-Good-Explanation%3A-A-Harmonized-View-of-Chen-Subhash/1a58f5462211e2e26ab65ee1f8629821608f744e (visited on 11/21/2023).

[19] Michael Chromik and Andreas Butz. “Human-XAI Interaction: A Review and Design Principles for Explanation User Interfaces”. en. In: *Human-Computer Interaction – INTERACT 2021*. Ed. by Carmelo Ardito et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021, pp. 619–640. ISBN: 978-3-030-85616-8. DOI: 10.1007/978-3-030-85616-8-36.

[20] Hyung Won Chung et al. “Scaling instruction-finetuned language models”. en. In: *arXiv.org* abs/2210.11416 (Oct. 2022). arXiv:2210.11416 [cs]. DOI: 10.48550/arXiv.2210.11416. URL: https://www.semanticscholar.org/paper/cdbd4f9b6ab2e2fd1ddf5400d5ed2c18960635d1 (visited on 06/05/2024).

[21] Lawrence Chung, Brian A. Nixon, Eric Yu, and John Mylopoulos. *Non-Functional Requirements in Software Engineering*. Boston, MA: Springer US, 2000. ISBN: 978-1-4613-7403-9. DOI: 10.1007/978-1-4615-5269-7. URL: http://link.springer.com/10.1007/978-1-4615-5269-7 (visited on 06/26/2024).

[22] Douglas Cirqueira, Markus Helfert, and Marija Bezbradica. “Towards Design Principles for User-Centric Explainable AI in Fraud Detection”. en. In: *Artificial Intelligence in HCI*. Ed. by Helmut Degen and Stavroula Ntoa. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021, pp. 21–40. ISBN: 978-3-030-77772-2. DOI: 10.1007/978-3-030-77772-2_2.

[23] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. “What Does BERT Look at? An Analysis of BERT’s Attention”. In: *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 276–286. DOI: 10.18653/v1/W19-4828. URL: https://aclanthology.org/W19-4828 (visited on 10/25/2023).

[24] Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. *Towards automated circuit discovery for mechanistic interpretability*. en. arXiv:2304.14997 [cs]. July 2023. URL: http://arxiv.org/abs/2304.14997 (visited on 09/18/2023).

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423 (visited on 02/15/2024).

[26] Finale Doshi-Velez and Been Kim. *Towards a rigorous science of interpretable machine learning*. en. arXiv:1702.08608 [cs, stat]. Mar. 2017. DOI: 10.48550/arXiv.1702.08608. URL: http://arxiv.org/abs/1702.08608 (visited on 09/28/2023).

[27] Mengnan Du, Ninghao Liu, and Xia Hu. “Techniques for interpretable machine learning”. In: *Communications of the ACM* 63.1 (Dec. 2019), pp. 68–77. ISSN: 0001-0782. DOI: 10.1145/3359786. URL: https://dl.acm.org/doi/10.1145/3359786 (visited on 10/04/2023).

[28] Mengnan Du, Ninghao Liu, Fan Yang, Shuiwang Ji, and Xia Hu. “On attribution of recurrent neural network predictions via additive decomposition”. en. In: *The World Wide Web Conference*. WWW ’19. New York, NY, USA: Association for Computing Machinery, May 2019, pp. 383–393. ISBN: 978-1-4503-6674-8. DOI: 10.1145/3308558.3313545. URL: https://doi.org/10.1145/3308558.3313545 (visited on 06/26/2024).

[29] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. “HotFlip: White-Box Adversarial Examples for Text Classification”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 31–36. DOI: 10.18653/v1/P18-2006. URL: https://aclanthology.org/P18-2006 (visited on 10/20/2023).

[30] Upol Ehsan and Mark O. Riedl. “Human-Centered Explainable AI: Towards a Reflective Sociotechnical Approach”. en. In: *HCI International 2020 - Late Breaking Papers: Multimodality and Intelligence*. Ed. by Constantine Stephanidis, Masaaki Kurosu, Helmut Degen, and Lauren Reinerman-Jones. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020, pp. 449–466. ISBN: 978-3-030-60117-1. DOI: 10.1007/978-3-030-60117-1_33.

[31] Upol Ehsan, Koustuv Saha, Munmun De Choudhury, and Mark O. Riedl. “Charting the Sociotechnical Gap in Explainable AI: A Framework to Address the Gap in XAI”. In: *Proceedings of the ACM on Human-Computer Interaction* 7.CSCW1 (Apr. 2023), 34:1–34:32. DOI: 10.1145/3579467. URL: https://dl.acm.org/doi/10.1145/3579467 (visited on 01/17/2024).

[32] Nils Feldhus et al. “InterroLang: exploring NLP models and datasets through dialogue-based explanations”. en. In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 5399–5421. DOI: 10.18653/v1/2023.findings-emnlp.359. URL: https://aclanthology.org/2023.findings-emnlp.359 (visited on 06/26/2024).

[33] Shi Feng et al. “Pathologies of Neural Models Make Interpretations Difficult”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 3719–3728. DOI: 10.18653/v1/D18-1407. URL: https://aclanthology.org/D18-1407 (visited on 10/10/2023).

[34] Isabel O. Gallegos et al. “Bias and fairness in large language models: a survey”. en. In: *Computational Linguistics* abs/2309.770 (Sept. 2023). DOI: 10.48550/arXiv.2309.00770. URL: https://www.semanticscholar.org/paper/bcfa73aedf1b2d1ee4f168e21298a37ac55a37f7 (visited on 04/15/2024).

[35] Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. “Transformer Feed-Forward Layers Build Predictions by Promoting Concepts in the Vocabulary Space”. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 30–45. DOI: 10.18653/v1/2022.emnlp-main.3. URL: https://aclanthology.org/2022.emnlp-main.3 (visited on 02/28/2024).

[36] Mor Geva et al. “LM-Debugger: An Interactive Tool for Inspection and Intervention in Transformer-Based Language Models”. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Ed. by Wanxiang Che and Ekaterina Shutova. Abu Dhabi, UAE: Association for Computational Linguistics, Dec. 2022, pp. 12–21. DOI: 10.18653/v1/2022.emnlp-demos.2. URL: https://aclanthology.org/2022.emnlp-demos.2 (visited on 11/02/2023).

[37] Bryce Goodman and Seth Flaxman. “European Union Regulations on Algorithmic Decision-Making and a “Right to Explanation””. en. In: *AI Magazine* 38.3 (Oct. 2017). Number: 3, pp. 50–57. ISSN: 2371-9621. DOI: 10.1609/aimag.v38i3.2741. URL: https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/2741 (visited on 09/28/2023).

[38] Mara Graziani et al. “A global taxonomy of interpretable AI: unifying the terminology for the technical and social sciences”. en. In: *Artificial Intelligence Review* 56.4 (Apr. 2023), pp. 3473–3504. ISSN: 1573-7462. DOI: 10.1007/s10462-022-10256-8. URL: https://doi.org/10.1007/s10462-022-10256-8 (visited on 11/18/2023).

[39] Hongyan Gu et al. “Augmenting Pathologists with NaviPath: Design and Evaluation of a Human-AI Collaborative Navigation System”. In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. CHI ’23. New York, NY, USA: Association for Computing Machinery, Apr. 2023, pp. 1–19. ISBN: 978-1-4503-9421-5. DOI: 10.1145/3544548.3580694. URL: https://dl.acm.org/doi/10.1145/3544548.3580694 (visited on 02/29/2024).

[40] Sai Gurrapu, Ajay Kulkarni, Lifu Huang, Ismini Lourentzou, and Feras A. Batarseh. “Rationalization for explainable NLP: a survey”. In: *Frontiers in Artificial Intelligence* 6 (Sept. 2023), p. 1225093. ISSN: 2624-8212. DOI: 10.3389/frai.2023.1225093. URL: https://www.frontiersin.org/articles/10.3389/frai.2023.1225093/full (visited on 11/08/2023).

[41] Yaru Hao, Li Dong, Furu Wei, and Ke Xu. “Self-Attention Attribution: Interpreting Information Interactions Inside Transformer”. en. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.14 (May 2021). Number: 14, pp. 12963–12971. ISSN: 2374-3468. DOI: 10.1609/aaai.v35i14.17533. URL: https://ojs.aaai.org/index.php/AAAI/article/view/17533 (visited on 10/20/2023).

[42] Andreas Hinderks, Anna-Lena Meiners, Francisco José Domínguez Mayo, and Jörg Thomaschewski. *Interpreting the Results from the User Experience Questionnaire (UEQ) Using the Importance-Performance Analysis (IPA)*. Sept. 2019. DOI: 10.5220/0008366503880395.

[43] Robert R. Hoffman, Shane T. Mueller, Gary Klein, and Jordan Litman. “Measures for explainable AI: Explanation goodness, user satisfaction, mental models, curiosity, trust, and human-AI performance”. In: *Frontiers in Computer Science* 5 (2023). ISSN: 2624-9898. URL: https://www.frontiersin.org/articles/10.3389/fcomp.2023.1096257 (visited on 11/21/2023).

[44] Robert R. Hoffman, Shane T. Mueller, Gary Klein, and Jordan Litman. “Metrics for explainable AI: challenges and prospects”. en. In: *arXiv.org* abs/1812.4608 (Dec. 2018). DOI: 10.48550/arXiv.1812.04608. URL: https://www.semanticscholar.org/paper/be711f681580d3a02c8bc4c4dab0c7a043f4e1d2 (visited on 01/30/2024).

[45] Fred Hohman, Andrew Head, Rich Caruana, Robert DeLine, and Steven M. Drucker. “Gamut: A Design Probe to Understand How Data Scientists Understand Machine Learning Models”. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI ’19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 1–13. ISBN: 978-1-4503-5970-2. DOI: 10.1145/3290605.3300809. URL: https://dl.acm.org/doi/10.1145/3290605.3300809 (visited on 11/24/2023).

[46] Andreas Holzinger, André Carrington, and Heimo Müller. “Measuring the Quality of Explanations: The System Causability Scale (SCS)”. en. In: *KI - Künstliche Intelligenz* 34.2 (June 2020), pp. 193–198. ISSN: 1610-1987. DOI: 10.1007/s13218-020-00636-z. URL: https://doi.org/10.1007/s13218-020-00636-z (visited on 01/12/2024).

[47] Sungsoo Ray Hong, Jessica Hullman, and Enrico Bertini. “Human Factors in Model Interpretability: Industry Practices, Challenges, and Needs”. In: *Proceedings of the ACM on Human-Computer Interaction* 4.CSCW1 (May 2020), 68:1–68:26. DOI: 10.1145/3392878. URL: https://dl.acm.org/doi/10.1145/3392878 (visited on 10/04/2023).

[48] Benjamin Hoover, Hendrik Strobelt, and Sebastian Gehrmann. “exBERT: A Visual Analysis Tool to Explore Learned Representations in Transformer Models”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Online: Association for Computational Linguistics, July 2020, pp. 187–196. DOI: 10.18653/v1/2020.acl-demos.22. URL: https://aclanthology.org/2020.acl-demos.22 (visited on 09/19/2023).

[49] Alon Jacovi and Yoav Goldberg. “Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness?” In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 4198–4205. DOI: 10.18653/v1/2020.acl-main.386. URL: https://aclanthology.org/2020.acl-main.386 (visited on 10/04/2023).

[50] Sarthak Jain and Byron C. Wallace. “Attention is not Explanation”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 3543–3556. DOI: 10.18653/v1/N19-1357. URL: https://aclanthology.org/N19-1357 (visited on 09/19/2023).

[51] Albert Q. Jiang et al. *Mistral 7B*. en. arXiv:2310.06825 [cs]. Oct. 2023. DOI: 10.48550/arXiv.2310.06825. URL: http://arxiv.org/abs/2310.06825 (visited on 05/14/2024).

[52] Jinglu Jiang, Surinder Kahai, and Ming Yang. “Who needs explanation and when? Juggling explainable AI and user epistemic uncertainty”. In: *International Journal of Human-Computer Studies* 165 (Sept. 2022), p. 102839. ISSN: 1071-5819. DOI: 10.1016/j.ijhcs.2022.102839. URL: https://www.sciencedirect.com/science/article/pii/S1071581922000660 (visited on 11/23/2023).

[53] Jean Kaddour et al. *Challenges and applications of large language models*. en. arXiv:2307.10169 [cs]. July 2023. URL: http://arxiv.org/abs/2307.10169 (visited on 09/18/2023).

[54] Harmanpreet Kaur et al. “Interpreting Interpretability: Understanding Data Scientists’ Use of Interpretability Tools for Machine Learning”. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI ’20. New York, NY, USA: Association for Computing Machinery, Apr. 2020, pp. 1–14. ISBN: 978-1-4503-6708-0. DOI: 10.1145/3313831.3376219. URL: https://dl.acm.org/doi/10.1145/3313831.3376219 (visited on 02/22/2024).

[55] Minjung Kim, Saebyeol Kim, Jinwoo Kim, Tae-Jin Song, and Yuyoung Kim. “Do stakeholder needs differ? - Designing stakeholder-tailored Explainable Artificial Intelligence (XAI) interfaces”. In: *International Journal of Human-Computer Studies* 181 (Jan. 2024), p. 103160. ISSN: 1071-5819. DOI: 10.1016/j.ijhcs.2023.103160. URL: https://www.sciencedirect.com/science/article/pii/S1071581923001696 (visited on 11/21/2023).

[56] Miryung Kim, Thomas Zimmermann, Robert DeLine, and Andrew Begel. “Data Scientists in Software Teams: State of the Art and Challenges”. en. In: *IEEE Transactions on Software Engineering* 44.11 (Nov. 2018), pp. 1024–1038. ISSN: 0098-5589, 1939-3520, 2326-3881. DOI: 10.1109/TSE.2017.2754374. URL: https://ieeexplore.ieee.org/document/8046093/ (visited on 04/04/2024).

[57] Maximilian A. Köhl et al. “Explainability as a Non-Functional Requirement”. In: *2019 IEEE 27th International Requirements Engineering Conference (RE)*. ISSN: 2332-6441. Sept. 2019, pp. 363–368. DOI: 10.1109/RE.2019.00046. URL: https://ieeexplore.ieee.org/document/8920711 (visited on 11/20/2023).

[58] Enja Kokalj, Blaž Škrlj, Nada Lavrač, Senja Pollak, and Marko Robnik-Šikonja. “BERT meets Shapley: Extending SHAP Explanations to Transformer-based Classifiers”. In: *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*. Online: Association for Computational Linguistics, Apr. 2021, pp. 16–21. URL: https://aclanthology.org/2021.hackashop-1.3 (visited on 10/10/2023).

[59] Narine Kokhlikyan et al. “Captum: A unified and generic model interpretability library for PyTorch”. In: *ArXiv* (Sept. 2020). URL: https://www.semanticscholar.org/paper/Captum%3A-A-unified-and-generic-model-library-for-Kokhlikyan-Miglani/8c3babcb113081d0c4cfdfbd6fb3518a595892c9 (visited on 11/22/2023).

[60] Bonan Kou, Shengmai Chen, Zhijie Wang, Lei Ma, and Tianyi Zhang. *Is model attention aligned with human attention? An empirical study on large language models for code generation*. en. arXiv:2306.01220 [cs]. June 2023. DOI: 10.1145/3660807. URL: http://arxiv.org/abs/2306.01220 (visited on 10/23/2023).

[61] Josua Krause, Adam Perer, and Kenney Ng. “Interacting with Predictions: Visual Inspection of Black-box Machine Learning Models”. en. In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. San Jose California USA: ACM, May 2016, pp. 5686–5697. ISBN: 978-1-4503-3362-7. DOI: 10.1145/2858036.2858529. URL: https://dl.acm.org/doi/10.1145/2858036.2858529 (visited on 04/26/2024).

[62] Rodrigo Schames Kreitchmann, Francisco J. Abad, Vicente Ponsoda, Maria Dolores Nieto, and Daniel Morillo. “Controlling for Response Biases in Self-Report Scales: Forced-Choice vs. Psychometric Modeling of Likert Items”. English. In: *Frontiers in Psychology* 10 (Oct. 2019). Publisher: Frontiers. ISSN: 1664-1078. DOI: 10.3389/fpsyg.2019.02309. URL: https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2019.02309/full (visited on 05/20/2024).

[63] Markus Langer et al. “What do we want from Explainable Artificial Intelligence (XAI)? – A stakeholder perspective on XAI and a conceptual model guiding interdisciplinary XAI research”. In: *Artificial Intelligence* 296 (July 2021), p. 103473. ISSN: 0004-3702. DOI: 10.1016/j.artint.2021.103473. URL: https://www.sciencedirect.com/science/article/pii/S0004370221000242 (visited on 11/20/2023).

[64] Bettina Laugwitz, Theo Held, and Martin Schrepp. “Construction and evaluation of a user experience questionnaire”. In: *HCI and usability for education and work: 4th symposium of the workgroup human-computer interaction and usability engineering of the austrian computer society, USAB 2008, graz, austria, november 20-21, 2008. Proceedings 4*. Springer, 2008, pp. 63–76.

[65] Dong-Ho Lee et al. “XMD: an end-to-end framework for interactive explanation-based debugging of NLP models”. en. In: *Annual Meeting of the Association for Computational Linguistics*. Vol. abs/2210.16978. arXiv:2210.16978 [cs]. arXiv, Oct. 2022. DOI: 10.48550/arXiv.2210.16978. URL: https://www.semanticscholar.org/paper/d10f3857f69edf74565ff786afd5d8632849666a (visited on 05/10/2024).

[66] Jiwei Li, Will Monroe, and Dan Jurafsky. “Understanding Neural Networks through Representation Erasure”. In: (2016). Publisher: arXiv Version Number: 3. DOI: 10.48550/ARXIV.1612.08220. URL: https://arxiv.org/abs/1612.08220 (visited on 11/09/2023).

[67] Yingji Li, Mengnan Du, Rui Song, Xin Wang, and Ying Wang. *A survey on fairness in large language models*. en. arXiv:2308.10149 [cs]. Aug. 2023. DOI: 10.48550/arXiv.2308.10149. URL: http://arxiv.org/abs/2308.10149 (visited on 12/11/2023).

[68] Yingji Li, Mengnan Du, Xin Wang, and Ying Wang. “Prompt Tuning Pushes Farther, Contrastive Learning Pulls Closer: A Two-Stage Approach to Mitigate Social Biases”. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 14254–14267. DOI: 10.18653/v1/2023.acl-long.797. URL: https://aclanthology.org/2023.acl-long.797 (visited on 04/15/2024).

[69] Q. Vera Liao, Daniel Gruen, and Sarah Miller. “Questioning the AI: Informing Design Practices for Explainable AI User Experiences”. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. arXiv:2001.02478 [cs]. Apr. 2020, pp. 1–15. DOI: 10.1145/3313831.3376590. URL: http://arxiv.org/abs/2001.02478 (visited on 10/04/2023).

[70] Q. Vera Liao and Jennifer Wortman Vaughan. *AI transparency in the age of LLMs: a human-centered research roadmap*. en. arXiv:2306.01941 [cs]. Aug. 2023. URL: http://arxiv.org/abs/2306.01941 (visited on 10/25/2023).

[71] Q. Vera Liao and Ziang Xiao. *Rethinking model evaluation as narrowing the socio-technical gap*. en. arXiv:2306.03100 [cs]. June 2023. DOI: 10.48550/arXiv.2306.03100. URL: http://arxiv.org/abs/2306.03100 (visited on 12/08/2023).

[72] Q. Vera Liao, Yunfeng Zhang, Ronny Luss, Finale Doshi-Velez, and Amit Dhurandhar. “Connecting Algorithmic Research and Usage Contexts: A Perspective of Contextualized Evaluation for Explainable AI”. en. In: *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing* 10 (Oct. 2022), pp. 147–159. ISSN: 2769-1349. DOI: 10.1609/hcomp.v10i1.21995. URL: https://ojs.aaai.org/index.php/HCOMP/article/view/21995 (visited on 11/20/2023).

[73] Zachary C. Lipton. “The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery.” en. In: *Queue* 16.3 (June 2018), pp. 31–57. ISSN: 1542-7730, 1542-7749. DOI: 10.1145/3236386.3241340. URL: https://dl.acm.org/doi/10.1145/3236386.3241340 (visited on 09/28/2023).

[74] Yiheng Liu et al. “Understanding LLMs: a comprehensive overview from training to inference”. en. In: *arXiv.org* abs/2401.2038 (Jan. 2024). arXiv:2401.02038 [cs]. DOI: 10.48550/arXiv.2401.02038. URL: https://www.semanticscholar.org/paper/efc5e94635a850ede9c1f8dbce65d5dc536f3bfb (visited on 01/08/2024).

[75] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. “Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 8086–8098. DOI: 10.18653/v1/2022.acl-long.556. URL: https://aclanthology.org/2022.acl-long.556 (visited on 06/04/2024).

[76] Scott Lundberg and Su-In Lee. “A Unified Approach to Interpreting Model Predictions”. In: (2017). Publisher: arXiv Version Number: 2. DOI: 10.48550/ARXIV.1705.07874. URL: https://arxiv.org/abs/1705.07874 (visited on 11/13/2023).

[77] Haoyan Luo and Lucia Specia. *From understanding to utilization: a survey on explainability for large language models*. en. arXiv:2401.12874 [cs]. Feb. 2024. DOI: 10.48550/arXiv.2401.12874. URL: http://arxiv.org/abs/2401.12874 (visited on 06/26/2024).

[78] Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. “Towards Faithful Model Explanation in NLP: A Survey”. In: *Computational Linguistics* (Jan. 2024), pp. 1–70. ISSN: 0891-2017. DOI: 10.1162/coli_a_00511. URL: https://doi.org/10.1162/coli_a_00511 (visited on 04/04/2024).

[79] Rajib Mall. *Fundamentals of software engineering*. en. Fifth edition. Eastern economy edition. Delhi: PHI Learning Private Limited, 2019. ISBN: 978-93-88028-03-5.

[80] Iñigo Martinez, Elisabeth Viles, and Igor G. Olaizola. “Data Science Methodologies: Current Challenges and Future Approaches”. In: *Big Data Research* 24 (May 2021), p. 100183. ISSN: 2214-5796. DOI: 10.1016/j.bdr.2020.100183. URL: https://www.sciencedirect.com/science/article/pii/S2214579620300514 (visited on 05/14/2024).

[81] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. “Locating and Editing Factual Associations in GPT”. en. In: *Advances in Neural Information Processing Systems* 35 (Dec. 2022), pp. 17359–17372. URL: https://proceedings.neurips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html (visited on 10/04/2023).

[82] Vivek Miglani, Aobo Yang, Aram Markosyan, Diego Garcia-Olano, and Narine Kokhlikyan. “Using captum to explain generative language models”. In: *Proceedings of the 3rd workshop for natural language processing open source software (NLP-OSS 2023)*. Ed. by Liling Tan, Dmitrijs Milajevs, Geeticka Chauhan, Jeremy Gwinnup, and Elijah Rippeth. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 165–173. DOI: 10.18653/v1/2023.nlposs-1.19. URL: https://aclanthology.org/2023.nlposs-1.19.

[83] Tim Miller. “Explanation in artificial intelligence: Insights from the social sciences”. In: *Artificial Intelligence* 267 (Feb. 2019), pp. 1–38. ISSN: 0004-3702. DOI: 10.1016/j.artint.2018.07.007. URL: https://www.sciencedirect.com/science/article/pii/S0004370218305988 (visited on 10/04/2023).

[84] Akash Kumar Mohankumar et al. “Towards Transparent and Explainable Attention Models”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 4206–4216. DOI: 10.18653/v1/2020.acl-main.387. URL: https://aclanthology.org/2020.acl-main.387 (visited on 09/19/2023).

[85] Sina Mohseni, Niloofar Zarei, and Eric D. Ragan. “A Multidisciplinary Survey and Framework for Design and Evaluation of Explainable AI Systems”. In: *ACM Transactions on Interactive Intelligent Systems* 11.3-4 (Sept. 2021), 24:1–24:45. ISSN: 2160-6455. DOI: 10.1145/3387166. URL: https://dl.acm.org/doi/10.1145/3387166 (visited on 11/21/2023).

[86] Christoph Molnar. *Interpretable Machine Learning*. en. Lulu.com, 2020. ISBN: 978-0-244-76852-2.

[87] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. “Layer-Wise Relevance Propagation: An Overview”. In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Berlin, Heidelberg: Springer-Verlag, Dec. 2022, pp. 193–209. ISBN: 978-3-030-28953-9. URL: https://doi.org/10.1007/978-3-030-28954-6_10 (visited on 10/11/2023).

[88] Edoardo Mosca et al. “IFAN: an explainability-focused interaction framework for humans and NLP models”. en. In: *International Joint Conference on Natural Language Processing*. Vol. abs/2303.3124. arXiv:2303.03124 [cs]. arXiv, Mar. 2023. DOI: 10.48550/arXiv.2303.03124. URL: https://www.semanticscholar.org/paper/693f97562b6e57cecb07da20c4b96a96c238b6df (visited on 05/10/2024).

[89] Shane T. Mueller et al. “Principles of explanation in human-AI systems”. en. In: *arXiv.org* abs/2102.4972 (Feb. 2021). arXiv:2102.04972 [cs]. DOI: 10.48550/arXiv.2102.04972. URL: https://www.semanticscholar.org/paper/d591e9a55f7bbf123ce2262500b2d8794052dc0f (visited on 06/03/2024).

[90] Muhammad Naeem, Wilson Ozuem, Kerry Howell, and Silvia Ranfagni. “A Step-by-Step Process of Thematic Analysis to Develop a Conceptual Model in Qualitative Research”. en. In: *International Journal of Qualitative Methods* 22 (Mar. 2023). Publisher: SAGE Publications Inc, p. 16094069231205789. ISSN: 1609-4069. DOI: 10.1177/16094069231205789. URL: https://doi.org/10.1177/16094069231205789 (visited on 03/19/2024).

[91] Neel Nanda and Joseph Bloom. *TransformerLens*. 2022. URL: https://github.com/neelnanda-io/TransformerLens.

[92] Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. *Progress measures for grokking via mechanistic interpretability*. en. arXiv:2301.05217 [cs]. Oct. 2023. DOI: 10.48550/arXiv.2301.05217. URL: http://arxiv.org/abs/2301.05217 (visited on 11/07/2023).

[93] Humza Naveed et al. “A Comprehensive Overview of Large Language Models”. In: (2023). Publisher: arXiv Version Number: 5. DOI: 10.48550/ARXIV.2307.06435. URL: https://arxiv.org/abs/2307.06435 (visited on 11/09/2023).

[94] Ingrid Nunes and Dietmar Jannach. “A systematic review and taxonomy of explanations in decision support and recommender systems”. en. In: *User Modeling and User-Adapted Interaction* 27.3 (Dec. 2017), pp. 393–444. ISSN: 1573-1391. DOI: 10.1007/s11257-017-9195-0. URL: https://doi.org/10.1007/s11257-017-9195-0 (visited on 10/06/2023).

[95] OpenAI. *ChatGPT: a large-scale generative model for open-domain chat*. 2021. URL: https://github.com/openai/gpt-3.

[96] OpenAI et al. *GPT-4 technical report*. en. arXiv:2303.08774 [cs]. Mar. 2024. DOI: 10.48550/arXiv.2303.08774. URL: http://arxiv.org/abs/2303.08774 (visited on 06/05/2024).

[97] Cecilia Panigutti et al. "The role of explainable AI in the context of the AI act". en. In: *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '23. New York, NY, USA: Association for Computing Machinery, June 2023, pp. 1139–1150. ISBN: 9798400701924. DOI: 10.1145/3593013.3594069. URL: https://dl.acm.org/doi/10.1145/3593013.3594069 (visited on 01/09/2024).

[98] Luis Perez, Lizi Ottens, and Sudharshan Viswanathan. *Automatic code generation using pre-trained language models*. en. arXiv:2102.10535 [cs]. Feb. 2021. DOI: 10.48550/arXiv.2102.10535. URL: http://arxiv.org/abs/2102.10535 (visited on 06/25/2024).

[99] Charles Pierse. *Transformers interpret*. Feb. 2021. URL: https://github.com/cdpierse/transformers-interpret.

[100] Libo Qin et al. "Large language models meet NLP: a survey". en. In: *arXiv.org* abs/2405.12819 (May 2024). arXiv:2405.12819 [cs]. DOI: 10.48550/arXiv.2405.12819. URL: https://www.semanticscholar.org/paper/6180615110d86033887d0dc4feebe8e8a162346f (visited on 06/04/2024).

[101] Oliver Radley-Gardner, Hugh Beale, and Reinhard Zimmermann, eds. *Fundamental Texts On European Private Law*. en. Hart Publishing, 2016. ISBN: 978-1-78225-864-3. DOI: 10.5040/9781782258674. URL: http://www.bloomsburycollections.com/book/fundamental-texts-on-european-private-law-1 (visited on 02/15/2024).

[102] Gabriëlle Ras, Marcel van Gerven, and Pim Haselager. "Explanation Methods in Deep Learning: Users, Values, Concerns and Challenges". en. In: *Explainable and Interpretable Models in Computer Vision and Machine Learning*. Ed. by Hugo Jair Escalante et al. The Springer Series on Challenges in Machine Learning. Cham: Springer International Publishing, 2018, pp. 19–36. ISBN: 978-3-319-98131-4. URL: https://doi.org/10.1007/978-3-319-98131-42 (visited on 11/30/2023).

[103] Marco Ribeiro, Sameer Singh, and Carlos Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 97–101. DOI: 10.18653/v1/N16-3020. URL: https://aclanthology.org/N16-3020 (visited on 10/11/2023).

[104] Mireia Ribera Turró and Agata Lapedriza. "Can we do better explanations? A proposal of User-Centered Explainable AI". In: Mar. 2019.

[105] Gabriele Sarti, Nils Feldhus, Ludwig Sickert, and Oskar van der Wal. "Inseq: An Interpretability Toolkit for Sequence Generation Models". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*. Ed. by Danushka Bollegala, Ruihong Huang, and Alan Ritter. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 421–435. DOI: 10.18653/v1/2023.acl-demo.40. URL: https://aclanthology.org/2023.acl-demo.40 (visited on 01/02/2024).

[106] Tjeerd A. J. Schoonderwoerd, Wiard Jorritsma, Mark A. Neerincx, and Karel van den Bosch. "Human-centered XAI: Developing design patterns for explanations of clinical decision support systems". In: *International Journal of Human-Computer Studies* 154 (Oct. 2021), p. 102684. ISSN: 1071-5819. DOI: 10.1016/j.ijhcs.2021.102684. URL: https://www.sciencedirect.com/science/article/pii/S1071581921001026 (visited on 11/23/2023).

[107] Martin Schrepp, Andreas Hinderks, and Jörg Thomaschewski. *Applying the User Experience Questionnaire (UEQ) in Different Evaluation Scenarios*. Pages: 392. June 2014. ISBN: 978-3-319-07667-6. DOI: 10.1007/978-3-319-07668-3_37.

[108] Sofia Serrano and Noah A. Smith. "Is Attention Interpretable?" In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 2931–2951. DOI: 10.18653/v1/P19-1282. URL: https://aclanthology.org/P19-1282 (visited on 09/19/2023).

[109] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. “Learning important features through propagating activation differences”. en. In: *International conference on machine learning*. PMLR, 2017, pp. 3145–3153.

[110] Sandipan Sikdar, Parantapa Bhattacharya, and Kieran Heese. “Integrated Directional Gradients: Feature Interaction Attribution for Neural NLP Models”. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli. Online: Association for Computational Linguistics, Aug. 2021, pp. 865–878. DOI: 10.18653/v1/2021.acl-long.71. URL: https://aclanthology.org/2021.acl-long.71 (visited on 11/10/2023).

[111] Chandan Singh, Jeevana Priya Inala, Michel Galley, Rich Caruana, and Jianfeng Gao. “Rethinking interpretability in the era of large language models”. en. In: *arXiv.org* abs/2402.1761 (Jan. 2024). arXiv:2402.01761 [cs]. DOI: 10.48550/arXiv.2402.01761. URL: https://www.semanticscholar.org/paper/d9bf49d90e1c646ade1c535f8e93d2c7413da14b (visited on 04/08/2024).

[112] Kacper Sokol and Peter Flach. “Explainability fact sheets: a framework for systematic assessment of explainable approaches”. In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. FAT* ’20. New York, NY, USA: Association for Computing Machinery, Jan. 2020, pp. 56–67. ISBN: 978-1-4503-6936-7. DOI: 10.1145/3351095.3372870. URL: https://dl.acm.org/doi/10.1145/3351095.3372870 (visited on 06/03/2024).

[113] Timo Speith and Markus Langer. “A new perspective on evaluation methods for explainable artificial intelligence (XAI)”. en. In: *2023 IEEE 31st International Requirements Engineering Conference Workshops (REW)*. Hannover, Germany: IEEE, Sept. 2023, pp. 325–331. ISBN: 9798350326918. DOI: 10.1109/REW57809.2023.00061. URL: https://ieeexplore.ieee.org/document/10260827/ (visited on 05/16/2024).

[114] Jiayuan Su, Jing Luo, Hongwei Wang, and Lu Cheng. *API is enough: conformal prediction for large language models without logit-access*. en. arXiv:2403.01216 [cs]. Apr. 2024. DOI: 10.48550/arXiv.2403.01216. URL: http://arxiv.org/abs/2403.01216 (visited on 04/15/2024).

[115] Jiao Sun et al. “Investigating Explainability of Generative AI for Code through Scenario-based Design”. In: *27th International Conference on Intelligent User Interfaces*. IUI ’22. New York, NY, USA: Association for Computing Machinery, Mar. 2022, pp. 212–228. ISBN: 978-1-4503-9144-3. DOI: 10.1145/3490099.3511119. URL: https://dl.acm.org/doi/10.1145/3490099.3511119 (visited on 09/18/2023).

[116] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. “Axiomatic attribution for deep networks”. en. In: *Proceedings of the 34th international conference on machine learning - volume 70*. ICML’17. Place: Sydney, NSW, Australia Number of pages: 10. JMLR.org, 2017, pp. 3319–3328.

[117] Harini Suresh, Steven R. Gomez, Kevin K. Nam, and Arvind Satyanarayan. “Beyond Expertise and Roles: A Framework to Characterize the Stakeholders of Interpretable Machine Learning and their Needs”. In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. CHI ’21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 1–16. ISBN: 978-1-4503-8096-6. DOI: 10.1145/3411764.3445088. URL: https://dl.acm.org/doi/10.1145/3411764.3445088 (visited on 11/20/2023).

[118] Harini Suresh, Natalie Lao, and Ilaria Liccardi. “Misplaced Trust: Measuring the Interference of Machine Learning in Human Decision-Making”. In: *Proceedings of the 12th ACM Conference on Web Science*. WebSci ’20. New York, NY, USA: Association for Computing Machinery, July 2020, pp. 315–324. ISBN: 978-1-4503-7989-2. DOI: 10.1145/3394231.3397922. URL: https://dl.acm.org/doi/10.1145/3394231.3397922 (visited on 11/30/2023).

[119] Adly Templeton et al. “Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet”. In: *Transformer Circuits Thread* (2024). URL: https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

[120] Ian Tenney et al. “The Language Interpretability Tool: Extensible, Interactive Visualizations and Analysis for NLP Models”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Ed. by Qun Liu and David Schlangen. Online: Association for Computational Linguistics, Oct. 2020, pp. 107–118. DOI: 10.18653/v1/2020.emnlp-demos.15. URL: https://aclanthology.org/2020.emnlp-demos.15 (visited on 11/08/2023).

[121] Alexey Tikhonov and Ivan Yamshchikov. “Post turing: mapping the landscape of LLM evaluation”. en. In: *Proceedings of the third workshop on natural language generation, evaluation, and metrics (GEM)*. Ed. by Sebastian Gehrmann et al. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 398–412. URL: https://aclanthology.org/2023.gem-1.31.

[122] Richard Tomsett, Dave Braines, Dan Harborne, Alun Preece, and Supriyo Chakraborty. *Interpretable to whom? A role-based model for analyzing interpretable machine learning systems*. en. arXiv:1806.07552 [cs]. June 2018. DOI: 10.48550/arXiv.1806.07552. URL: http://arxiv.org/abs/1806.07552 (visited on 11/30/2023).

[123] Ashish Vaswani et al. “Attention is all you need”. In: *Proceedings of the 31st international conference on neural information processing systems*. NIPS’17. Number of pages: 11 Place: Long Beach, California, USA. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 6000–6010. ISBN: 978-1-5108-6096-4.

[124] Giulia Vilone and Luca Longo. “Notions of explainability and evaluation approaches for explainable artificial intelligence”. In: *Information Fusion* 76 (Dec. 2021), pp. 89–106. ISSN: 1566-2535. DOI: 10.1016/j.inffus.2021.05.009. URL: https://www.sciencedirect.com/science/article/pii/S1566253521001093 (visited on 11/21/2023).

[125] Eric Wallace et al. “AllenNLP Interpret: A Framework for Explaining Predictions of NLP Models”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 7–12. DOI: 10.18653/v1/D19-3002. URL: https://aclanthology.org/D19-3002 (visited on 10/10/2023).

[126] Boxin Wang et al. “DecodingTrust: a comprehensive assessment of trustworthiness in GPT models”. en. In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh et al. Vol. 36. Curran Associates, Inc., 2023, pp. 31232–31339. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/63cb9921eecf51bfad27a99b2c53dd6d-Paper-Datasets_and_Benchmarks.pdf.

[127] Dakuo Wang et al. “Human-AI Collaboration in Data Science: Exploring Data Scientists’ Perceptions of Automated AI”. In: *Proceedings of the ACM on Human-Computer Interaction* 3.CSCW (Nov. 2019), 211:1–211:24. DOI: 10.1145/3359313. URL: https://dl.acm.org/doi/10.1145/3359313 (visited on 05/15/2024).

[128] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. *Interpretability in the wild: a circuit for indirect object identification in GPT-2 small*. en. arXiv:2211.00593 [cs]. Nov. 2022. DOI: 10.48550/arXiv.2211.00593. URL: http://arxiv.org/abs/2211.00593 (visited on 11/07/2023).

[129] Qianli Wang et al. “LLMCheckup: conversational examination of large language models via interpretability tools”. en. In: *arXiv.org* abs/2401.12576 (Jan. 2024). arXiv:2401.12576 [cs]. DOI: 10.48550/arXiv.2401.12576. URL: https://www.semanticscholar.org/paper/f22456037cf3bfcb9e7e6b0706208a5f1403ceaf (visited on 02/21/2024).

[130] Christine T. Wolf. “Explainability scenarios: towards scenario-based XAI design”. en. In: *Proceedings of the 24th International Conference on Intelligent User Interfaces*. Marina del Ray California: ACM, Mar. 2019, pp. 252–257. ISBN: 978-1-4503-6272-6. DOI: 10.1145/3301275.3302317. URL: https://dl.acm.org/doi/10.1145/3301275.3302317 (visited on 03/21/2024).

[131] Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. “Polyjuice: Generating Counterfactuals for Explaining, Evaluating, and Improving Models”. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 6707–6723. DOI: 10.18653/v1/2021.acl-long.523. URL: https://aclanthology.org/2021.acl-long.523 (visited on 10/27/2023).

[132] Xuansheng Wu et al. *Usable XAI: 10 strategies towards exploiting explainability in the LLM era*. en. arXiv:2403.08946 [cs]. Mar. 2024. DOI: 10.48550/arXiv.2403.08946. URL: http://arxiv.org/abs/2403.08946 (visited on 04/11/2024).

[133] Wei Xu, Zaifeng Gao, and Marvin Dainoff. “An HCAI Methodological Framework: Putting It Into Action to Enable Human-Centered AI”. In: (2023). Publisher: arXiv Version Number: 3. DOI: 10.48550/ARXIV.2311.16027. URL: https://arxiv.org/abs/2311.16027 (visited on 12/04/2023).

[134] Kayo Yin and Graham Neubig. “Interpreting Language Models with Contrastive Explanations”. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 184–198. DOI: 10.18653/v1/2022.emnlp-main.14. URL: https://aclanthology.org/2022.emnlp-main.14 (visited on 05/21/2024).

[135] Haiyan Zhao et al. “Explainability for Large Language Models: A Survey”. en. In: *ACM Transactions on Intelligent Systems and Technology* 15.2 (Apr. 2024), pp. 1–38. ISSN: 2157-6904, 2157-6912. DOI: 10.1145/3639372. URL: https://dl.acm.org/doi/10.1145/3639372 (visited on 03/28/2024).

[136] Wayne Xin Zhao et al. “A Survey of Large Language Models”. In: (2023). DOI: 10.48550/ARXIV.2303.18223. URL: https://arxiv.org/abs/2303.18223 (visited on 11/13/2023).

[137] Xi Zhiheng, Zheng Rui, and Gui Tao. “Safety and Ethical Concerns of Large Language Models”. English. In: *Proceedings of the 22nd Chinese National Conference on Computational Linguistics (Volume 4: Tutorial Abstracts)*. Harbin, China: Chinese Information Processing Society of China, Aug. 2023, pp. 9–16. URL: https://aclanthology.org/2023.ccl-4.2 (visited on 10/23/2023).

[138] Jianlong Zhou, Amir H. Gandomi, Fang Chen, and Andreas Holzinger. “Evaluating the Quality of Machine Learning Explanations: A Survey on Methods and Metrics”. en. In: *Electronics* 10.5 (Jan. 2021). Number: 5 Publisher: Multidisciplinary Digital Publishing Institute, p. 593. ISSN: 2079-9292. DOI: 10.3390/electronics10050593. URL: https://www.mdpi.com/2079-9292/10/5/593 (visited on 11/22/2023).

[139] Julia El Zini and Mariette Awad. “On the Explainability of Natural Language Processing Deep Models”. In: *ACM Computing Surveys* 55.5 (May 2023). arXiv:2210.06929 [cs], pp. 1–31. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3529755. URL: http://arxiv.org/abs/2210.06929 (visited on 10/24/2023).

A

Interview Protocol Formative Study

This section presents the questions of the formative study. This was semi-structured, where all questions were answered but not always in this exact order. This was done by the author and transcribed automatically by software.

Total: 45 mins
A. Admin points (5 mins)

- Introduction to the project
- Explanation of this interview session
- Let the participant know they can leave at any time during the interview
- Ask the participant about the informed consent
- Make sure they are aware that they should be honest!
- Ask the participant if there are any questions at this point

B. *Personal* Background Questions (3 mins)

- For how long have you been a Data Scientist?
- How much experience do you have with AI and ML?
- (How would you rate your degree of understanding AI and ML?)
- What is your understanding of NLP techniques and LLMs?
- (How would you rate your degree of understanding LLMs?)
- Do you have an example of a finishing project that you have worked on that included LLMs?
- How much understanding do you have of XAI techniques? (added march 11th)

C. Understanding the data scientist (3 mins)

- How would you describe the role of a Data Scientist at Elsevier?
- What are the responsibilities of a Data Scientist?
- Is this definition changed for you over time?

D. Workflow LLM-development questions (15 mins)

- Based on the example that they just gave, could you show/explain the current workflow? Which stages are there?
- Which decisions are then made at each of those stages?
- How do you measure the success of the project? Or how do you evaluate it?
- Do you use open source models or inference of close source models? and why?
- How did you do this before OpenAI published this private hosting?
- Do you have any preferred resources or tools that you use to work on LLM-based products? (These can be “open tools” like Jupyter notebooks or anything that Elsevier built in-house)

- What would you say are the most challenging problems you face when working with LLMs?
- And what kind of tool, ideally speaking, would help you solve these problems?
- What other requirements would you like to see in this kind of tool?

E. Current usage of explainability techniques (15 mins)

- How would you define XAI?
- Do you have any transparency, trust or explainability concerns when using LLMs? And why Yes/No?
- What would you say is your degree of awareness of explainability tools and techniques?
- Have you ever used XAI in your data science projects? And specifically in LLMs?
- If **yes**;
  - What was the reason for using XAI?
  - Which techniques did you use?
  - Did you have to include them yourself or do you use a tool?
  - Did you face any issues with those?
  - Would you want better support when using XAI (in terms of information, in terms of usability, etc.)?
  - Is there anything completely missing that you would like to have when using XAI?
- If **no**;
  - Why not?
  - Was the reason that XAI was not needed?
  - Would you want to include some XAI components in your workflow? How would you use it? If no, why is that the case for you?
  - Did you ever try any of the tools available?
  - Do you not think that it could have benefits?
- Regarding LLMs, Prompt engineering for more explanations can also be seen as XAI. Did you consider this?
- Imagine having better explanations of your LLM, do you think that it could help achieve your goal faster or better?
- Is there a difference for you between LLMs and other AI techniques?

F. Ending the interview (7 min)

- Is there, at this point, something you would like to add?
- Is there something that I should have asked you but did not?
- Any suggestions of people we should also talk to?
- Are you still okay that this interview will be used as a data source for the research?

B

Codes

Table B.1 presents some examples of codes that were grouped into various sub-themes.

Table B.1: Various sub-themes with several examples of identified codes

Goals
Better Performance
Hypothesis Checking
Building PoC

Technical Challenges
Debugging
Correct output
Prompt limits

Potential benefits XAI
Increased Trust
Understanding Reasoning
Useful Insights

Steps
Data Preparation
Post processing
Start simple

Utilisation
Q&A
RAG
Classification

Tech Stack
Git
Sagemaker
Terminal

C

Implementation Details

C.1. Final folder structure

Below, the final folder structure is presented. This gives an overview of how such a tool could be structured when implemented. More details can be found on GitHub.

```
/
├── docs
├── examples
├── src
│   └── helmet
│       ├── explainers
│       ├── models
│       ├── utils
│       ├── __init__.py
│       └── updater.py
└── webapp
    ├── api
    └── client
pyproject.toml
README.md
```

C.2. Coding Examples

Here, several coding examples are given to give an idea of how to use HELMET.

```
device = "cuda" # Device setting
checkpoint = "microsoft/Phi-3-mini-4k-instruct" # The model from Huggingface

# The embeddings are needed for the XAI part. Please point to the correct path of
    the embeddings
embeddings = "model.embed_tokens"

model_type = "dec" # what kind of model it is (enc, enc-dec or dec)
```

Listing C.1: Settings for the model

```
question = "I'm going to watch Roland Garros. When does that take place?"

generation_arguments = {
    "max_new_tokens": 200,
    "temperature": 1.0,
    "do_sample": True
}

res, res_id = model.predict(question, generation_arguments)
```

Listing C.2: Prompting the model

D

Tool screenshots

Some screenshots of the final implemented tool are presented here. The following screenshots have been included:

- Home page (Figure D.1)
- Detailed Run without XAI (Figure D.2)
- Resources (Figure D.3)

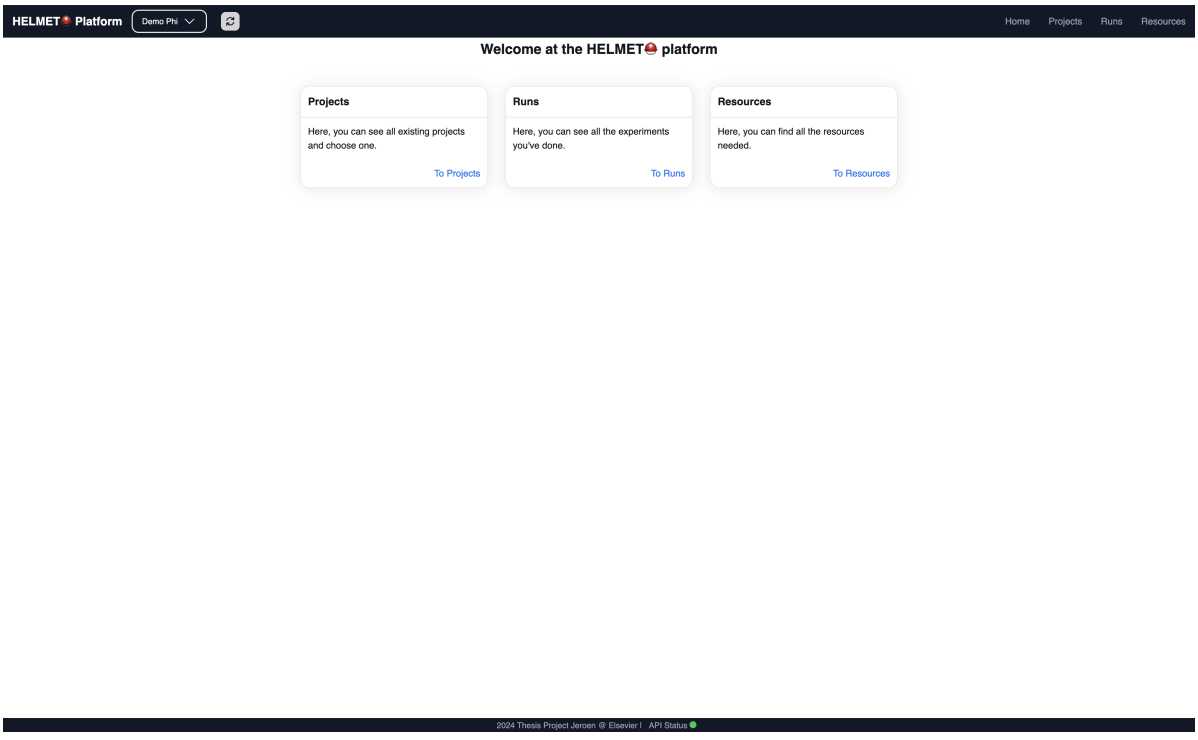


Figure D.1: Home page

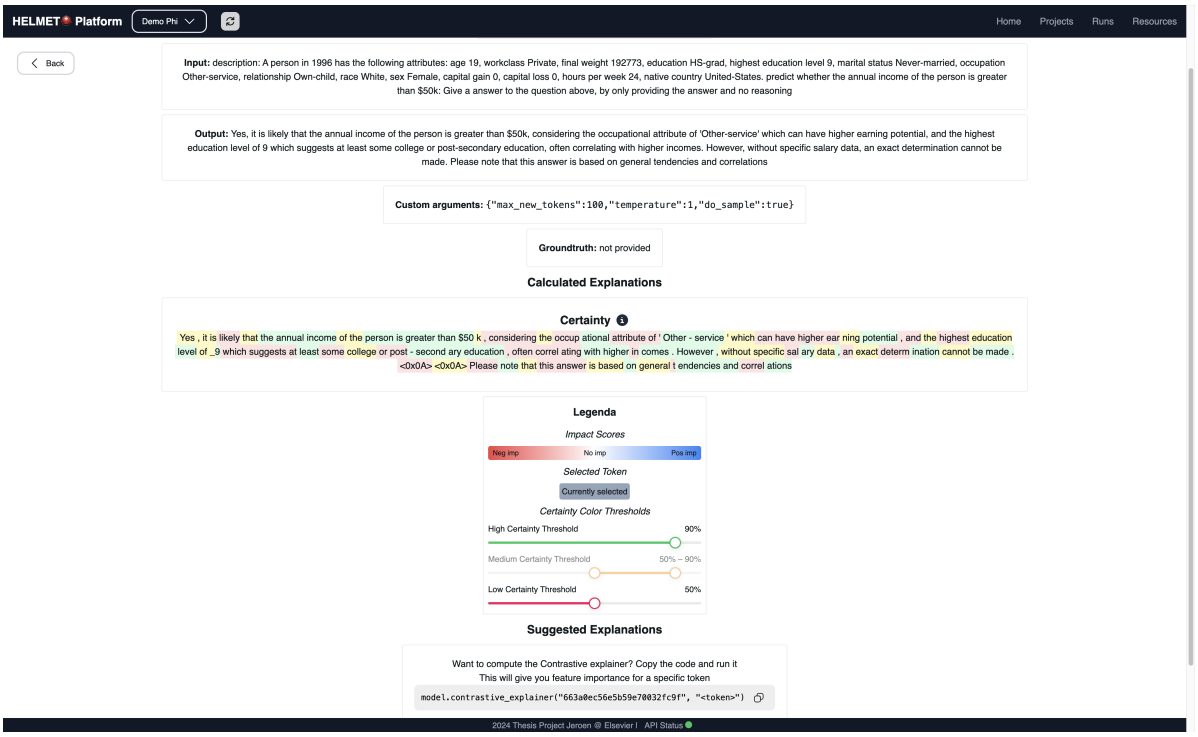


Figure D.2: Detailed page without XAI

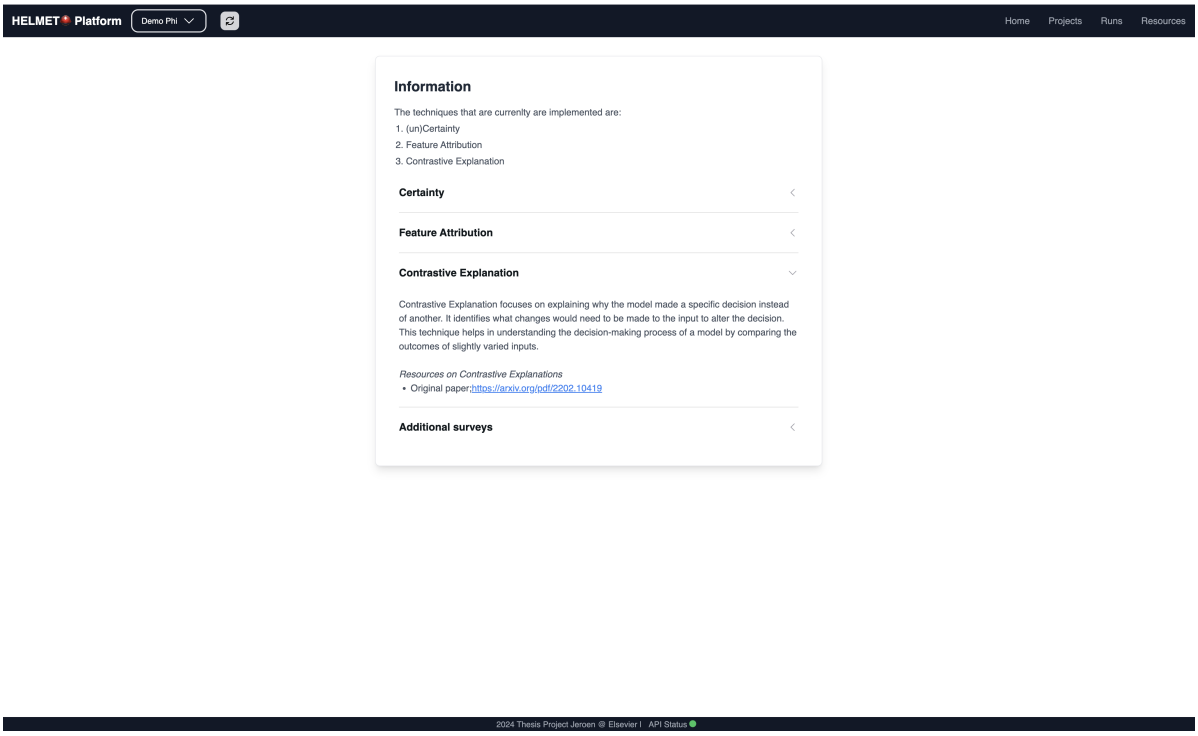


Figure D.3: Resources page

E

Evaluation Questionnaires

This section presents the final questionnaire that was used to evaluate the implementation. The results are presented in chapter 5

E.1. Explanation Satisfaction Scale (ESS)

The first part was the ESS, as was presented first by Hoffman et al. [44]. The questions were presented as follows:

	Strongly Agree	Agree	Neutral	Dis-agree	Strongly dis-agree
From the explanation, **I understand** how the LLM works	◯	◯	◯	◯	◯
This explanation of how the LLM works is **satisfying**	◯	◯	◯	◯	◯
This explanation of how the LLM works has **sufficient detail**.	◯	◯	◯	◯	◯
This explanation of how the LLM works seems **complete**.	◯	◯	◯	◯	◯
This explanation of how the LLM works **tells me how to use** it.	◯	◯	◯	◯	◯
This explanation of how the LLM works is **useful to my goals**.	◯	◯	◯	◯	◯
This explanation of the LLM shows me how **accurate** the LLM is.	◯	◯	◯	◯	◯
This explanation lets me judge when I should **trust and not trust** the LLM	◯	◯	◯	◯	◯

E.2. Fitting the workflow

Please answer the following questions. If it does not apply to you, it is okay to leave the question unanswered.

Question 1: Do you think the setup of having an external platform next to the Jupyter Notebook/Sagemaker is more or less usable?

Question 2: Can this tool be easily integrated into your workflow when creating LLM-based products? Why yes/no?

Question 3: Consider the task of error analysis and improving the prompt. Would this tool be helpful in that phase?

Question 4: Does this tool change your opinion on XAI for LLMs? If yes, in what way?

Question 5: Is there any feature you would like to add to this tool?

E.3. User Experience Evaluation

The last part of the evaluation was regarding the usability of the tool. This was done using the UEQ, a well-established evaluation survey by Laugwitz, Held, and Schrepp [64]. The information and questions are presented now.

For the assessment of the product, please fill out the following questionnaire. The questionnaire consists of pairs of contrasting attributes that may apply to the product. The circles between the attributes represent gradations between the opposites. You can agree with the characteristics by ticking the circle that most closely reflects your impression.

Example

attractive	○	⊗	○	○	○	○	○	unattractive

This response would mean you rate the application as more attractive than unattractive.

Please decide spontaneously. Don't think too long about your decision to make sure that you convey your original impression.
Sometimes you may not be completely sure about your agreement with a particular attribute or find that the attribute does not apply completely to the particular product. Nevertheless, please tick a circle in every line.
It is your personal opinion that counts. Please remember: **there is no wrong or right answer!**

The rest of this page is intentionally left blank. Find the questionnaire on the following page.

Please assess the product now by ticking one circle per line.

	1	2	3	4	5	6	7	
annoying	○	○	○	○	○	○	○	enjoyable
not understandable	○	○	○	○	○	○	○	understandable
creative	○	○	○	○	○	○	○	dull
easy to learn	○	○	○	○	○	○	○	difficult to learn
valuable	○	○	○	○	○	○	○	inferior
boring	○	○	○	○	○	○	○	exciting
not interesting	○	○	○	○	○	○	○	interesting
unpredictable	○	○	○	○	○	○	○	predictable
fast	○	○	○	○	○	○	○	slow
inventive	○	○	○	○	○	○	○	conventional
obstructive	○	○	○	○	○	○	○	supportive
good	○	○	○	○	○	○	○	bad
complicated	○	○	○	○	○	○	○	easy
unlikable	○	○	○	○	○	○	○	pleasing
usual	○	○	○	○	○	○	○	leading edge
unpleasant	○	○	○	○	○	○	○	pleasant
secure	○	○	○	○	○	○	○	not secure
motivating	○	○	○	○	○	○	○	demotivating
meets expectations	○	○	○	○	○	○	○	does not meet expectations
inefficient	○	○	○	○	○	○	○	efficient
clear	○	○	○	○	○	○	○	confusing
impractical	○	○	○	○	○	○	○	practical
organized	○	○	○	○	○	○	○	cluttered
attractive	○	○	○	○	○	○	○	unattractive
friendly	○	○	○	○	○	○	○	unfriendly
conservative	○	○	○	○	○	○	○	innovative