Explainable Prompt Learning for Movie Review Sentiment Analysis

by

Sean Stilwell

Thesis submitted to the University of Ottawa
in partial fulfillment of the requirements for the
Master of Computer Science degree

Examining Committee

The following served on the Examining Committee for this thesis.

Internal Members: Caroline Barrière
Assistant Professor, School of Electrical Engineering & Computer Science
University of Ottawa

Andre Vellino
Associate Professor, School of Information Studies
University of Ottawa

Supervisor: Diana Inkpen
Professor, School of Electrical Engineering & Computer Science
University of Ottawa

Declaration of Authorship

I hereby certify that this thesis is entirely my own original work except where otherwise indicated. I am aware of the University of Ottawa regulations concerning plagiarism, including those regarding consequent disciplinary actions. Any use of the works of any other author, in any form, is properly acknowledged at their point of use.

Abstract

Large language models have transformed the field of natural language processing with an outstanding ability to analyze and comprehend human texts. Recently, a popular approach for applying these models to various tasks is prompting, where we present a model with a prompt that guides it towards solving a particular task. This approach has achieved success on a variety of tasks and has led to the concept of prompt learning, where we fine-tune the language models on the prompt itself, leading to further success on many tasks.

In this work, we explore the use of prompting and prompt learning for sentiment analysis on movie reviews from the IMDB dataset. We conduct two experiments for our sentiment classification experiment. In the first experiment, we present a set of human-engineered prompts to a collection of language models along with the movie reviews, obtaining strong results. In the second experiment, we apply prompt learning by fine-tuning the selected language models on the prompts themselves. In this experiment, we achieved a state-of-the-art 98.53% accuracy with the Llama 2 model. We observe that all models achieve stronger results when we apply prompt learning, demonstrating the effectiveness of this approach.

In addition to the application of prompting and prompt learning, we also explore the field of Explainable AI (XAI). To the best of our knowledge, no existing work has applied XAI to prompt learning systems. We apply a variety of XAI methods to our prompt learning system to generate human-understandable explanations for the model predictions. We compare these XAI methods using a variety of metrics. We evaluate how well the explanations reflect the decision-making process using the Faithfulness-by-Construction pipeline, attaining a peak sufficiency of 77.05%. Through human evaluation, we also obtain an adequate justification rate of 75%, an understandability of 100%, and a trustworthiness of 86%.

This work contributes a novel prompt learning-based framework for sentiment analysis of movie reviews that achieves state-of-the-art results, along with the results of a comprehensive evaluation of a variety of language models for both prompting and prompt learning. This work also contributes a novel framework for applying XAI methods to prompt learning systems, along with a comprehensive evaluation of the quality of explanations generated by these methods. We also provide insights into the behaviours of various language models and the importance of effective prompt engineering for this task.

Acknowledgements

First, I would like to acknowledge the support of my supervisor, Dr. Diana Inkpen. Professor Inkpen put a lot of faith in my by agreeing to supervise both my undergraduate honours project and masters program. Throughout this experience, she has challenged me to try new experiences that have improved my academic career. She has constantly provided useful insights and feedback that significantly guided this work. I am very grateful for her hard work in supporting my research.

Thank you to all the participants of the biweekly NLP group chats for sharing their insights into NLP research and their feedback on my work. Their insights have been a significant boost for this work and have helped me improve on it constantly. In particular, thank you to those who contributed towards the end of my studies, especially during my practice thesis defence with the group.

I would also like to acknowledge my thesis evaluators for their time and effort in evaluating my work. The feedback provided has been invaluable in improving the quality of this work.

Throughout my academic career, my family has provided me with an endless supply of support. Thank you Rianna, Mom, Dad, Brianna, Stephen, and Dawn for all of your support. I am forever grateful for all of your encouragement and understanding to help me succeed.

And last but not least, to our three furry friends, Tilly, Artemis, and Jimmy, for the joy they've brought throughout my studies.

Contents

List of Tables

List of Figures

List of Acronyms

AI Artificial Intelligence

AUPRC Area Under Precision-Recall Curve

BERT Bidirectional Encoder Representations from Transformers

BLEU Bilingual Evaluation Understudy

DL Deep Learning

ELECTRA Efficiently Learning an Encoder that Classifies Token Replacements Accurately

FN False Negative

FP False Positive

GPT Generative Pre-trained Transformer

IMDB Internet Movie Database

IoU Intersection over Union

LIME Local Interpretable Model-Agnostic Explanations

LLM Large Language Model

ML Machine Learning

NLP Natural Language Processing

PLM Pre-trained Language Model

RoBERTa A Robustly Optimized BERT Pretraining Approach

RQ Research Question

SHAP SHapley Additive exPlanations

TN True Negative

TP True Positive

XAI Explainable Artificial Intelligence

Chapter 1

Introduction

In this chapter, we discuss the motivation behind this work and the importance of incorporating explainability into Artificial Intelligence (AI) systems. We discuss the objectives of the thesis through a set of research questions that will guide the research. We then discuss the key contributions of the thesis towards the fields of prompt learning, sentiment analysis, and explainable AI. Lastly, we provide an outline of the thesis structure.

1.1 Motivation

Movie reviews hold significant influence in shaping public perception of films, facilitating their commercial success by influencing decisions on whether to watch them or not (Yasen and Tedmori 2019). Analyzing these reviews with accuracy is therefore a valuable task for the entertainment industry, film critics, and moviegoers. Many moviegoers base their viewing choices on positive reviews of a movie. Likewise, the entertainment industry can adapt their work based on negative feedback. However, manually analyzing these reviews is a time-consuming and labour-intensive task; it is not feasible to analyze the vast amount of reviews that are produced for each film. Consequently, there is a need for automated systems that can analyze movie reviews and provide accurate insights into the sentiment expressed in the reviews.

Movie review sentiment analysis provides an interesting distinction from other fields, such as product reviews, as the opinions expressed in movie reviews are often more subjective. For example, a product review might be straightforward, such as expressing whether a product works as intended, whereas movie reviews are based on

individual preferences. Movie reviews also involve more creative expression to cover these distinctions, whereas a product review could be exclusively technical. Lastly, understanding movie reviews often requires contextual understanding of a plot or genre. For example, being scared is a good thing when it comes to a horror film, but not a good thing for a family-friendly movie.

Sentiment analysis of movie reviews has been widely explored with a wide variety of approaches, including traditional machine learning methods and deep learning methods. Despite these varied approaches, there is only limited research into the application of prompt-driven approaches using a Large Language Model (LLM). These novel approaches have achieved significant success in a wide variety of fields, including sentiment analysis, but have not yet been explored in the context of movie reviews. Additionally, LLMs remain black-box systems that lack transparency in how they reach their conclusions, leading to skepticism and mistrust in their predictions. Therefore, there is a need for explainable systems that can provide human-understandable explanations for their predictions.

The integration of explainability methods elevates the trustworthiness of AI systems, allowing users to understand how the system reached its conclusions and identify potential biases or limitations. Little to no research into explainability for prompt learning approaches has been conducted despite the pressing need for human-understandable explanations. The application of these Explainable Artificial Intelligence (XAI) methods offers an opportunity to unravel the "black-box" nature of these sophisticated LLMs and provide insight into their decision-making process. We will explore explainability for predictions generated by models to give these insights into this process.

With the rapid and accelerating growth of AI systems in recent years, there is a pressing need to ensure that these systems are understandable by humans. The findings of this research could have broad implications on the use of LLMs by providing a framework for explainable prompt learning systems. These findings could also have implications on the use of LLMs in other domains, such as legal applications, by providing a framework for explainable LLMs.

1.2 Thesis Objectives

While prompt learning has seen an explosion of research in recent years, there is still limited research into various domain-specific natural language processing (Natural Language Processing (NLP)) tasks with prompt learning. There are two primary

objectives of this thesis. First, we aim to create a novel system that performs sentiment analysis on movie reviews using prompt learning. Second, we aim to create an explainable system to allow the model to explain itself in a human-understandable format. Overall, we seek to address the following research questions:

- **Research Question 1:** How can we apply prompting techniques to movie sentiment classification?
- **Research Question 2:** What are the most suitable language models and architectures to perform prompt-based movie sentiment classification?
- **Research Question 3:** How does the performance of a prompt-based movie sentiment classification system compare to traditional Machine Learning (ML) and non-prompt-based language models?
- **Research Question 4:** How can XAI techniques be integrated into a prompt-based movie sentiment classification system to provide human-understandable explanations for model predictions?
- **Research Question 5:** What is the quality of the explanations generated by the XAI component and how can it be assessed?

These research questions provide a comprehensive framework for exploring the potential of prompt learning techniques and explainable AI in movie review sentiment analysis, enabling researchers to address the challenges and opportunities inherent in this domain. By investigating the effective application of prompts, suitable AI models, and integration of XAI, this research aims to advance the field by delivering a more accurate, transparent, and interpretable movie sentiment classification system.

1.3 Thesis Contributions

This thesis produces several significant contributions to the fields of Natural Language Processing (NLP) and Explainable Artificial Intelligence (XAI), with a focus on sentiment analysis of movie reviews. The key contributions of this thesis are as follows:

- **Development of a Novel Explainable Prompt Learning System:** This thesis introduces an innovative prompt learning system for movie review sentiment analysis that incorporates explainability methods to provide human-understandable explanations of model predictions. To the best of our knowledge, this is the first system to apply explainability methods to a prompt learning system, as well as the first system to apply prompt learning to movie review sentiment analysis.

- **Comprehensive State-of-the-Art Model Evaluation:** Using a variety of prompts, we evaluate a total of 12 LLMs, including state-of-the-art models like GPT-3.5, GPT-4 and Llama 2, as well as previous state-of-the-art models like BERT or RoBERTa. We provide detailed results for each model before and after a fine-tuning step using the prompts. This evaluation is performed on the widely-used IMDB dataset, ensuring the results are comparable to existing research. We provide a detailed breakdown of results using standard metrics for this dataset. Our proposed system achieves state-of-the-art results on this dataset, outperforming previous state-of-the-art models.

- **Application of Diverse Explainability Approaches:** We apply a variety of explainability methods to the proposed system, including Local Interpretable Model-Agnostic Explanations (LIME), SHapley Additive exPlanations (SHAP), and Integrated Gradients. In particular, we use several approaches to produce human-understandable explanations of model predictions using the results obtained by these methods. We evaluate the quality of the explanations generated by each method using an automated evaluation of sufficiency and a human evaluation of the understandability, trustworthiness, and adequate justification of explanations. This evaluation provides a comprehensive analysis of the quality of explanations generated by each method.

- **Foundational Work for Future Research:** LLMs and the use of prompting for NLP-related tasks remains a relatively new field of research. This thesis provides foundational work for future research in this field by providing a framework for explainable prompt learning systems. This framework can be applied to other NLP tasks, such as text classification or question answering. The approaches used in this work can also be used to evaluate the quality of explanations generated by XAI methods.

Through these contributions, this thesis strengthens the recent field of Prompt Learning, as well as the fields of Sentiment Analysis and Explainable AI and provides a strong foundation for future research at the intersection of these fields. It

also presents numerous avenues for further work, such as through the application of prompt learning on additional datasets, the use of automated prompt engineering to produce the prompts used, and further development of XAI systems.

1.4 Thesis Organization

The thesis is structured according to the following structure:

- **Chapter 2 - Background:** This chapter presents the background information required to understand the thesis. It contains high-level introductions to the topics of AI, NLP, and XAI, as well as more detailed information on the specific approaches used in this thesis.
- **Chapter 3 - Literature Review:** This chapter discusses the state-of-the-art research in the thesis topic. It begins with a review of standard approaches to sentiment analysis for movie reviews, followed by a review of prompt learning techniques with emphasis on prompt learning for text classification and sentiment analysis. It then provides an overview of XAI systems and current approaches for their evaluation. This chapter discusses what concepts can be applied to this thesis and the gaps that this thesis can address.
- **Chapter 4 - Methodology:** This chapter describes the proposed framework of the thesis. It begins with a general overview of the architecture of the proposed system. It discusses the experimental setup, including the environments used to conduct these experiments. It then discusses the datasets used to train and evaluate the models, as well as the splitting performed for these sets and the preprocessing steps applied to prepare these sets for analysis. It then provides an overview of the selected models, experiments, and evaluation for both the sentiment classifiers and the explainability portions of this thesis.
- **Chapter 5 - Experiments and Discussion:** This chapter focuses on the experiments conducted to create the proposed system. It provides a detailed description of the two experiments performed to create the sentiment classifier system and the results of each experiment. It also compares the results of these experiments to existing baseline results. It then focuses on the experiments conducted to create the XAI portion of the system, providing a detailed description of the experiments performed to create the system and the evaluation of different explanation methods. It concludes with a discussion of the

experiments and the findings. In particular, it reviews the research questions proposed earlier in this chapter and discusses the findings in the context of these questions. It also discusses the limitations of the research.

- **Chapter 6 - Conclusion and Future Work:** The final chapter of this thesis presents the conclusions and findings reached through this research, as well as potential avenues for future work.

Additionally, we include the following appendices:

- **Appendix A - Results for Sentiment Classifier Experiments:** This appendix contains the full results for each model evaluated during the sentiment classifier experiments.
- **Appendix B - Explainability Human Evaluation Process:** This appendix contains additional information about the human evaluation conducted for the explainability experiments.
- **Appendix C - Responsible NLP Checklist:** This appendix contains a completed version of the ACL 2023 Responsible NLP Checklist to ensure that this research is conducted in an ethical manner.

Chapter 2

Background

This chapter provides a brief introduction to the thesis topics. It begins with an overview of Artificial Intelligence (AI) and the sub-fields that are used in this research. It then provides an overview of NLP and the sub-fields of text classification and sentiment analysis, which are key concepts in this work, as well as some of the methods used to conduct NLP and to evaluate performance in NLP. Lastly, it provides an overview of XAI, the types of XAI systems available, and how they can be evaluated.

2.1 Artificial Intelligence

Lu 2019 describes AI as "any theory, method, and technique that helps machines (especially computers) to analyze, simulate, exploit, and explore human thinking process and behavior". Put simply, AI is a field that aims to create algorithms that can learn, think, and behave like a human would. The topic has gradually expanded to a very broad field that can be applied to a variety of different tasks, such as image classification, speech recognition, or text classification. This section will provide a brief overview of the sub-fields of AI that are relevant to this thesis.

2.1.1 Machine Learning

Machine Learning (ML) is a subset of AI that involves designing algorithms that automatically learn from inputs and produce their own predictions or decisions based

on those inputs (Zhou 2021). It is based on how humans learn from experiences and can rapidly make judgements or decisions based on our past experiences (Zhou 2021).

In principle, the idea of ML is to provide computers with a significant amount of sample data (known as training data) and use algorithms that can analyze the data and learn from it to make predictions or decisions on previously unseen data (known as test data). The learning process is referred to as *training*, and the resulting system that can make these predictions is referred to as a *model*. A wide variety of approaches have been proposed and used, such as simplistic rule-based approaches such as decision trees, to complex statistical approaches such as neural networks. These concepts can be applied in a variety of fields, such as NLP, computer vision, or medical diagnosis.

ML includes different methods, such as supervised learning, unsupervised learning, and reinforcement learning. These methods are described further in this section.

Supervised Learning

In supervised learning, the model is trained on a pre-labeled dataset that provides an input and an anticipated output. The objective of the system is to learn a way to map from the inputs to the anticipated outputs so that the model can make accurate predictions for unseen data later. An example is depicted in Figure 2.1, where a supervised learning model could identify that a trend line accurately represents the data points of a given class and groups them together.

Unsupervised Learning

In unsupervised learning, the model is trained on a dataset that does not provide any labels with anticipated results. The objective of the system is to learn a way to group the data into clusters based on similarities between the data points. This can be useful for identifying patterns in data that may not be obvious to humans. An example is depicted in Figure 2.1, where an unsupervised model would simply group similar data points together, identifying three distinct clusters.

Reinforcement Learning

In reinforcement learning, the model learns to make decisions by interacting with an environment and receiving feedback on its actions, such as a reward or a penalty. The objective is to learn a policy that maximizes its reward over time.

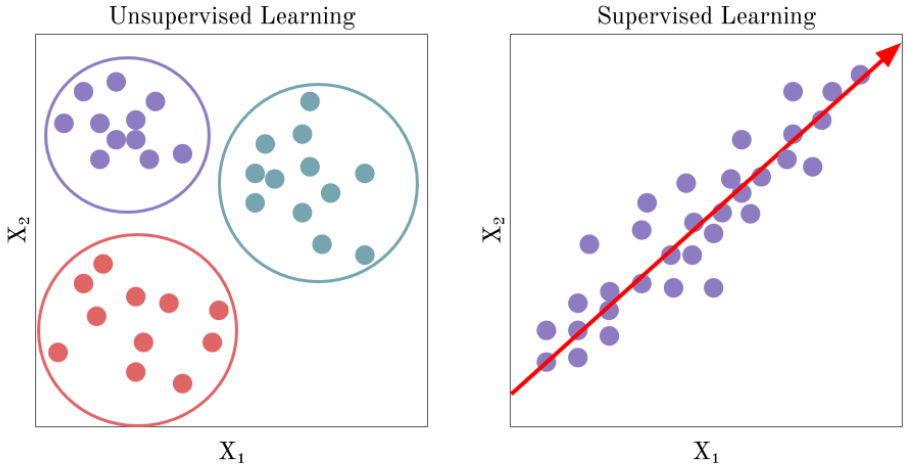

Figure 2.1: A depiction of results for unsupervised and supervised models, from Wilson 2020

2.1.2 Neural Networks

Neural networks are a class of machine learning models that are inspired by the structure and function of the human brain (Migliaccio 2023). A neural network is composed of interconnected nodes (or neurons) that constantly exchange information. Neurons are organized into layers, with each layer performing a different function. The first layer is the input layer, which receives the input data. The last layer is the output layer, which produces the final output of the model. The layers in between are referred to as hidden layers, which perform various functions to process the input data (Migliaccio 2023).

Each neuron receives inputs from either the input layer or a previous hidden layer in the network. Each input $x_1, x_2, ..., x_n$ is assigned a weight $w_1, w_2, ..., w_n$, which represents the strength of the connection between the neuron and the input. The node calculates the weighted sum following Equation 2.1. In some cases, a bias term is added to the weighted sum, allowing the node to shift its activation function along the input axis, as depicted in Equation 2.2 (Migliaccio 2023).

$$Sum = \sum_{i=1}^{n} w_i x_i \tag{2.1}$$

$$Sum = (\sum_{i=1}^{n} w_i x_i) + Bias \tag{2.2}$$

The weighted sum of the inputs is then used in an activation function by the node. The activation function determines the output of the node based on the weighted sum of the inputs. There are a variety of different activation functions that can be used, such as the sigmoid function, the hyperbolic tangent function, or the rectified linear unit function. The activation function is typically chosen based on the type of problem being solved, as well as the type of model being used. The output of the activation function is then passed to the next layer of the network (Migliaccio 2023).

To train a neural network, the weights and biases of the nodes are simply adjusted using an optimization algorithm and a loss function that calculates the difference between the predictions and actual target values. The objective is to minimize this loss function, which indicates that a network is making more accurate predictions.

2.1.3 Deep Learning

Deep Learning (DL) is a sub-field of ML that focuses on neural networks with multiple layers, which can also be referred to as deep neural networks. These networks are capable of learning very complex representations of data, allowing them to model complicated patterns and relationships in large datasets (Migliaccio 2023). DL has been applied to a variety of different fields, such as computer vision, speech recognition, and NLP. Some common types of DL models include:

- **Convolutional Neural Networks (CNNs):** These networks are designed to process grid-like data, such as images, using convolutional layers that apply filters to smaller, local regions of the input. CNNs are very successful in tasks like image classification, object detection, and segmentation.
- **Recurrent Neural Networks (RNNs):** These networks can process sequences of data by maintaining an internal state or memory, allowing them to learn temporal relationships in the input. RNNs are widely used in natural language processing, speech recognition, and time series analysis.
- **Long Short-Term Memory (LSTM)** and **Gated Recurrent Unit (GRU):** These are variants of RNNs that are designed to address the vanishing gradient problem, which harms the learning process in RNNs. They are widely used in natural language processing and speech recognition.

- **Transformers:** These models rely on the self-attention mechanism, which allows them to weigh and consider different parts of the input sequence when generating an output. Transformers have been particularly successful in natural language processing tasks, leading to the development of large-scale pre-trained models like Bidirectional Encoder Representations from Transformers (BERT) and Generative Pre-trained Transformer (GPT).

Transformer models were first proposed in Vaswani et al. 2017 and have since become the dominant architecture for NLP tasks. The key innovation of transformer models is the self-attention mechanism, which allows them to weigh and consider different parts of an input sequence when generating an output. This mechanism helps the model efficiently capture long-range dependencies and context within text data. The typical transformer architecture consists of two main components:

- **Encoder:** The encoder is a stack of identical layers, each containing a multi-head self-attention mechanism followed by a position-wise feed-forward network. The input text is first embedded into continuous vectors and combined with positional encodings to incorporate sequence information. The encoder processes this input representation and generates a continuous output representation that captures the contextual information of the input sequence (Vaswani et al. 2017).
- **Decoder:** The decoder is also a stack of identical layers, containing a multi-head self-attention mechanism, a position-wise feed-forward network, and an additional encoder-decoder attention layer. The decoder takes the output representation from the encoder and generates the final output, such as a translated sentence in machine translation tasks or a masked word in masked language modeling tasks.

Transformer models have achieved unprecedented success in a wide variety of NLP tasks, including machine translation, sentiment analysis, text summarization, and question-answering. Their success has led to the development of large-scale pre-trained models like BERT and GPT, which are described further in the following section.

2.1.4 Evaluation of AI Systems

AI systems are commonly evaluated using various methods to assess their performance and capabilities. This section will provide a brief overview of some of the

most common evaluation methods.

At the base of many metrics are True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN). These are defined as follows:

- **True Positive (TP):** A true positive is a result that was correctly predicted as positive by the model. For example, if a model correctly predicts that a patient has a virus, this would be a true positive.
- **True Negative (TN):** A true negative is a result that was correctly predicted as negative by the model. For example, if a model correctly predicts that a patient does not have a virus, this would be a true negative.
- **False Positive (FP):** A false positive is a result that was incorrectly predicted as positive by the model. For example, if a model predicts that a patient has a virus when they are actually virus-free, this would be a false positive.
- **False Negative (FN):** A false negative is a result that was incorrectly predicted as negative by the model. For example, if a model predicts that a patient does not have a virus when they actually have a virus, this would be a false negative.

These metrics are often used to construct a *confusion matrix*, which is a table that summarizes the performance of a classification model. An example of a confusion matrix is shown in Table 2.1. In this example data, the model correctly predicted 67 positive examples and 54 negative examples, while incorrectly predicting 13 negative examples (false positives) and 6 positive examples (false negatives). These values will be used to calculate the metrics described below.

Table 2.1: Example of a confusion matrix

		Actual Class	
		Positive	**Negative**
Predicted Class	**Positive**	67	13
	Negative	6	54

Accuracy

Accuracy measures how well an AI system performs on a specific task compared to a ground truth or human-labeled data. It calculates the percentage of correct

predictions made by the system based on the overall number of guesses. Using the measures defined above, accuracy can be calculated using Equation 2.3.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.3}$$

While accuracy is a simple and intuitive metric, it can be misleading in some cases. For example, if a model is trained on a dataset that contains 90% positive examples and 10% negative examples, it can achieve 90% accuracy by simply predicting that every example is positive. This is problematic because the model is not actually learning anything useful and is simply predicting the majority class. Therefore, accuracy is not always a good metric to use, especially when the dataset is imbalanced.

Using the measures in the confusion matrix shown in Table 2.1, we can also calculate the accuracy as an example, achieving an accuracy of 0.864 or 86.4%.

$$Accuracy = \frac{67 + 54}{67 + 54 + 13 + 6} = 0.864 \tag{2.4}$$

Precision and Recall

Precision measures the proportion of true positive results out of all positive predictions made by the system, while recall measures the proportion of true positive results out of all actual positive instances. These metrics are often used in classification tasks where identifying true positives and avoiding false positives or false negatives is important. Precision and recall can be calculated using Equations 2.5 and 2.6.

$$Precision = \frac{TP}{TP + FP} \tag{2.5}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.6}$$

Using the measures in the confusion matrix shown in Table 2.1, we can also calculate the precision and recall as an example, achieving a precision of 0.837 or 83.7% and a recall of 0.918 or 91.8%.

$$Precision = \frac{67}{67 + 13} = 0.837 \tag{2.7}$$

$$Recall = \frac{67}{67 + 6} = 0.918 \tag{2.8}$$

F1-score

The F1-score is a metric that combines precision and recall into a single score. It is calculated using Equation 2.9.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.9}$$

F1-score can be applied in two key ways, referred to as micro-averaging and macro-averaging. Micro-averaging calculates the F1-score using the total sum of the TP, FP, and FN for all classes, while macro-averaging calculates the F1-score for each class and then averages them together. Micro-averaging is useful when the dataset is imbalanced, while macro-averaging is useful when each class is equally important.

Since the sample results in the confusion matrix shown in Table 2.1 consist of only one class, the micro-averaged F1-score and macro-averaged F1-score are the same. Using the measures calculated for precision and recall, we can calculate the F1-score as an example, achieving an F1-score of 0.875 or 87.5%.

$$F1 = 2 \times \frac{0.837 \times 0.918}{0.837 + 0.918} = 0.875 \tag{2.10}$$

2.2 Natural Language Processing

Natural Language Processing (NLP) is a sub-field of AI that focuses on the interaction between computers and human languages. It involves developing algorithms and models that allow computers to understand, interpret, and use natural language text in a meaningful way. Figure 2.2 depicts the relationship between NLP and the concepts seen before, namely AI, ML, and DL. Note that this figure exclusively illustrates the relationships relevant to this work, there are a wide variety of other sub-fields of AI, such as computer vision or robotics. We can observe that NLP is

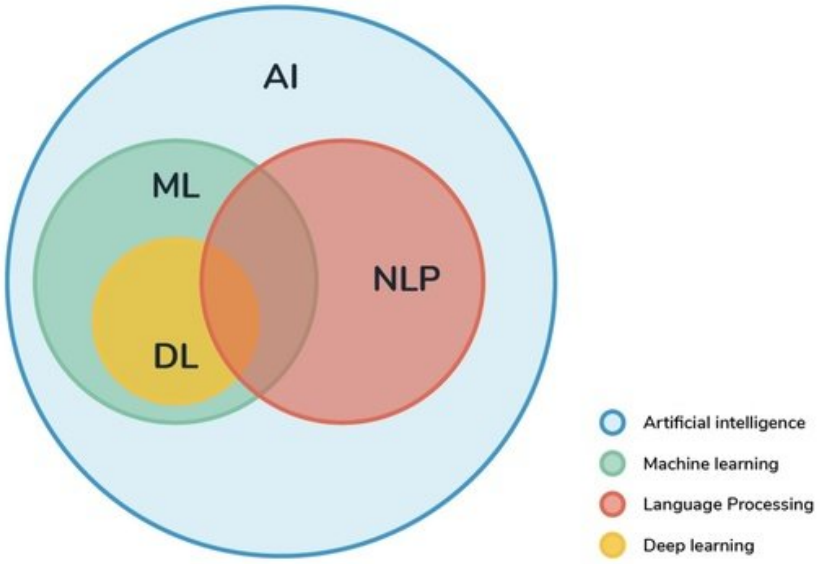

Figure 2.2: Relationship between AI, ML, DL, and NLP, from Mehra and Hasanuzzaman 2020

a sub-field of AI and sometimes uses ML and DL techniques to accomplish its tasks (Mehra and Hasanuzzaman 2020).

NLP encompasses a wide variety of tasks and applications, such as:

- **Sentiment Analysis:** Determining the sentiment or opinion expressed in a piece of text, such as positive, negative, or neutral.
- **Machine Translation:** Automatically translating text from one language to another.
- **Named entity recognition (NER):** Identifying and classifying entities such as people, organizations, locations, or dates within a text.
- **Text summarization:** Generating a concise and meaningful summary of a longer text while preserving its key information and context.
- **Question-answering systems:** Providing accurate and relevant answers to user questions based on a given text, corpus, or knowledge base.

In this work, we focus on the task of sentiment analysis, which is described further in the following section. NLP relies on various techniques and approaches

from linguistics, computer science, and machine learning to analyze and model the structure, meaning, and context of language data. With the advent of deep learning and large-scale pre-trained models like BERT and GPT, NLP has achieved significant progress in recent years, leading to increasingly sophisticated and accurate language understanding and generation capabilities in AI systems.

2.2.1 Sentiment Analysis

Sentiment analysis, sometimes also known as opinion mining, is an NLP task that involves determining the sentiment or emotion expressed in a piece of text. Typically, this involves classifying a given text as positive, negative, or neutral. sentiment analysis is an important task that provides significant benefits to a variety of different applications, such as allowing companies to aggregate customer reviews to help improve product offerings, or allowing governments to analyze public opinion on a given topic (Moreno 2020).

In recent years, DL methods like CNNs, RNNs, and transformer-based models have achieved significant success in sentiment analysis tasks. Transformer-based models will be the focus of this thesis, and they are described later in this section as part of the background on language models.

2.2.2 Text Preprocessing

Preprocessing of text in NLP involves a variety of techniques used to clean, normalize, and prepare text data for further analysis or modeling. This is necessary because raw text can often be noisy, unstructured, and contain inconsistencies that can negatively impact models or algorithms. Some common preprocessing techniques include, but are not limited to:

- **Tokenization:** This is the process of breaking down a text into individual words or units, known as tokens. This converts data to a more structured format for analysis. For example, "You're happy" would be tokenized into "You", "'re", and "happy".
- **Stopword removal:** This involves removing common words that do not provide any useful information for analysis, such as "the", "a", or "an". Removing these words often helps focusing on more important words.

- **Lemmatization:** Lemmatization uses linguistic knowledge to convert words to their base or dictionary form (lemma). For example, "running" would be converted to "run".

- **Removing HTML or URLs:** Raw text data from the Internet often contains HTML tags, URLs, or other web-specific elements that are not useful for analysis. These can be removed to clean the data.

The choice of preprocessing techniques depends on the specific NLP task, the characteristics of the data, and the requirements of the downstream algorithms or models. Proper text preprocessing can significantly improve the performance and accuracy of NLP applications, although not all of these steps are required for advanced NLP models. The specific steps used in this thesis will be discussed in Section 4.

2.2.3 Language Models

A language model is a statistical model used in NLP to predict the likelihood of a sequence of words or tokens occurring in a language. An objective of a language model is to estimate the probability distribution over sequences of words, which can be used to generate new text or evaluate the likelihood of a given sequence of words. Language models have quickly become a fundamental component of many NLP tasks, such as machine translation, text summarization, and question-answering. There are several types of language models:

- **N-gram Models:** These models predict the probability of a word based on the previous $n-1$ words. For example, a bigram model would predict the probability of a word based on the previous word, while a trigram model would predict the probability of a word based on the previous two words. These models are simple and computationally efficient, but their ability to model long-range dependencies is extremely limited.

- **Neural Language Models:** These models use neural networks to learn the probability distribution of word sequences. Neural language models can capture more complex relationships and long-range dependencies in the data compared to N-gram models but often require larger amounts of data and computational resources for training.

- **Transformer-based Language Models:** These are large-scale, deep learning-based neural language models, such as BERT and GPT, which have achieved significant success in recent years. Transformer models utilize self-attention mechanisms to effectively capture context and dependencies in text data. They are often pre-trained on vast amounts of unsupervised text data and can be fine-tuned on specific tasks with smaller labeled datasets, leading to strong performance across a wide range of NLP applications.

Language models are critical components in the development of advanced NLP systems, and ongoing research in this area continues to drive improvements in language understanding and generation capabilities.

Transformer-based Language Models

The structure of transformer-based language models was discussed in the previous section on Deep Learning. This section will further discuss some of the most popular transformer-based language models, which are BERT and GPT.

- **Bidirectional Encoder Representations from Transformers (BERT):** This model was proposed in Devlin et al. 2019 and has since become one of the most popular and widely used language models. It is a multi-layer bidirectional transformer encoder that is pre-trained on vast amounts of unlabeled text data. It can be fine-tuned on specific tasks with smaller labeled datasets, leading to strong performance across a wide range of NLP applications. It has also been used as a basis for many other transformer-based models, such as A Robustly Optimized BERT Pretraining Approach (RoBERTa) (Yinhan Liu et al. 2019) and DistilBERT Sanh et al. 2019.
- **Generative Pre-trained Transformer (GPT):** GPT was proposed in Radford, Narasimhan, et al. 2018 is a unidirectional transformer model pre-trained using a next-word prediction task. It is designed to generate text by predicting the next word in a sequence given the previous context. GPT and its successors, GPT-2 (Radford, J. Wu, et al. 2019), GPT-3 (Brown et al. 2020), and GPT-4 (OpenAI 2023), have demonstrated impressive text generation capabilities and have been applied to tasks like machine translation, summarization, and code generation. In particular, GPT has gained significant worldwide attention for its use in ChatGPT, a powerful chatbot that can generate human-like responses to user messages (Yiheng Liu et al. 2023).

- **Llama:** Llama was proposed in Touvron et al. 2023 and is an auto-regressive language-optimized transformer that is enhanced through supervised fine-tuning and reinforcement learning. It has since been developed into Llama 2 by Meta. The Llama 2 model has achieved strong results in various benchmarks for reasoning, coding, and knowledge. It can also be further fine-tuned to achieve strong results on specific tasks and has various sizes available, ranging from 7 billion to 70 billion parameters.
- **Efficiently Learning an Encoder that Classifies Token Replacements Accurately (ELECTRA):** ELECTRA was proposed in Clark et al. 2020 and is a transformer model that uses a generator-discriminator setup, where the generator replaces some tokens with possible alternatives and the discriminator aims to distinguish between the original and generated tokens. By training the discriminator to be effective in this setup, ELECTRA achieves more efficient and accurate representation learning, making it a powerful tool for a variety of natural language processing tasks.

Pre-trained transformer models (or PLMs) have significantly advanced research in NLP and are used in a wide variety of applications, such as search engines, chatbots, and content generation tools. Despite their success, these models require large amounts of data and computational resources for training, making them reliant on powerful hardware such as GPUs and specialized accelerators like TPUs. However, once a Pre-trained Language Model (PLM) is trained, it can be fine-tuned on specific tasks using smaller labeled datasets. Fine-tuning allows the model to adapt its general language understanding to the target task, leading to strong performance across various NLP applications. This is one of the most popular approaches to NLP in current research.

2.2.4 Prompt Learning

With the introduction of many large-scale PLMs in recent years, there has been a surge of research into prompt learning, which is considered "a new paradigm in modern natural language processing" (Ding et al. 2022). In prompt learning, the language model is conditioned on a specific text input, called the prompt, which is designed to guide the model towards generating the desired output for a task (Lester, Al-Rfou, and Constant 2021, Li and Liang 2021).

Prompts typically contain examples, instructions, or questions that help the model understand the context and objective of the task. The model learns to generate

appropriate responses to these prompts by leveraging its vast knowledge of language and patterns learned during pre-training. An illustration is provided in Figure 2.3, which depicts three different approaches to sentiment analysis on movie reviews. (c) depicts a prompt-based approach using the template "It was [MASK]". The model is conditioned on this prompt and learns to fill in the blank with the correct sentiment, such as "It was great" or "It was terrible". This approach is more efficient than the pre-training of a masked language model, as shown in (b), or regular fine-tuning, as shown in (b), because it only requires limited fine-tuning on the data. It also allows the model to be easily adapted to different tasks by simply changing the prompt.

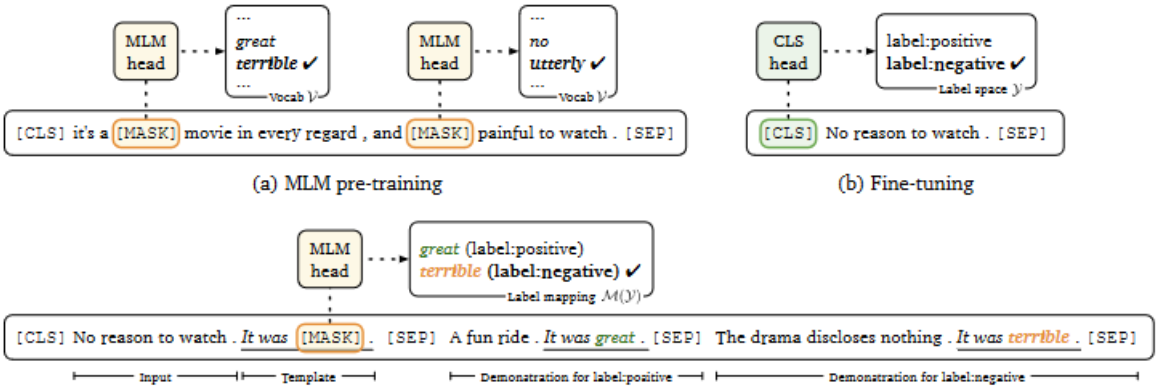

Figure 2.3: An illustration of (a) masked language model (MLM) pre-training, (b) standard fine-tuning, and (c) LM-BFF using prompt-based fine-tuning with demonstrations (Gao, Fisch, and Chen 2021).

Prompt learning provides several key advantages:

- **Efficiency:** By utilizing the general knowledge encoded in pre-trained models, prompt learning requires fewer task-specific labeled examples to adapt the model to new tasks, reducing the need for large labeled datasets.
- **Flexibility:** Prompt learning allows the same PLM to be easily adapted to different tasks by changing the prompt, allowing quick deployment in a variety of applications.
- **Simplicity:** The process of fine-tuning with prompts is often simpler and more interpretable than designing custom architectures or training models from scratch, leading to easier debugging and maintenance. Even more simple is that

language models may need no fine-tuning to be used for a task, which is referred to as *zero shot learning*.

Creating effective prompts can be challenging and sometimes requires domain knowledge of expertise. Regardless, it has become a popular method for applying PLMs for NLP tasks and achieves strong performance in these tasks. Additionally, recent research has demonstrated that prompts can be automatically generated using a variety of techniques, such as templates, keywords, or natural language generation.

Difference Between Prompt Learning, Prompt Tuning, and Prompt Engineering

The idea of using prompts with LLMs is explored in multiple different ways, including the previously discussed prompt learning. However, there are two other related concepts that are important to distinguish: prompt tuning and prompt engineering:

- **Prompt Tuning:** Prompt tuning refers to the process of fine-tuning a PLM to respond more effectively to certain prompts, such as by adjusting the model's parameters to optimize responses to a set of prompts. The focus is on adjusting the model or its inputs to optimize these responses.
- **Prompt Engineering:** This is a form of feature engineering applied to prompts, involving crafting, selecting, or transforming prompts to be most effective for a given task or model. The focus is on designing the prompts themselves to optimize responses.

These are in contrast to Prompt Learning, where the focus is on the learning process of the AI model itself, with the model learning how to respond to various prompts based on its training data. The focus of this work is on prompt learning, as we will be training the model using a variety of prompts. However, we also apply prompt engineering to design the prompts that we will use for training. This process is discussed in the Methodology chapter, and the results of Experiment 1 demonstrate the impacts of differently engineered prompts.

2.3 Explainable AI

Explainable Artificial Intelligence (XAI) refers to the development of AI systems that are understandable to humans. This means that the system can explain its

inner workings and decision-making processes in a way that is easy for humans to understand, especially without a technical background.

The need for XAI exists because many modern AI systems are referred to as *black boxes*, which means that their internal workings are complex and difficult to understand. This is especially true for DL models discussed in earlier sections, which use complicated neural network systems. This makes it difficult for humans to understand how the model makes its decisions, which is problematic since it can lead to distrust and skepticism from users, as can be observed in Figure 2.4. This is especially problematic in fields such as medicine or law, where the decisions made by AI systems can have significant impacts on people's lives. Therefore, the field of XAI seeks to design systems that are more transparent and can justify their decisions, leading to increased trust and acceptance of AI systems and the results they provide. XAI models can often be integrated with existing ML models to provide explanations for their decisions, as depicted in Figure 2.4.

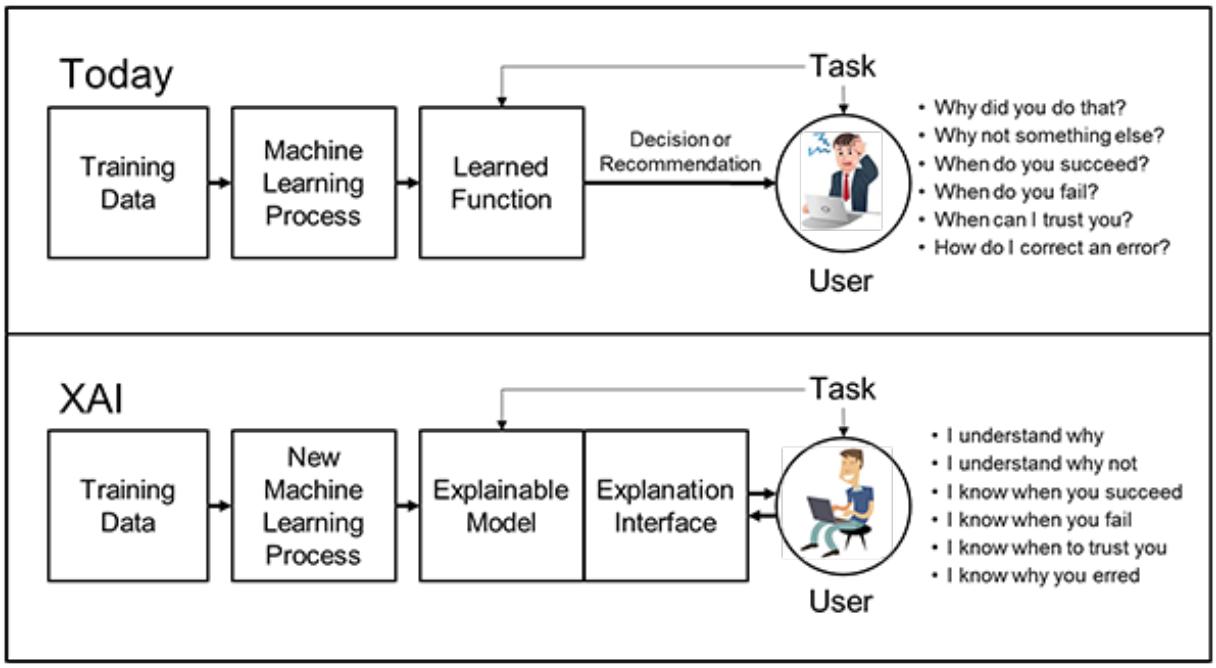

Figure 2.4: An illustration of how an XAI system is integrated with an ML model and its impact on users (Turek n.d.).

XAI systems are typically categorized into whether they provide local or global explanations (Vale, El-Sharif, and Ali 2022):

- **Global Explanations:** Provide an overall understanding of how the AI system works and why it makes the decisions it does. These explanations are usually provided at a high level, and are designed to give users a broad understanding of the decision-making process. Global explanations are useful for providing an overview of the system, but may not be detailed enough to understand specific decisions.

- **Local Explanations:** Explain individual decisions made by the AI system. These explanations provide a detailed understanding of how the system arrived at a particular decision, and can help users to understand the factors that influenced the decision. Local explanations are particularly useful for high-stakes applications where it is important to understand why a particular decision was made.

XAI systems can also be categorized based on whether they are self-explaining or post-hoc explainers (Vale, El-Sharif, and Ali 2022):

- **Post-hoc Explainers:** Designed to explain the decisions made by complex machine learning models that are inherently opaque. These systems work by analyzing the behaviour of a model and generating explanations for its decisions after the fact. Post-hoc explainers can be applied to any type of machine learning model, but they may not always provide a complete understanding of the decision-making process.

- **Self Explainers:** Designed to be inherently transparent and understandable to humans. These systems are often rule-based, and are designed to provide clear, understandable reasons for the decisions they make. For example, a decision tree allows us to directly follow a given input to an output by following the given conditions. Self-explaining AI systems are often used in applications where transparency and accountability are important, such as in healthcare or finance.

This section will provide a brief overview of XAI and some of the methods used in XAI.

2.3.1 Post-hoc Explainers

As described in the preceding section, a post-hoc explainer is applied after a model has been trained to explore its decision-making process. In particular, they do not

require access to the inner workings of a model, which can be useful for certain closed-source models where that access is unavailable.

LIME (Ribeiro, Singh, and Guestrin 2016) and SHAP (Lundberg and S.-I. Lee 2017) are two popular techniques in XAI that aim to provide human-understandable explanations for predictions of complicated ML models. Other approaches that have been used in text classification are Integrated Gradients, which uses a gradient-based approach to measure feature importance (Sanyal and Ren 2021), and surrogate models, which use an inherently explainable model like a decision tree to approximate the behaviour of a complicated model.

Local Interpretable Model-Agnostic Explanations (LIME)

Proposed in Ribeiro, Singh, and Guestrin 2016, LIME brings the idea of approximating the behaviour of a complicated model with a simpler model that works on a local level. This is done by taking a sample of interest and creating many small perturbations of it, collecting their corresponding predictions using the complicated model. Using these predictions, it can fit a simple, interpretable model that can approximate the complicated model's behaviour for that sample. Using this simple model, it can determine the importance of each token from the original sample and measure how much it contributes to the prediction.

SHapley Additive exPlanations (SHAP)

Proposed in Lundberg and S.-I. Lee 2017, SHAP brings a way to measure feature importance for ML models, based on the concept of Shapley Values (Shapley et al. 1953) in game theory. Shapley Values provide a fair evaluation about how much each player in a game contributed to the outcome of the game. For machine learning, the features of the model would be the "players", and the prediction would be the "outcome". SHAP provides a way to measure the importance of each feature to the prediction, which can be used to explain the model's predictions.

Integrated Gradients

Integrated Gradients (Sundararajan, Taly, and Yan 2017) is a post-hoc explanation method that works by measuring feature importance based on the model's average output gradient with respect to the input (Sanyal and Ren 2021). It uses the partial

derivatives of the outputs according to each input feature as they are interpolated along a path from the given input to a baseline value. For example, if the input is a sentence, the baseline value could be a sentence with all words removed. By calculating the gradients along this path, it can determine the importance of each feature to the model's prediction.

Surrogate Model

The use of surrogate models incorporates a self-explaining system, such as those described in the following subsection. The objective is to approximate the behaviour of a model that is difficult to explain using a model that is self-explaining, like a decision tree. For example, you can train the decision tree using a neural network's predictions so that it can approximate the behaviour of the neural network and create explanations for its predictions.

2.3.2 Self-Explainers

Self-explainers are models that have been specifically designed to be understood by humans. In particular, the model must be trained with the ability to provide these explanations. Some examples of self-explaining models are the following:

- **Decision Tree:** A decision tree consists of a structured system of decisions, each based on specific features, leading to a final prediction. We can specifically analyze each decision of the decision tree for an input, making it easy to follow the model's decisions.
- **Rule-based Models:** A rule-based model consists of a composed set of rules that map specific input features to outcomes. Each rule implies a decision based on one or more features, which can be easily understood.
- **Bayesian Networks:** A Bayesian network is a graphical model that represents the probabilistic relationships between variables. The structure can reveal relationships, allowing for a natural explanation of how variables influence each other to determine the final outcome.

Unfortunately, the ability of language models to perform self-explaining is more limited. Unlike post-hoc systems, the same set of systems cannot be applied to a wide variety of models. The next chapter, on literature review, will explore recent works to add self-explaining abilities to language models.

2.3.3 Evaluation of Explainable AI Systems

One of the significant challenges posed to XAI is the evaluation of XAI systems. This is because there is no clear way to evaluate the quality of explanations, as it is a subjective task typically without any "ground truth" for interpretability. One particular concern is stated in Danilevsky et al. 2020, which indicates that papers have almost no standardized system for evaluating XAI systems and often include only an informal evaluation of the outputs. This is problematic because it makes it difficult to compare different XAI systems, as there is no standardized way to evaluate them.

Typical quantitative evaluation of an XAI system involves the use of fidelity, which measures how well an explanation represents the model's behaviour, consistency, which examines the similarity of explanations for similar inputs, stability, which assesses how much the explanation is changed by small changes in input, and coherence, which measures whether the explanations align with human expectations. Qualitative evaluation is also frequently applied, where users may be asked to perform tasks with the system and provide feedback on the explanations.

Proposed evaluation metrics for XAI are discussed further in the literature review, which will provide the guideline needed for this thesis to evaluate the quality of explanations generated by the proposed system.

2.4 Summary

In this chapter, we discussed the background of the topics relevant for this thesis. We began with a discussion of AI, where we introduced the concepts of ML. As part of this discussion, we explored how neural networks, a key ML model, work and how massive neural networks have led to the field of DL.

We then discussed the field of NLP and how it intersects with the previously viewed topics of AI, ML, and DL. We introduced the task of sentiment analysis and other key concepts of NLP that will be used throughout the thesis. In particular, we introduced the field of prompt learning, which is explored in-depth throughout this thesis.

We concluded with a discussion of the field of XAI and the types of explainable systems currently used for AI models. We also discussed the challenges posed to XAI and how they can be evaluated. The key concepts explored in this chapter are essential for the development of this thesis.

Chapter 3

Literature Review

This chapter reviews the state-of-the-art approaches for sentiment analysis for movie reviews sentiment analysis, prompt learning, and XAI. It discusses the methods that have previously been employed to perform sentiment analysis on movie reviews. It also explores prompt learning methods and some of their recent applications in related topics. Lastly, it discusses explainable AI methods for NLP and how they can be evaluated. This literature review provides a strong basis on all the relevant topics of this thesis, which we can build upon to propose a novel system that integrates all the relevant topics.

3.1 Sentiment Analysis of Movie Reviews

Sentiment analysis is a significant topic in NLP, as it is a common task that can be applied to a variety of different domains, such as e-commerce or movie reviews (S. Yang et al. 2020). Platforms frequently allow users to rate content and provide written commentary about these reviews, which can easily be used in NLP for sentiment analysis datasets. Such datasets have been produced using Amazon for e-commerce reviews in Guan et al. 2016 and using Internet Movie Database (IMDB) for movie reviews in Maas et al. 2011. These datasets are frequently used for sentiment analysis tasks, as they are publicly available and provide a large amount of data for training and testing. This section will provide a brief overview of the state-of-the-art research into sentiment analysis on movie reviews, which will provide a strong basis for our proposed system.

Maas et al. 2011 proposed the IMDB dataset used in much of the research into movie review sentiment analysis and used it with a model that combines unsupervised and supervised techniques to learn word vectors that capture semantic term-document information (Maas et al. 2011). The dataset contains 50,000 reviews from IMDB with a maximum of 30 reviews taken from each movie. In particular, the dataset contains an even number of positive and negative reviews, with positive reviews defined as anything above 7/10 and a negative review as anything under 4/10. This ensures that strong sentiment is expressed, as ratings between 4 and 7 are more neutral. This dataset is frequently used for sentiment analysis tasks, as it is publicly available and provides a large amount of data for training and testing, and will be discussed further in Section 4. This paper demonstrates that sentiment analysis on movie reviews is a relevant topic for research and provides a strong dataset that can be used for this task.

Due to the popularity of the IMDB dataset, it is frequently used as a benchmark for newly trained models. Numerous models use it to demonstrate baseline results of their proposed systems, such as DistilBERT (Sanh et al. 2019), Longformer (Beltagy, Peters, and Cohan 2020), and PoWER-BERT (Goyal et al. 2020). Furthermore, there is extensive work in the literature on models being fine-tuned on the IMDB dataset, such as Sun et al. 2019 that fine-tunes the original BERT model proposed in Devlin et al. 2019, and Tan et al. 2022 that fine-tunes the RoBERTa model proposed in Yinhan Liu et al. 2019. These papers demonstrate a variety of common approaches to sentiment analysis on movie reviews, which provides a strong basis for the proposed system. Additionally, these papers provide an extensive set of results for a variety of systems that can be used in the evaluation of the proposed system.

As of the writing of this thesis, the best-performing model on the IMDB dataset, as reported on PapersWithCode [1] as of December 2023, is a model called "XLNet" proposed in Z. Yang et al. 2020. This model is a generalized autoregressive pretraining method that uses permutation language modeling to learn bidirectional contexts. Put simply, it is based on BERT that retains the connections of masked words. This model achieves a test accuracy of 96.21% on the IMDB dataset, which is the highest reported accuracy on this dataset. This paper demonstrates a potential model that this thesis could use to demonstrate sentiment analysis on movie reviews, as well as a model with which we can effectively compare our work.

BERT-based LLM are also well represented on the PapersWithCode leaderboard for the IMDB dataset. The second place results achieves an accuracy of 96.2% by using the RoBERTa Large model with a concept called Heinsen Routing, which "takes

[1] https://paperswithcode.com/sota/sentiment-analysis-on-imdb

a sequence of vectors and computes a new sequence with specified length and vector size" (Heinsen 2022). This is a strong approach, achieving state-of-the-art accuracy on the IMDB sentiment analysis task. Bingyu and Arefyev 2022 uses the RoBERTa base model, which is trained on fewer parameters than the RoBERTa Large model, but still achieves an accuracy of 95.94% using a logistic regression classifier trained on document vectors with cosine similarity. These two papers demonstrate a strong performance using RoBERTa, but BERT Large is also extensively used, as demonstrated in Xie et al. 2020 where they use unsupervised data augmentation, achieving a result of 95.8%, and Sun et al. 2020, which fine-tunes BERT Large and achieves a 95.79% result. Overall, these papers demonstrate an outstanding performance by BERT-based models on the IMDB dataset, which provides a strong basis for their evaluation as part of this thesis.

While not reported on the PapersWithCode leaderboard, GPT models have also been applied to the IMDB dataset. One model available on HuggingFace is a GPT-2 model that was fine-tuned for a single epoch on the IMDB set[2], although they do not report any results and we found no papers that used this model. Saini 2023 demonstrates sentiment analysis on the IMDB dataset using both GPT-3 and GPT-3.5, but again they do not report any results. Overall, these two papers demonstrate that there is some work on analyzing the IMDB set with GPT models, but there is a gap in actual published results for these models. This is a gap that this thesis can attempt to fill by evaluating various GPT models for this purpose, especially the state-of-the-art GPT-3.5 and GPT-4 models.

These papers demonstrate the state-of-the-art systems being applied to movie review sentiment analysis, which provides a strong basis for the proposed system. In particular, these papers provide useful benchmarks for us to compare against, which will be particularly helpful for addressing Research Question 3 which investigates how these models compare against the proposed system.

3.2 Prompt Learning

As described in the previous chapter, a standard approach to many NLP tasks involves taking an input x that contains a given text and predicting an input y based on a given model (P. Liu et al. 2021). These labels can represent a variety of different tasks, such as a mood for sentiment analysis, a summary for text summarization, or a class for text classification. One of the challenges that P. Liu et al. 2021 identifies

[2]https://huggingface.co/lvwerra/gpt2-imdb

with this model is that it is necessary to have labelled training data for this task, which can be challenging to find or create for many NLP tasks. Prompt learning attempts to solve this issue by instead using a pre-trained language model that models a probability for the text x itself and then uses that probability to predict y (P. Liu et al. 2021). This section will provide a brief overview for the state-of-the-art research into prompt learning, along with a sub-section that will focus on the state-of-the-art research into prompt learning for text classification and sentiment analysis.

Ding et al. 2022 acknowledges that "prompt-learning has become a new paradigm in modern natural language processing" and attempts to implement a standard framework for prompt learning with the intention of making it faster for users to deploy prompt-learning systems and evaluate them on different NLP tasks. They implement a Python system called OpenPrompt, which supports loading PLMs directly from the HuggingFace Transformers collection (Wolf et al. 2020). The system provides many useful functions for prompt learning under a unified standard, such as templating, verbalizing, or optimization. Ding et al. 2022 says that this is a research-friendly framework and greatly simplifies the effort required to implement prompt-learning systems. This offers a strong framework that could be used to implement prompt-learning systems for movie review sentiment classification.

Xia et al. 2022 demonstrates prompt learning on a wide variety of tasks, including sentiment classification tasks and natural language inference tasks. In particular, they evaluate the ELECTRA PLM (Clark et al. 2020) for these tasks against BERT (Devlin et al. 2019) and RoBERTa Yinhan Liu et al. 2019. While BERT and RoBERTa are masked language models, meaning they are trained by masking words in inputs and maximizing the probability of original tokens replacing them, ELECTRA is a discriminative language model, meaning it adapts the word prediction problem into a binary classification problem. Xia et al. 2022 finds that ELECTRA achieves superior results compared to both masked language models. This paper demonstrates a potential language model in ELECTRA that this thesis could use to demonstrate prompt learning in movie review sentiment analysis, as well as models with which it can be effectively compared.

While there has been a recent surge in research into prompt learning, as demonstrated in the previous papers, Webson and Pavlick 2022 demonstrates a potential concern about the usefulness of prompt learning, as they find "that models can learn just as fast with as many prompts that are intentionally irrelevant or even pathologically misleading" as they can with good prompts (Webson and Pavlick 2022). This paper contradicts "a hypothesis commonly assumed in the literature that models use prompts as semantically meaningful task instructions in ways analogous to humans"

(Webson and Pavlick 2022), a contradiction that should be concerning for research into prompt learning. To consider this impact, we can analyze the prompts being used and determine whether models for movie review text classification learn even with misleading or false prompts.

These papers demonstrate the general state-of-the-art research into prompt learning, which can be used to guide our research on movie review sentiment analysis. The following subsection will consider prompt learning for the specific tasks of text classification and sentiment analysis.

3.2.1 Prompt Learning for Text Classification and Sentiment Analysis

This sub-section will provide a brief overview of the state-of-the-art research into prompt learning for general text classification and sentiment analysis, which we can then adapt to use for movie review sentiment classification.

One of several tasks demonstrated in Su et al. 2022 is sentiment analysis on the IMDB dataset, similar to the objective of this thesis. This paper performs cross-task transferability across a total of seventeen tasks, including the IMDB sentiment analysis task, to "analyze the transferability of prompts across different tasks and models" (Su et al. 2022). Their research focuses on the impact of transferability, finding that trained soft prompts can effectively transfer to similar tasks and that trained soft prompts of similar tasks can significantly speed up the training process. This paper is one key demonstration of prompt learning on movie review sentiment analysis, encouraging us to apply prompt learning to movie review sentiment analysis.

Cross-domain sentiment analysis is explored in H. Wu and Shi 2022. They identify a concern that is relevant for sentiment analysis on movie reviews, which is where some reviews may not specifically indicate a "good" or "bad" sentiment. In their example, they find that a high-frequency sentiment label for a book review is "useful", which is not equivalent to a "good" or "bad" sentiment. To address this issue and improve on cross-domain sentiment analysis with PLMs, they use soft prompts that are composed of multiple learnable vectors and make use of the [MASK] token. When performing on an Amazon review dataset, they found that their method obtains an accuracy of 93.14%, which improves on baseline results. This is a similar project to movie review sentiment analysis, as they are handling user reviews of a product (in our case, movies). This paper demonstrates that prompt learning can be used to improve on sentiment analysis tasks, as well as a potential approach that this thesis

can consider. These results are promising for the use of prompt learning for movie review sentiment analysis.

Deng et al. 2022 applied prompt learning for classifying patronizing and condescending language as part of SemEval-2022 Task 4, achieving the strongest results on the shared task. They use a very simple approach by including the prompt *"It is patronizing or condescending? [MASK]"*, then determines whether words associated to *YES* are mostly likely to replace the mask or not. This paper demonstrates that prompt learning is a strong approach for text classification by outperforming all other submissions with a very simple prompt. This is promising for the use of prompt learning for movie review sentiment analysis, as it demonstrates that prompt learning can be used to achieve state-of-the-art results with a simple prompt for a sentiment-related task. In addition to the strong results achieved, Y. Wang et al. 2022 used a similar approach for the same shared task and achieved the second strongest results for the sub-task, demonstrating that this approach is not trivial and can be used to achieve strong results for text classification tasks.

H. Zhang et al. 2022 proposes a prompt-based meta-learning model, hoping to mitigate the requirement for significant amounts of data, to accomplish few-shot learning tasks. They apply this model to several text classification datasets for this experiment, such as relation classification with the FewRel dataset (Han et al. 2018), news headline and article classification with the Huffington Post and Reuters datasets (Misra 2018, Lewis 1997), as well as reviews classification using Amazon reviews. They demonstrate excellent compatibility with meta-learning and prompt-tuning and achieve results comparable to the state-of-the-art for these tasks. This paper demonstrates that prompt learning can be used to achieve state-of-the-art results for text classification tasks, including on Amazon reviews, which is promising for the use of prompt learning for movie review sentiment analysis.

These papers have broadly demonstrated the ability of prompt learning models to handle text classification and sentiment analysis, including on similar datasets that this thesis will focus on. This is promising for the use of prompt learning for movie review sentiment analysis, as it demonstrates that prompt learning can be used to achieve state-of-the-art results for these tasks.

3.3 Explainable AI

A significant survey of XAI for NLP was performed in Danilevsky et al. 2020. The authors explain that the field has generally transitioned from *white box* techniques

to *black box* techniques. A white box technique is something that is inherently explainable, such as rule-based approaches or decision trees, which can be understood by humans (Danilevsky et al. 2020). A black box technique is something that is difficult or even impossible for a human to visualize and understand, such as neural networks or other DL strategies (Danilevsky et al. 2020). Black box techniques generally sacrifice the benefits of explainability in favour of much stronger performance. This is concerning, since it can lead to eroded trust in AI systems since it can be impossible to understand how they reach their results. XAI systems seek to provide the benefits of an understandable model to these black box techniques.

Danilevsky et al. 2020 uses two categories to separate explanation techniques. The first is a local or global category, which is whether an explanation is made for an individual prediction (local) or for the model's entire prediction process (global). The second category categorizes them based on whether the prediction process itself creates the explanation (self-explaining) or if a post-processing step is applied (post-hoc). For this thesis, we will focus on local post-hoc explainability techniques, as we prefer to have a system that can explain individual predictions on PLMs. In particular, we will attempt to apply LIME (Ribeiro, Singh, and Guestrin 2016) and SHAP (Lundberg and S.-I. Lee 2017), which are discussed further in the following sub-sections, along with research that uses these techniques.

An evaluation of various XAI algorithms is performed in Hase and Bansal 2020. This paper evaluates the performance of local post-hoc XAI techniques, including LIME. The paper uses two datasets, the first being a sentiment analysis dataset on movie review excerpts and the second being a dataset of individuals' background with labels of their annual income. The paper evaluates whether human subjects could predict a model's behaviour on new inputs. From their experiments, they found that only LIME was able to achieve a statistically significant result. This paper proposes various ways that this thesis could evaluate the performance of the XAI techniques that are implemented and also indicates that LIME may achieve the best results.

Clinciu and Hastie 2019 discusses the recent growth of interest in the field of explainable AI (XAI) but expresses a concern that the field is still in its infancy and that there are a variety of terms being used interchangeably, leading to confusion. The objective of the paper is to establish a set of common terms that can be used by the XAI community. In particular, there is concern about the difference between the terms *transparency*, *interpretability*, *intelligibility*, and *explainability*. They separate the terms as follows:

- **Transparency** is a measure of how well a person can understand how the

system works. The paper argues that it is a broad term that encompasses the other terms discussed in the work.

- **Intelligibility** is the ability of a system to make itself understood through communication to the user. It is a sub-field of transparency.
- **Interpretability** is the ability to determine what the intended meaning of something is. This term overlaps with explainability, but also contains its own distinct approaches. For example, a ranked numbering system might be inherently interpretable, even though it does not provide an explanation by itself.
- **Explainability** is the ability to determine the intended meaning through given information. In most cases, "it means providing a way to improve the understanding of the user" (Clinciu and Hastie 2019), such as through written explanations.

Clinciu and Hastie 2019 indicates a clear concern that should be considered in this study, as these distinctions are important to consider when introducing XAI methods to an AI system.

While XAI has seen a growth in research lately, there is still limited research into XAI for prompt learning models, which is a gap that we seek to address with Research Question 4. The following sub-sections will discuss some of the research that has been performed in this area using several XAI methods, along with related work for these methods.

3.3.1 Standard XAI Systems

This section discusses some of the XAI systems that have been used for NLP tasks. These systems are not specific to prompt learning models, but they are still relevant to this thesis since they can be used to explain the predictions of prompt learning models.

LIME-based XAI

Proposed in Ribeiro, Singh, and Guestrin 2016, LIME is a model for generating faithful explanations for individual predictions by a complex ML model. It works by

approximating the complex model in the local region around the instance being explained with a simpler and more interpretable model. To do this, LIME perturbs the input to be explained and generates a set of artificial examples. These perturbations are intended to be small while still being significant enough to change the output of the model. By sampling these perturbations and the resulting predictions, LIME can determine the importance of each input. These importance measures are useful for various reasons, such as explaining why the model made the prediction or for identifying biases and errors. LIME is particularly powerful since it is model-agnostic, which means it can be used with any ML model, including popular NLP models. For these reasons, Danilevsky et al. 2020 describes it as a popular baseline for XAI with NLP, which suggests it will be useful for explaining predictions made by the movie review sentiment analysis system and understanding the impact of prompt learning on the model's predictions.

While prompt learning remains a relatively novel field, there is some research that uses LIME to explain the predictions of prompt learning models. Jimenez Gutierrez et al. 2022 compared the few-shot performance of GPT-3 with the fine-tuning of smaller language models like BERT for text classification tasks. While their analysis did not match their expectation, with fine-tuning still providing superior results, they performed error analysis with LIME to analyze the predictions of both methods, finding that GPT-3 is more susceptible to surface-level signals and therefore more likely to make mistakes. This is an example of how LIME can be used to analyze the predictions of prompt learning models, which is promising for the use of LIME to analyze the movie review sentiment analysis system.

S. Wang et al. 2022 also analyzed the results of their prompt-based assertion classification using LIME. They used prompt learning for a particularly challenging task where they classify whether a diagnosis or condition is present, absent, or possible. This is challenging due to significant class imbalance where possible assertions very rarely appear and are often expressed vaguely. These challenges impose limitations on the ability of models, including modern DL methods, to accurately classify the texts. They found that prompt learning achieved a modest 2% improvement over previous works in the field, with particular success at classifying classes with few instances. They also used LIME to analyze the predictions of the model, finding that the model was able to identify the correct words in the text that led to the prediction by matching to human analysis. This is important for a medical application, as it can help doctors understand the model's predictions and identify potential errors. This is another example of how LIME can be used to analyze the predictions of prompt learning models, which is promising for the use of LIME to analyze the movie review

sentiment analysis system.

While research into XAI for prompt learning remains limited, these papers demonstrate that LIME is a useful tool for analyzing the predictions of prompt learning models. This suggests that LIME will be useful for analyzing the movie review sentiment analysis system and understanding the impact of prompt learning on the model's predictions.

SHAP-based XAI

SHAP is proposed in Lundberg and S.-I. Lee 2017 and is a similarly model-agnostic method for explaining predictions of complex ML models. SHAP is based on Shapley Values (Shapley et al. 1953), which are a concept in cooperative game theory that assigns values to each player in a game based on their contribution to the overall outcome of the game. In this case, SHAP calculates the contribution of each input feature to the overall prediction by comparing the output for all possible subsets of the input features. Using a weighted linear regression model, SHAP combines these contributions and generates an importance ranking of the features, the weights in the regression model being the Shapley Values of the input features. Additionally, while this thesis is focusing on local explanations, SHAP has an advantage over other XAI methods as it can also provide global explanations that provide a more general understanding of the model's behaviour across a dataset. SHAP provides an additional XAI model that can be used to explain the predictions of the movie review sentiment analysis system and to understand the impact of prompt learning on the model's predictions.

C. Yang et al. 2023 describes the usage of Shapley Values for XAI as widespread, including for neural text classification models. However, they argue that it is prohibitive to use them for LLMs due to a large number of model evaluations required to compute them. They propose a model that "directly predicts each input feature's Shapley Values" that is trained on a set of examples whose Shapley Values are estimated from a large number of model evaluations. They achieved a 60x improvement in execution speed compared to traditional approaches for computing Shapley Values.

Mosca et al. 2022 provides a general review of SHAP-based explanation models for NLP interpretability. They describe SHAP as "a core contribution to the field of eXplainable Artificial Intelligence" and state that after its development, "a variety of explainability approaches based on SHAP's methodology has populated the literature" and that the trend continues to grow (Mosca et al. 2022). This paper provides

an extensive list of SHAP-based explanation models, some of which are explored in this section.

Crucially, Kokalj et al. 2021 demonstrates the use of SHAP on transformer-based classifiers. This thesis will be using a transformer-based classifier for movie review sentiment analysis, so this paper is directly relevant to the thesis. They propose a new method called TransSHAP "that adapts SHAP to transformer models including BERT-based text classifiers" that "advanced SHAP visualizations by showing explanations in a sequential matter, assessed by human evaluators" Kokalj et al. 2021. They demonstrate this work on a Twitter sentiment analysis dataset that contains positive, negative, and neutral labels, visualizing which words in a tweet have a significant impact on the model's prediction. This is a similar task to the movie review sentiment analysis task, as they are both sentiment analysis tasks on social media posts. This suggests that SHAP will be useful for analyzing the movie review sentiment analysis system and understanding the impact of prompt learning on the model's predictions.

Pluciński and Klimczak 2021 demonstrates the use of SHAP to detect toxic spans in texts as part of SemEval-2021 Task 5. Their approach was to examine if XAI methods explaining high-performing models can lead to a similar prediction quality as dedicated models. While this approach did not outperform a typical BERT classifier, it achieves a competitive result in the shared task and demonstrates that "explainable methods can be sufficient for many tasks where binary decision models are used" (Pluciński and Klimczak 2021).

Self-Explaining Approaches

A self-explaining approach means the model is able to provide its own adequate explanation for its predictions.

For natural language applications, using attention scores for feature attribution (Xu et al. 2016) has been the primary method for developing self-explaining neural classifiers. Rajagopal et al. 2021 indicates that these approaches can provide local explanations using the relevance of input features, but that such interpretations are unreliable. Their work attempts to provide both global and local explainability using two layers that augment existing neural classifiers, creating the first self-explaining approach to provide both in a single model. This model is entitled SELFEXPLAIN and they use human evaluation that demonstrated a 22% improvement in ability to predict a model's decision using its explanations.

Another example of self-explaining XAI is Z. Wang et al. 2019, where they use a tree-structured LSTM to learn a representation of each unit with parameter sharing that is context-independent. They achieve strong results on their task, with a best result of about 65%.

3.3.2 Evaluation of XAI Systems

Boyd-Graber et al. 2022 states that while there are a variety of algorithms and models that can explain model predictions, there is much less consensus on how these explanations can be evaluated, especially since these explanations should be understandable by humans. They propose that "it is important to take a human-centered approach to their evaluation, meaning evaluating with respect to human criteria" (Boyd-Graber et al. 2022), including measuring views of the explanations and whether they fill the needs of people. Despite identifying this concern, they struggle to identify a suitable and specific method for evaluating XAI systems. This is a concern that should be considered in this work, as it is important to evaluate the XAI system in a way that is meaningful to humans.

The challenge of evaluating explanations quantitatively is described as "notoriously difficult" in Rajagopal et al. 2021 during the evaluation of their self-explaining XAI system. They use human evaluators to evaluate the understandability of explanations (how well a user can understand the explanation produced by the model), the trustworthiness of models (if a user trusts the model's decision, based on its explanation), and the sufficiency of the justification (if the factors cited in an explanation sufficiently justify the decision). Additionally, they use an approach where a BERT model is trained exclusively with the explanations to perform the task. If explanations alone can be used to perform the task, then the explanations are sufficient.

As part of their survey on the state of XAI, Danilevsky et al. 2020 provide a section that discusses the state of the field in terms of defining and measuring explanation quality. They are particularly concerned about this, since a "majority of the works reviewed either lack a standardized evaluation or include only an informal evaluation" (Danilevsky et al. 2020). They also identify that a small number of papers looked at formal evaluation approaches, such as ground truth data and human evaluation. Danilevsky et al. 2020 identifies several approaches that are primarily used for evaluation of XAI systems in literature:

- **Informal examination of explanations:** This approach involves high-level

discussions of how the generated explanations align with human intuition. This discussion sometimes involves only a small number of examples or a comparison to other baseline approaches, such as models like LIME. This approach struggles to provide a formal evaluation of the explanations, but it can provide a high-level understanding of the explanations.

- **Comparison to ground truth:** Using ground truth data, researchers can apply standards methods like precision, recall, and F1-score to evaluate the performance directly, or metrics like Bilingual Evaluation Understudy (BLEU) (Papineni et al. 2002) that calculate the similarity of an explanation to the ground truth. One disadvantage is that ground truth data sometimes falsely assumes that there is only one good explanation. It also requires a large amount of ground truth data, which can be difficult to obtain and can be influenced by subjective human opinions.

- **Human evaluation:** This approach assesses the quality of explanations by asking human evaluators to rate the effectiveness of generated explanation. This approach is useful since it addresses the desire of Boyd-Graber et al. 2022 to be human-centric, but human evaluations can be expensive and time-consuming.

Attanasio et al. 2023 presents a Python library to simplify the use and comparison of XAI methods on transformer-based models. The library allows users to visualize and compare the explanations of transformer-based models using a variety of XAI methods, including LIME and SHAP. The library implements explanations about the faithfulness and plausibility of the explanation. Faithfulness is describes as "how accurately the explanation reflects the inner working of the model" (Attanasio et al. 2023) and is measured using comprehensiveness, sufficiency, and correlation with a "leave-one-out" score, which we describe further below. Attanasio et al. 2023 defines plausibility as how the explanations align with human reasoning by comparing the explanations with human rationales using Intersection over Union (IoU), F1-scores, and Area Under Precision-Recall Curve (AUPRC), as described below. The following are the specific metrics used for both categories:

- **Comprehensiveness (Faithfulness):** This metric evaluates whether the explanation correctly captures the tokens used by the model to make its prediction. It is measured by removing the highlighted tokens and observing the change in probability of a given prediction.

- **Sufficiency (Faithfulness):** This metric evaluates if the tokens used in the model are sufficient for the model to make its prediction. A low sufficiency score indicates that the model relies on other tokens to make its prediction.
- **Correlation (Faithfulness):** This metric calculates the leave-one-out scores for all tokens, where a token is omitted and the difference in the model prediction is measured. For every token, this provides an importance score. The correlation with these scores and the explanation indicates strong faithfulness.
- **IOU (Plausibility):** This metric calculates the intersection over union between the explanation and the human rationale (when it is available). A high IOU indicates that the explanation is similar to the human rationale.
- **F1-score (Plausibility):** This metric calculates the F1-score between the explanation and the human rationale. A high F1-score indicates that the explanation is similar to the human rationale.
- **AUPRC (Plausibility):** This metric calculates the area under the precision-recall curve between the explanation and the human rationale. A high AUPRC indicates that the explanation is similar to the human rationale.

Using this library, one could evaluate the explanations generated by the XAI component of the movie review sentiment analysis system with these metrics. This would allow effective evaluation of the faithfulness and plausibility of the explanations, which would provide a strong evaluation of the explanations generated by the system.

An appropriate evaluation method is essential for this thesis, as it is important to evaluate the XAI system in a way that is meaningful to humans. This section has provided a brief overview of the state-of-the-art research into XAI evaluation, which can be used to guide the evaluation of the proposed system.

3.4 Summary

In this chapter, we explored much of the existing research into the relevant topics of this thesis. We began with an introduction to existing approaches for sentiment analysis of movie reviews, exploring many state-of-the-art methods that typically lead to very strong results. In particular, we explored the strengths of these papers and how they can be used to guide this thesis.

We then explored the topic of prompt learning, which is a relatively new topic in NLP that has seen a surge in research lately. We discussed the state-of-the-art research into prompt learning, which demonstrates that prompt learning can be used to achieve state-of-the-art results for text classification tasks, including some research on its use on movie reviews. This is promising for the use of prompt learning for movie review sentiment analysis, as it demonstrates that prompt learning can be used to achieve state-of-the-art results for these tasks.

Lastly, we discussed the topic of XAI, which is another topic experiencing a surge in research in recent years. We explored the state-of-the-art research into XAI, which demonstrates that XAI can be used to explain the predictions of PLMs. This is promising for the use of XAI for movie review sentiment analysis, as it demonstrates that XAI can be used to explain the predictions of PLMs.

This literature review allows us to identify some key gaps in existing work. For example, there is little to no research into explainability for prompt learning models. This thesis attempts to address this gap by proposing a novel system that integrates prompt learning and XAI to explain the predictions of a movie review sentiment analysis system. This thesis also attempts to address the gap in evaluation of XAI systems by using a variety of evaluation metrics, including some that are proposed in this literature review.

Chapter 4

Methodology

This chapter presents a detailed description of the methods and procedures used in this research. This chapter begins with an overview of the general architecture of the proposed system, followed by a detailed description of the experimental setup used to train and run the system. This chapter then describes the IMDB dataset used for this research, followed by an explanation of the text preprocessing methods used for the dataset. This chapter then describes the detailed architecture of the proposed system, followed by a description of the evaluation metrics used to evaluate the system, before concluding with a summary.

4.1 General Architecture

The proposed system is a movie review sentiment analysis system that uses prompting to classify the sentiment of movie reviews and then uses XAI to explain these predictions. We can divide the system into three main components, consisting of a Data Processing Component, a Sentiment Classifier Component, and an Explainability Component. These components and the associated processes can be observed in Figure 4.1. The following sections will describe each of these components in detail.

4.1.1 Data Processing

The data processing component is the first component used in the proposed system and serves a few key functions:

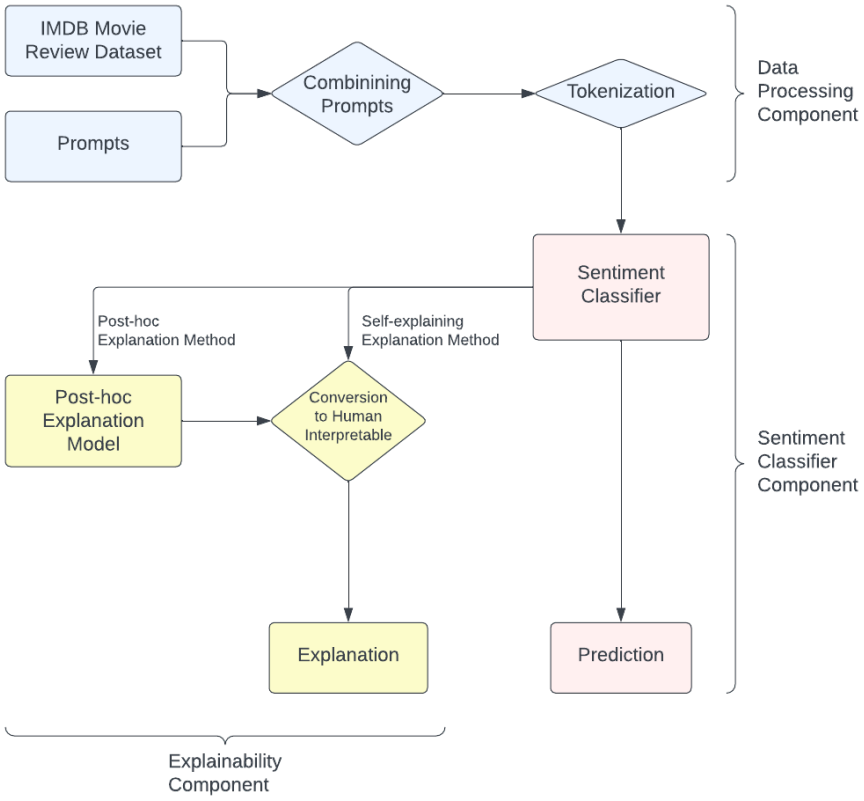

Figure 4.1: Visualization of the architecture of the final system.

1. **Inputs:** The data processing component takes the IMDB dataset as input. It then combines the reviews with proposed prompts that help guide the model towards a prediction. The dataset is described in more detail in Section 4.3.

2. **Preprocessing:** The data processing component includes the preprocessing of the data, which includes any tokenization, truncation, and padding necessary for this process. The preprocessing steps are described in more detail in Section 4.4.

3. **Outputs:** The result of the data processing component is the preprocessed text which is used as input for the sentiment classifier component.

It is necessary to determine exactly what preprocessing steps are necessary. For example, previous systems have performed additional steps like removing punctu-

ation and stop words. We will determine exactly what preprocessing steps are required, either following recommendations from existing works or through experimentation.

4.1.2 Sentiment Classifier

The sentiment classifier is the core component of the proposed system. Its key function is to take the preprocessed inputs from the data processing component and classify the sentiment of the movie review. The component will produce this classification as an output, which is also used to evaluate the performance of the component.

A PLM will be used for this component, as they have been shown to achieve state-of-the-art results for movie review sentiment analysis tasks. We propose some key experiments to evaluate the PLM and the prompt learning approach:

1. **Without Fine-tuning:** We will evaluate the performance of the PLM without fine-tuning. This will allow us to determine the performance of the PLM without any fine-tuning, which will allow us to determine the effectiveness of the prompt learning approach.
2. **With Fine-tuning:** We will evaluate the performance of the PLM with fine-tuning using the proposed prompts (referred to as prompt learning). We can compare the results of this experiment with the previous experiment to demonstrate the effectiveness of the prompt learning approach and ensure that this approach is useful.

We will evaluate these experiments using metrics proposed later in this chapter. The best-performing model from the latter experiments will be used as part of the proposed system. We will also compare the results of the latter experiment with the state-of-the-art results for the IMDB dataset, which will allow us to determine if the proposed system achieves state-of-the-art results.

4.1.3 Explainability

The final component of the proposed system is the explainability component. Its key function is to determine why the PLM described above made the prediction it did and

to convey that reasoning to the user in a written explanation. The component will produce this explanation as an output, which is also used to evaluate the performance of the component.

We will select several XAI methods to evaluate for this component. We will evaluate these methods using metrics proposed later in this chapter. The best-performing method will be used as part of the proposed system. Both the post-hoc and self-explaining methods described in Chapter 2 will be evaluated for use in this component.

The explainability component is the most novel component of the proposed system. There is little to no research into explainability for prompt learning models, so we will need to determine the best approach for this component. We will also need to determine the best approach for evaluating the performance of this component, which is described in Section 4.6.

4.2 Experimental Setup

For this research, all development is performed using Python 3.10.6 (Van Rossum and Drake 2009). A full list of Python packages (including versions) used for this research can be found in Appendix B, but the most important packages are listed below.

- **pandas (McKinney 2010):** A package for data manipulation and analysis that provides data structures and methods to handle structured data. This work is under the BSD 3-Clause license.
- **NumPy (Harris et al. 2020):** A package for scientific numerical computing that provides efficient multidimensional array operations, mathematical functions, and tools for working with arrays. This work is under the BSD 3-Clause license.
- **transformers (Wolf et al. 2020):** A package that provides a high-level interface for ML models, including PLMs. This work is under the Apache License 2.0.
- **openai:** A toolkit that allows developers to interact with and use OpenAI's models, such as GPT-3.5. OpenAI provides terms of use for their API at `https://openai.com/policies/terms-of-use`. All usage in this research is in accordance with these terms.

Unless otherwise stated, the experiments conducted in this research are performed on a computer with an Intel Core i5-12600K CPU and an NVIDIA GeForce GTX 1060 GPU. The GPU's CUDA driver version is 11.7 and cuda-python version 12.1.0 is used. The machine has 32 GB of RAM and the OS used is Windows 11 Pro.

The only exception to this setup is for the use and fine-tuning of Llama 2, where we use a T4 GPU with increased memory in a Google Colab environment, as the model requires more memory than we have available on our local machine. The T4 GPU has 16 GB of memory, which is sufficient for fine-tuning Llama 2.

4.3 Dataset Description

The dataset used for this thesis is the IMDB dataset proposed by Maas et al. 2011 for binary text classification of sentiment analysis. It is a popular dataset that is intended for sentiment analysis research and is extensively used because it provides a significant amount of data, especially compared to previous sentiment analysis datasets. This makes it particularly well suited for training and evaluating DL models. This dataset has results reported from a wide variety of systems, as discussed in the literature review. Selecting this dataset ensures we can compare against a wide variety of existing baselines. This section will briefly describe the collection and structure of the dataset.

Maas et al. 2011 constructs the collection with 50,000 reviews from IMDB. The dataset is built such that there is a maximum of 30 reviews per movie due to the possible existence of correlations between the reviews and the associated movies, mitigating such an impact on the dataset. Additionally, they only consider highly polarized reviews by only allowing reviews with either a rating of ≤ 4 (negative reviews) or ≥ 7 (positive reviews), both out of a maximum score of 10. This ensures that there is a clear and strong sentiment associated with every review. Importantly, the constructed dataset contains an even number of positive and negative reviews, so a completely random guesser should achieve an accuracy of about 50%. This is important, as it ensures that the dataset is not biased towards either class.

The dataset is an English-language dataset that has been fully anonymized to ensure no personal information is included in the dataset. Only the review texts and their appropriate labels are included. Maas et al. 2011 does not specify a license for the dataset, simply requesting that their work be cited where it is used.

The dataset is provided in a CSV format, with each row of the file containing a review and the corresponding label. The review is a string containing the text of

the review, while the label is either "positive" or "negative". An example of reviews directly taken from the dataset is shown in Table 4.1.

Already, we can observe some challenges with this data in Table 4.1, such as the HTML tags remaining present. Some of these challenges will be explored in the following section while describing the text preprocessing steps. Other challenges cannot be resolved through a preprocessing step, such as user spelling errors or the presence of sarcasm.

4.3.1 Data Splitting

The dataset is provided in a single CSV file, which is split into a training set and a test set. The training set contains 25,000 reviews, while the test set contains 25,000 reviews. The training set is used to train the model, while the test set is used to evaluate the model. This is a common approach for ML tasks, as it allows the model to be evaluated on data that it has not seen before. This is important, as it allows us to determine if the model is able to generalize to unseen data.

We additionally create a subset of the data that will be used for the explainability component of this thesis. One limitation of XAI techniques is that they frequently require a large number of model evaluations. This is a limitation discussed in C. Yang et al. 2023 for Shapley values, where they describe an approach for efficiently estimating Shapley values in texts. This is a limitation that we will need to consider for this thesis, as we will need to evaluate the model many times to generate explanations. We will use a subset of the test set that contains 50 reviews, which will be used for the explainability component of this thesis. This subset will be randomly selected from the test set while ensuring that it contains an equal number of positive and negative reviews.

4.3.2 Related Datasets

There are numerous datasets that were used for sentiment analysis tasks. This work will focus on the IMDB dataset, with the potential for other datasets to be used in future work. Engaging with additional datasets requires more data preprocessing and model tuning steps, which is challenging to do in the scope of this thesis. We focus on the IMDB dataset to allow for more in-depth analysis and comparison to many existing works that use that dataset. Nonetheless, this section will briefly describe some of the related datasets that can be used for sentiment analysis tasks.

Table 4.1: Example of data from the IMDB dataset.

Review	**Label**
Somewhat too long and going over the top towards the end, this comedy is an utterly delightful, never condescending or ridiculing look into the problems of a "power man", who likes to wear womenÂ´s clothes at nite. Julie Walters is lovely as always, but Adrian Pasdar is utterly credible and steals the film. He (she)is absolutely gorgeous in high heels and silk stockings.	positive
People say that this film is a 'typical teen horror movie'... well it's a horror movie with a teenage girl in it.. what do you expect! It's a good film, I counted 3 actual screams in the audience whilst the film was on and it was a very jumpy scary film. I wasn't bored in the film at any point and I was even on the edge of my seat at one point. The only thing that was slightly bad was that it was a tiny bit slow in getting into the actual storyline but this all led up to why she was where she was and why what happened, happened. The acting was good, the scenery was good and the storyline was good too, I hope to see a 'When A Stranger Calls 2' in a few months! Good film!	Positive
Here is what happened: 1) Head of BBC3 needs to make programmes aimed at different audience to BBC1 and BBC2 to keep licence and job. 2) Lenny Henry offers his unfunny friends up. 3) Head of BBC3 snaps them up, completely ignoring the fact that they are not funny. Worst of all, it is arguably racist, as all the characters play up to bad stereotypes. If a white person did this kind of thing, there'd be uproar! Trash.	negative
I wouldn't call myself a big fan of the genre inventive silliness, so i might not be the best audience for this show. Although, being a critic i do have a sense for what i personally like and dislike, this being the later. Lack of humor is a big turnoff when it comes to comedy, things can be catchy, cool and perky for about 4 minutes and after that you start getting bored unless its the badger animation from a couple of years ago (?) This is the exact opposite, with a stiff script and all overacted voice-overs its just plain silly and very very boring to be subjected to. Unfortunately, since it did have a big market ahead of itself, and a lot of potential. A waste of time.	negative

Another reason to choose this dataset is that it is used in many papers that work on explaining text classifiers Lucaci and Inkpen 2021. The task of sentiment classification on this dataset is rather simple, allowing us and other researchers to focus on explaining the classifiers or their decisions. Other works that have used this dataset as a baseline include DistilBERT (Sanh et al. 2019), Longformer (Beltagy, Peters, and Cohan 2020), and PoWER-BERT (Goyal et al. 2020), as well as fine-tuned versions of BERT (Sun et al. 2019) and RoBERTa (Tan et al. 2022) that use this dataset.

Amazon Product Reviews

The Amazon Product Reviews dataset (He and McAuley 2016) contains 82.83 million unique reviews, consisting of the review text and ratings, along with various additional metadata. Ratings are given from 1 to 5, while the IMDB dataset uses binary positive or negative labels. This is a massive and diverse dataset, containing a substantial amount of text data, allowing a wide variety of sentiments and opinions to be captured. Additionally, due to the diversity of products on Amazon, this dataset is well suited for making generalized models.

Yelp Polarity Reviews

The Yelp Polarity Reviews dataset (X. Zhang, Zhao, and LeCun 2016) contains 1.5 million samples that have review texts and ratings. The ratings are given from 1 to 5, while the IMDB dataset uses binary positive or negative labels. This dataset is also well suited for making generalized models, as it contains a wide variety of reviews for different businesses.

Rotten Tomatoes Reviews

Similarly to the IMDB dataset, the Rotten Tomatoes dataset (Pang and L. Lee 2005) contains 5331 positive and 5331 negative sentences from movie reviews on Rotten Tomatoes. This dataset is also popular for movie review sentiment analysis tasks, although it is much smaller than the IMDB and therefore may not be as well suited for training DL models and may also be less capable of making generalized models.

Table 4.2: Example of challenging data in the IMDB dataset.

Review	**Challenge**
(Elijah Wood is the victim in both films) and wait.....it hypnotizes (stings) its victim and wraps them up.....uh hello???? And the whole machine vs. humans theme WAS the Matrix..or Terminator..... 	HTML tags, various ellipses
So: IF you want to see the WORST movie ever... go ahead, I recommend it :)	Sarcasm (recommending it), emoticons
The film is neither as good as as bad as some people say here.	Negation (neither), typo (as as)

4.4 Text Preprocessing

The reviews contained in the IMDB dataset are typically very informally written, similar to how people write on social media. These texts can pose many challenges for NLP models due to sarcasm, a lack of proper punctuation, spelling errors, abbreviations, various localisms, and slang. Additionally, the data may contain emojis that are intended to express emotions or special characters that are exclusive to other languages. Some of these challenges are described in Table 4.2. One significant challenge is the presence of HTML tags in the data, which is already evident in Table 4.1. These tags are intended to format the text for the website, but are not typically useful for NLP models. This section will describe the preprocessing steps that are applied to the IMDB dataset to mitigate these challenges.

Some of the typical steps for preprocessing text data are described below, along with the justification for whether to use them or not.

- **Punctuation Removal:** Punctuation is generally not removed from the data because LLMs are designed to learn from the context of the text. Punctuation can be useful for determining the meaning of a sentence (such as the famous "Let's eat grandma" and "Let's eat, grandma" example), so it is **not removed** from the data.

- **Stopword Removal:** Stopwords are words that are insignificant to the overall meaning of a text, such as "the", "a", "an", "and", etc. These words are typically removed from the data because they do not provide much value to

the model. However, LLMs are designed to learn with these words, so they are **not removed** from the data.

- **HTML Tags:** The IMDB dataset contains HTML tags that are used to format the text for the website. These tags are not typically useful for NLP models, so they are **removed** from the data. This is done using the `BeautifulSoup` library (Richardson 2007), which includes an HTML parser to remove the tags.
- **Special characters:** Special characters consist of non alpha-numerical characters. Some, like emojis, are typically used to express emotions, which can be useful for sentiment analysis. However, they are not typically useful for LLMs, so they are **removed** from the data. This is done using a regular expression to match any non alpha-numerical characters.

These steps will be applied to the IMDB dataset to mitigate some of the challenges described in Table 4.2. The resulting dataset will be used to train the LLMs in the following chapters.

Text preprocessing also includes the step of tokenization. Tokenization is the process of breaking down a text into individual words or units, known as tokens. Depending on the model, these tokens can be words, sub-words, characters, or any other meaningful unit of text, such as symbols. Tokenization is a fundamental step of NLP tasks and is crucial for allowing machines to understand and analyze human language.

Different transformers have different tokenizers to accommodate various language characteristics, tokenization strategies, and model architectures. For example, the "BertTokenizer" from the HuggingFace Transformers package (Wolf et al. 2020) for BERT or "RobertaTokenizer" for RoBERTa. Table 4.3 displays the tokenized versions of the same text using various tokenizers, demonstrating the distinctions between these models. For BERT, RoBERTa, DistilBERT, ELECTRA, and Llama 2 we will use the appropriate tokenizer for each model by using the "AutoTokenizer" class from the HuggingFace Transformers package, which will automatically select the appropriate tokenizer for the model. For GPT-3.5, text is tokenized through the OpenAI API and is therefore not handled in our code. Instead, the GPT-2 tokenization is demonstrated in Table 4.3, which is likely similar to the tokenization process for GPT-3.5.

Table 4.3: Examples of a review tokenized for several large language models.

Model	**Review**
None (original)	This film tried to be too many things all at once stinging political satire Hollywood blockbuster sappy romantic comedy family values promo the list goes on and on It failed miserably at all of them but there was enough interest to keep me from turning it off until the end
bert-large-uncased	['this', 'film', 'tried', 'to', 'be', 'too', 'many', 'things', 'all', 'at', 'once', 'stinging', 'political', 'satire', 'hollywood', 'blockbuster', 'sap', '##py', 'romantic', 'comedy', 'family', 'values', 'promo', 'the', 'list', 'goes', 'on', 'and', 'on', 'it', 'failed', 'mis', '##era', '##bly', 'at', 'all', 'of', 'them', 'but', 'there', 'was', 'enough', 'interest', 'to', 'keep', 'me', 'from', 'turning', 'it', 'off', 'until', 'the', 'end']
xlm-roberta-large	['This', 'film', 'tried', 'to', 'be', 'too', 'many', 'things', 'all', 'at', 'once', 'sting', 'ing', 'political', 'sati', 're', 'Hollywood', 'block', 'bu', 'ster', 'sa', 'ppy', 'romantic', 'comedy', 'family', 'values', 'promo', 'the', 'list', 'goes', 'on', 'and', 'on', 'It', 'failed', 'miser', 'ably', 'at', 'all', 'of', 'them', 'but', 'there', 'was', 'enough', 'interest', 'to', 'keep', 'me', 'from', 'turning', 'it', 'off', 'until', 'the', 'end']
distilbert-base-uncased	['this', 'film', 'tried', 'to', 'be', 'too', 'many', 'things', 'all', 'at', 'once', 'stinging', 'political', 'satire', 'hollywood', 'blockbuster', 'sap', '##py', 'romantic', 'comedy', 'family', 'values', 'promo', 'the', 'list', 'goes', 'on', 'and', 'on', 'it', 'failed', 'mis', '##era', '##bly', 'at', 'all', 'of', 'them', 'but', 'there', 'was', 'enough', 'interest', 'to', 'keep', 'me', 'from', 'turning', 'it', 'off', 'until', 'the', 'end']
google/electra-large-generator	['this', 'film', 'tried', 'to', 'be', 'too', 'many', 'things', 'all', 'at', 'once', 'stinging', 'political', 'satire', 'hollywood', 'blockbuster', 'sap', '##py', 'romantic', 'comedy', 'family', 'values', 'promo', 'the', 'list', 'goes', 'on', 'and', 'on', 'it', 'failed', 'mis', '##era', '##bly', 'at', 'all', 'of', 'them', 'but', 'there', 'was', 'enough', 'interest', 'to', 'keep', 'me', 'from', 'turning', 'it', 'off', 'until', 'the', 'end']
gpt2	['This', 'Ġfilm', 'Ġtried', 'Ġto', 'Ġbe', 'Ġtoo', 'Ġmany', 'Ġthings', 'Ġall', 'Ġat', 'Ġonce', 'Ġst', 'inging', 'Ġpolitical', 'Ġsatire', 'ĠHollywood', 'Ġblockbuster', 'Ġsa', 'ppy', 'Ġromantic', 'Ġcomedy', 'Ġfamily', 'Ġvalues', 'Ġpromo', 'Ġthe', 'Ġlist', 'Ġgoes', 'Ġon', 'Ġand', 'Ġon', 'ĠIt', 'Ġfailed', 'Ġmiser', 'ably', 'Ġat', 'Ġall', 'Ġof', 'Ġthem', 'Ġbut', 'Ġthere', 'Ġwas', 'Ġenough', 'Ġinterest', 'Ġto', 'Ġkeep', 'Ġme', 'Ġfrom', 'Ġturning', 'Ġit', 'Ġoff', 'Ġuntil', 'Ġthe', 'Ġend']

4.5 Sentiment Classifier

As described earlier in the section, the sentiment classifier is the core component of the proposed system. We propose two sets of experiments to evaluate the performance of PLMs before and after applying prompt learning. The first set of experiments evaluates these models without any additional fine-tuning, while the second set of experiments fine-tunes them using the prompts and evaluates their performance. The following section provides more insight into the methodology of these two experiments.

4.5.1 Selected Models

The first step in creating the sentiment classifier is to select a set of PLMs to evaluate for this task. These models will be used for both of the proposed experiments. We propose the following criteria to select models for this task:

- **Baseline Results:** We will consider models that have been used on the IMDB dataset in previous research. This will allow us to compare the performance of the proposed system to previous research.
- **Model Size:** Prompt learning requires LLMs since they have sufficient language knowledge to perform this task. Small or "base"-sized models are typically too small to perform prompt learning, although we can evaluate this impact by comparing the performance of small and large models.
- **Model Type:** We should consider a diverse set of language models. We will consider, at minimum, BERT-based and GPT-based models. This will allow us to evaluate the impact of prompt learning on multiple types of models.
- **Open Source:** We should focus on using models that are open source and publicly available. This will allow us to use these models in the proposed system.
- **State of the Art:** We should consider models that are state-of-the-art for the NLP. This will allow us to evaluate these innovative systems for this task.

Based on these criteria, we can select a set of models for evaluation. Table 4.4 depicts the model families selected for this evaluation, along with the justification for their selection according to these proposed criteria.

Table 4.4: Model families selected for evaluation and the justification for their selection.

Model	**Justification**
BERT	Strong baseline results to compare against. Both a "base" and "large" sized model are available to compare. Open source.
RoBERTa	Strong baseline results to compare against. Both a "base" and "large" sized model are available to compare. Open source. State-of-the-art model for this task.
DistilBERT	Strong baseline results. Open source.
ELECTRA	Some baseline results. Each of "small", "bases", and "large" sized models are available to compare. Non-BERT and non-GPT to introduce variety. Open source.
GPT	Few to no reported baseline results. The ada, babbage, curie, and davinci versions can be compared, providing various sizes to compare. Often considered "state-of-the-art" for NLP (Gupta et al. 2023).
Llama 2	Few to no reported baseline results. Offers sizes ranging from base (7 billion parameters) to large (70 billion) Often considered "state-of-the-art" for NLP (Touvron et al. 2023).

Table 4.5: Specific models chosen for evaluation for each model family.

Model Family	**Small**	**Base**	**Large**
BERT		bert-base-uncased	bert-large-uncased
RoBERTa		xlm-roberta-base	xlm-roberta-large
DistilBERT		distilbert-base-uncased	
ELECTRA	google/electra-small-generator	google/electra-base-generator	google/electra-large-generator
GPT	text-babbage-001 (gpt-3.5)	text-davinci-003 (gpt-3.5)	gpt-4
Llama 2		meta-llama/Llama-2-7b	

By selecting this set of models, we can compare the performance of a wide range of popular and powerful models, each with its own unique characteristics. BERT and RoBERTa are well known for their deep understanding of context and long-ranged dependencies, along with two differently sized models each that can contribute to answering Research Question (RQ)2. DistilBERT provides only one model, but this can also be compared to BERT and RoBERTa to evaluate the impact of size as part of RQ2. ELECTRA introduces a novel training approach and excels in various NLP tasks, including previous work in prompt learning seen in the literature review. One benefit of ELECTRA is that it provides three different sizes of models, which can be used to evaluate the impact of size as part of RQ2. GPT has seen an explosion of popularity recently but has limited reported baseline results for this set. It also offers different sizes or versions that can be evaluated. Finally, Llama 2 is selected as it is a state-of-the-art LLM released in 2023 and has achieved outstanding results on other tasks. This selection allows for a comprehensive evaluation of model sizes, training methodologies, and capabilities, providing valuable insights in a direct comparison like this.

The specific versions of each model chosen are displayed in Table 4.5.

Overall, this selection of models provides a wide range of models to evaluate, each with its own unique characteristics. This will allow us to evaluate the impact of prompt learning on a variety of models, which will allow us to answer RQ2 by

identifying the language models most suitable for prompt-based movie sentiment classification.

4.5.2 Prompt Engineering

As described in the background chapter, prompt engineering refers to the process of crafting effective prompts or instructions for generating desired outputs from a language model or AI system. It involves formulating clear and specific instructions to guide the model's response generation. The goal of prompt engineering is to elicit accurate and relevant information or responses from the model by providing it with the right context and constraints. This section will describe the process of prompt engineering, including the definition of prompts and the templates that will be used. The creation and evaluation of these prompts will allow us to address RQ1 and RQ2, showing how we can apply prompting techniques and the most suitable language models to do so.

Prompt Definition and Types

A prompt refers to a specific instruction or input given to a language model that can guide its generation of a desired output. These prompts are an essential component to control the behaviour of LLMs and can be carefully crafted to elicit more accurate and useful responses from the model.

Prompts can take a variety of forms. A few potential prompts are listed below:

- **Instructional Prompts:** Instructional prompts provide instructions to the model about what is expected in the response. For example, an instructional prompt for a language model might be: "Write a summary of the main points in the given article."
- **Completion Prompts:** Completion prompts involve providing a partial sentence or phrase that the model is expected to continue. For instance, a completion prompt could be: "In the future, technology will...". These can also take the form of fill-in-the-blank prompts, such as: "The capital of Canada is [MASK]", with the model expected to fill in the blank [MASK] space.
- **Question Prompts:** Question prompts involve asking a specific question to the model. This type of prompt can be used to gather information or seek an

opinion. An example of a question prompt would be: "What are the benefits of exercise?"

- **Dialogue Prompts:** Dialogue prompts simulate a conversation by providing both user and model inputs. These prompts are commonly used to create interactive conversational experiences. For example:
  - User: "What is the weather like today?"
  - AI: "The current temperature is 25 degrees Celsius with clear skies."
- **Contextual Prompts:** Contextual prompts provide relevant background information to guide the model's response. They help the model understand the context and generate more accurate outputs. For instance: "You are a tour guide in Paris. A tourist asks you for recommendations on local attractions."

These prompts are just a few examples of the types of prompts that developers can use in prompt engineering. They provide significant control over the behaviour of language models to ensure they generate a desired output.

For this thesis, we will focus on three types of prompts from the above list: instructional prompts, completion prompts, and question prompts. This variety of prompts is used to demonstrate different behaviour across models for different prompts. For example, a masked language model may perform better on completion prompts. This variety of prompts will allow us to compare which ones are most effective, giving a strong result for RQ1. The specific prompts that will be used are described in the following subsections.

Instructional Prompts

As mentioned, instructional prompts provide specific instructions to the model about what is expected in the response. Essentially, they tell the model what to do. For our task, we want to provide an instruction to tell the model to determine the sentiment of a given review. We propose the following prompt templates for this task:

1. Determine the sentiment of the following review: [REVIEW]. The sentiment is [SENTIMENT].
   - This prompt template is the simplest of the three. It provides the model with a clear instruction to determine the sentiment of the given review.

However, the instruction is not specific, so the model may choose to generate a response other than "positive" or "negative", such as "happy" or "sad". This may be a problem, as we want the model to generate a response that is consistent with the labels in the dataset.

2. Indicate whether the following review is positive or negative: [REVIEW]. The sentiment is [SENTIMENT].

    - This prompt template is more specific than the previous one. It provides the model with a clear instruction to generate a response that is either "positive" or "negative". This should help the model generate a response that is consistent with the labels in the dataset.

3. Provide the sentiment of the following review as either positive or negative: [REVIEW]. The sentiment is [SENTIMENT].

    - This prompt template is very similar to the first template, with the difference that it specifies that the review should be classified as either "positive" or "negative". This should help the model generate a response that is consistent with the labels in the dataset. However, the wording of the prompt is more complex, which may make it more difficult for the model to understand.

These reviews provide a small set of instructional prompts that can be used to guide the model's generation of sentiment, varying in complexity and specificity. These prompts will be used to guide the generation of sentiment in the sentiment classifier, which will allow us to answer RQ1 and RQ2.

Completion Prompts

As described in the earlier section, completion prompts involve providing a partial sentence or phrase that the model is expected to continue. One approach for completion prompts is to use a "fill-in-the-blank" style prompt, where the model is expected to fill in the space. For our task, we want to provide a partial sentence that the model can complete to generate the sentiment of the given review. We propose the following prompt templates for this task:

1. [REVIEW]. The review expresses a [SENTIMENT] opinion of the movie.

- This prompt template provides the model with a partial sentence that it can complete to generate the sentiment of the given review. The wording is intended to limit the model's response to "positive" or "negative", which should help the model generate a response that is consistent with the labels in the dataset. There is a risk that other words will be selected, such as "happy" or "sad", which may be a problem.

2. [REVIEW]. The user has a [SENTIMENT] opinion of the movie.

    - This prompt template is similar to the first one, with the focus on the user's opinion of the movie rather than the review's. Similarly to the above, it is intended to restrict the response to "positive" or "negative", but there is a risk that other words will be selected.

3. [REVIEW]. The reviewer's impression of the movie is [SENTIMENT].

    - The third prompt template changes the structure of the sentence to provide a different context for the model. The sentence may not restrict the outputs to "positive" or "negative", but they should be in the top few choices.

4. [REVIEW]. The reviewer's overall assessment of the movie is [SENTIMENT].

    - The fourth prompt template is similar to the third one, with the difference that it uses the word "assessment" instead of "impression". This may change the context of the sentence, which may affect the model's response.

5. [REVIEW]. The sentiment expressed in the review is [SENTIMENT].

    - The fifth prompt template switched to the passive voice, which may change the context of the sentence and affect the model's response.

6. [REVIEW]. The reviewer's reaction to the movie is [SENTIMENT].

    - The last completion prompt template is similar to Prompts 3 and 4, changing "impression" or "overall assessment" to "reaction". This may change the context of the sentence, which may affect the model's response.

These prompts provide a long set of completion prompts that can be used to guide the model's generation of sentiment, varying in complexity and specificity, as well as similarity to each other. These prompts will be used to guide the generation of sentiment in the sentiment classifier, which will allow us to answer RQ1 and RQ2.

Question Prompts

The final proposed prompts are question prompts, which involve providing a question that the model is expected to answer. For our task, we want to provide a question that the model can answer to generate the sentiment of the given review. They are relatively similar to instructional prompts, but they are phrased as a question rather than an instruction. We propose the following prompt templates for this task:

1. What is the sentiment of the following review? [REVIEW]. The sentiment is [SENTIMENT].
    - This prompt template is based on the first instructional prompt, rephrased as a question. It provides the model with a clear question to answer about the sentiment of the review, although it retains the concern of the first instructional prompt that the sentiment could be something other than "positive" or "negative".
2. Is the following review positive or negative? [REVIEW]. The sentiment is [SENTIMENT].
    - Similarly to the second instructional prompt, this prompt now provides the options of either "positive" or "negative" to the model. This should help the model generate a response that is consistent with the labels in the dataset.
3. Is the sentiment of the following review positive or negative? [REVIEW]. The sentiment is [SENTIMENT].
    - This prompt template is based on the third instructional prompt, but also shares a lot of similarity to the second question prompt. It again provides the options to the model, intended to guide the model to generate a response that is consistent with the labels in the dataset.

4.5.3 Proposed Experiments

This section provides a brief overview of the experiments proposed for creating and evaluating the sentiment classifier. The experiments are designed to answer RQ1, RQ2, and RQ3. The experiments are designed to evaluate the effectiveness of the proposed prompts, as well as the effectiveness of the proposed models.

Experiment 1: Standard Models with Prompts

The first set of experiments is designed as a baseline for this system. The goal of this experiment is to evaluate the PLMs without any additional fine-tuning to see how they perform on the sentiment classification task when prompts are applied. This experiment will provide insight for RQ1 as it demonstrates one method that prompting can be applied for this task, as well as RQ2 as it demonstrates various suitable language models to perform prompt-based movie sentiment analysis. Additionally, it serves as a "usefulness" check for Experiment 2. If the models perform better on the following task, it validates our hypothesis that prompt learning will improve the model performance.

Using the prompts proposed in the previous section, we will evaluate each model's performance on the sentiment classification task. This will allow us to evaluate both the prompts used and the models themselves, giving insight into effective prompts for this task and the performance of models.

Experiment 2: Prompt Learning and Prompting of Models

The second set of experiments is designed to demonstrate the impact of prompt learning on the performance of these models. The goal of this experiment is to perform fine-tuning on the PLMs using the prompts proposed in the previous section and evaluate their performance on the sentiment classification task. This experiment will provide insight into RQ1 and RQ2 as it again demonstrates prompting methods for sentiment classification, as well as RQ3 since the results of this experiment can be compared to traditional ML methods to determine if prompt learning improves the performance of these models. The results of this experiment will prove or disprove our hypothesis that incorporating prompt learning will improve the model's performance.

To conduct this experiment, we will begin by creating a dataset for the fine-tuning of our models. We will use the training dataset and the prompts proposed in the previous section. We will complete the prompts such that they provide the model with the intended behaviour on each of them. In particular, we will only use a limited set of prompts for this task, allowing us to also evaluate the systems on previously unseen prompts. In this example, we will use Instructional Prompts 1 and 2, Completion Prompts 1 to 4, and Question Prompts 1 and 2. This holds Instructional Prompt 3, Completion Prompts 4 and 6, and Question Prompt 3 aside to evaluate as "unknown" prompts. We will then fine-tune each model on the dataset and evaluate their performance on the sentiment classification task.

Once again, we will use the prompts proposed in the previous section to evaluate each model's performance on the sentiment analysis task when prompt learning has been applied. This will allow us to evaluate the prompts and models themselves again and compare the results to the previous section to validate whether prompt learning has a positive impact on these results. Assuming the results of this experiment are better than the previous experiment, we will select the best-performing model for use in the proposed system.

4.5.4 Evaluation

Selecting useful evaluation metrics is a critical step to assess the performance of ML systems. These metrics provide objective measures to compare the performance of different models, which is particularly useful for the comparison of models for RQ2 and for comparison with existing baselines for RQ3. Useful metrics also provide strong insight into the performance of a model, such as where it struggles and where it excels. This section will describe the evaluation metrics applied to the sentiment classifier.

The standard method used with other models on the IMDB sentiment analysis task is to use accuracy as the evaluation metric. Accuracy is defined as the number of correct predictions divided by the total number of predictions. This metric is useful as it provides a simple and straightforward measure of the model's performance. However, it does not provide a complete picture of the model's performance. For example, if the model predicts the correct label for 90% of the reviews, it would have an accuracy of 90%. However, this does not tell us how the model performs on positive reviews versus negative reviews. It is possible that the model performs well on positive reviews but poorly on negative reviews, or vice versa. This is why we will also use precision, recall, and F1-score as evaluation metrics.

These metrics are described in detail in Chapter 2. The combination of these four metrics will provide strong insight into the strengths and weaknesses of each model, while also allowing us to compare the performance with other models to address RQ3.

4.6 Explainability

The second effort of this paper is to create the first Explainable Artificial Intelligence (XAI) for a prompt learning system. We will create an XAI system that explains the

predictions of the sentiment classifier models described in the previous chapter using several different approaches, including both post-hoc and self-explaining systems.

This section will describe the approach taken to create the XAI system, including the specific methods chosen for evaluation, the experiments conducted to evaluate their effectiveness, and how those experiments will be evaluated. We will seek to address RQ4 by exploring how XAI techniques can be integrated into a prompt learning system, and RQ5 by evaluating the quality of the explanations provided by the system.

4.6.1 Selected Methods

We seek to evaluate the effectiveness of several different XAI methods on the sentiment classifier. Where possible, we will evaluate both post-hoc and self-explaining models to determine the effectiveness of each approach. This section will describe the XAI methods selected for evaluation.

Post-hoc Methods

As described in Chapter 2, a post-hoc explainer is a system applied after a model has been trained. This enables explanations using any type of model, as post-hoc methods do not require access to the internals of the model it needs to explain. For this work, this is particularly important since OpenAI's GPT models are not open source and we therefore cannot access the internals of the model. We will evaluate the following post-hoc methods:

- **LIME -** LIME (Ribeiro, Singh, and Guestrin 2016) is a popular post-hoc explainer that uses a local surrogate model to explain the predictions of a black-box model. LIME is a popular choice for text classification tasks, making it a suitable choice for this work.
- **SHAP -** SHAP (Lundberg and S.-I. Lee 2017) is a popular post-hoc explainer that uses a game theoretic approach to explain the predictions of a black-box model. SHAP is a popular choice for text classification tasks, making it a suitable choice for this work.
- **Integrated Gradients** - Integrated Gradients (Sundararajan, Taly, and Yan 2017) is another post-hoc explanation method that measures how each input

feature contributes to the prediction by following a path from a baseline (such as all padding strings) to the input. This is another post-hoc method as it does not require access to the internals of the model.

We will apply these methods to the sentiment classifier described in the previous chapter to evaluate the strengths and weaknesses of each approach. Additionally, we will also apply strategies discussed in the literature review to create human-interpretable explanations from these methods. This will allow us to evaluate the effectiveness of these strategies in the context of prompt learning.

Self-explaining Methods

As described in Chapter 2, a self-explaining model is a model that is inherently designed to be understood by humans. This typically requires access to the internals of the model, which can be challenging with modern language models. We will evaluate the self-explaining approaches explored in the literature review, including:

- **SELFEXPLAIN -** SELFEXPLAIN is a model discussed in Chapter 3. It enables both global and local explanations using two additional layers that augment the neural classifiers. The global explanation layer pulls features from the model's training data to justify a decision, while the local explanation layer pulls features from the specific input sample to justify a decision. This method has been applied with RoBERTa previously, so we will evaluate its effectiveness with RoBERTa for this task. Modifying this approach to work with other models would require significant modifications that are beyond the scope of this work.
- **Generated Explanations -** Another approach for language models is to simply include a prompt that generates an explanation. This requires caution, as the statistical models may not reflect the underlying decision-making process, so we will evaluate this approach with caution and with emphasis on ensuring it reflects the model's decision-making process.

4.6.2 Proposed Experiments

This section provides a brief overview of the experiments proposed for creating and evaluating the explainability component. These experiments are designed to address

RQs 4 and 5. We will determine how we can integrate explainability into a prompt-based movie sentiment classification system for human-understandable explanations, then we will evaluate the quality of these explanations.

Experiment 1: LIME and SHAP

Due to the simplicity of the LIME and SHAP methods, we will begin by evaluating these methods. We will apply these methods to the sentiment classifier described in the previous chapter to evaluate the strengths and weaknesses of each approach. Additionally, we will also apply strategies discussed in the literature review to create human-interpretable explanations from these methods. This will allow us to evaluate the effectiveness of these strategies in the context of prompt learning.

Experiment 2: Integrated Gradients

Integrated Gradients is another post-hoc explainer that we will evaluate with our sentiment classifier. We will implement a version of the algorithm for text classification tasks and apply it to our sentiment classifier. We will then evaluate the effectiveness of this method for explaining the predictions of the sentiment classifier.

Experiment 3: SELFEXPLAIN with RoBERTa

The third experiment will evaluate the effectiveness of the SELFEXPLAIN model with RoBERTa. We will implement the method and apply it to the RoBERTa sentiment classifier created in the previous section. Due to limitations with the OpenAI API, we do not have access to the internals of the GPT models and therefore cannot apply this method to them. In contrast, SELFEXPLAIN has been previously applied to RoBERTa, so we will evaluate its effectiveness with RoBERTa for this task.

Experiment 4: Generated Explanations

Our novel approach for this problem is that we will evaluate whether prompting the model to provide an explanation is an effective method for generating explanations. We will create a prompt that asks the model to provide an explanation for its prediction and evaluate the effectiveness of this approach. This will allow us to evaluate the effectiveness of this approach for generating explanations. As mentioned before,

we must use caution and ensure that the generated explanations reflect the model's decision-making process.

4.6.3 Evaluation

We will closely follow the proposed evaluation metrics in Rajagopal et al. 2021 to evaluate our explanations. These metrics are briefly described in Chapter 3, but we will provide a more detailed description here. We will evaluate the explanations using the following metrics:

- **Adequate Justification:** Adequate justification is whether the explanation provides sufficient reasoning to justify the prediction of the model. In Rajagopal et al. 2021, they evaluate this by providing the input, label, prediction, and explanation and asking them "Does the explanation adequately justify the model prediction?" with a binary yes or no answer. We will follow this approach to evaluate the adequacy of our explanations.

- **Understandability:** Understandability is a metric of the ability of a user to understand the model explanation. In Rajagopal et al. 2021, human annotators are asked to compare explanations with a baseline to evaluate the understandability. In our case, we do not have baseline explanations. Instead, we will use a similar approach to the above and ask "Is the explanation understandable?" with a binary yes or no answer.

- **Trustworthiness:** Trustworthiness measures whether an explanation instills a sense of trust in users. In Rajagopal et al. 2021, they provide human annotators with the explanation and prediction and asked them to rate on a Likert scale of 1-5 how much they trust the model prediction. Note that judges would not be given the true label in this case. We will follow this approach to evaluate the trustworthiness of our explanations.

- **Sufficiency:** Sufficiency "aims to evaluate whether model explanations alone are highly indicative of the predicted label" ('Rajagopal et al. 2021). To evaluate sufficiency, they train a BERT classifier using generated explanations alone. If the classifier achieves a high accuracy, then the explanations do reflect the model's predictions. We will follow this approach to evaluate the sufficiency of our explanations.

These metrics will provide strong insight into the quality of the explanations provided by our system. They will allow us to evaluate the effectiveness of each method and compare the results to determine the most effective method for this task.

4.7 Summary

In this chapter, we described the general architecture of the proposed system. We began with a high-level overview of the system, describing the inputs and outputs of the system, as well as the components that make up the system. We then described the specific approach used to create each component of the system, beginning with the dataset and preprocessing steps, then the sentiment classifier, and finally, the explainability of the system.

We described the experiments to be performed to create prompt-based sentiment classifiers, then the experiments to create and evaluate various XAI methods for the classifier. We also described the evaluation metrics to be used for each experiment, which will allow us to evaluate the effectiveness of each component of the system. These experiments will be conducted in the next chapter, where we report and discuss the results obtained from conducting them.

Chapter 5

Experiments and Discussion

This chapter will describe the experiments conducted to create and evaluate the sentiment classifier and the XAI system. This chapter will begin with an overview of the experiments conducted to create the sentiment classifier and the results of those experiments. It will then discuss the experiments conducted to create and evaluate the XAI system and the results of those experiments. Finally, the results of these experiments will be the subject of further discussion.

5.1 Sentiment Classifier

As described in the previous chapter, the sentiment classifier constitutes the core of the proposed system. This section will describe the experiments conducted to create and evaluate the sentiment classifier. As described in the methodology, we will conduct two experiments to evaluate the effectiveness of the proposed prompts and models:

1. **Experiment 1: Prompting Standard Models -** We will evaluate the PLMs without any additional fine-tuning to see how they perform with the prompts, establishing a baseline for the following experiment.
2. **Experiment 2: Fine-tuning and Prompting Models -** We will perform fine-tuning on the PLMs using the prompts and repeat the earlier experiment to demonstrate the impact of prompt learning on the performance of these models.

These experiments are designed to address RQs 1, 2, and 3. We will determine how we can apply prompt learning techniques to movie sentiment classification, we will determine what the most suitable models and architectures are for this task, and we will compare the results of these models to existing baselines. The best performing model from these experiments will be selected for use in the proposed system alongside the selected explanation method.

5.1.1 Experiment 1: Evaluation of Standard Models with Prompts

The first experiment is designed to demonstrate the effectiveness of the proposed prompts for sentiment classification. The goal of this experiment is to evaluate the performance of the PLMs on the sentiment classification task using the prompts proposed in the previous section without performing any additional fine-tuning or further training. This experiment will provide insight into RQ1 and RQ2 as it demonstrates the effectiveness of the proposed prompts for sentiment classification. We will prove or disprove our hypothesis that the proposed prompts are effective for sentiment classification by comparing the results of this experiment to those of Experiment 2.

This section will describe how the experiment is conducted, then provide an overview of the results with some discussion and analysis of the results.

Experiment Design

To conduct this experiment, we are simply using each model and evaluating its performance on the test set when we apply each of the proposed prompts. While this is a simple approach, there is a slightly different approach using OpenAI's GPT models compared to BERT-based or ELECTRA models, since GPT is used through the OpenAI API while the others use transformers. This section will describe the approach for each model.

BERT-based and ELECTRA Models

We use a common script for BERT, RoBERTa, DistilBERT, and ELECTRA models using the transformers library. After loading the model and tokenizer, we initialize a fill-mask pipeline that will allow the model to complete the prompts. We create a function entitled craft_prompt that takes a review and an integer to represent which prompt to use, it then returns the prompt with the review and appropriate mask integrated. We then create a function entitled predict_sentiment, which takes

the review and prompt ID and calls craft_prompt to predict the sentiment. In some cases, the review is too long for a model (normally 512 tokens), in which case the review is broken into chunks, each combined with the prompt as well. The model then predicts and returns the sentiment. In the event that multiple chunks are needed, it will return the most common sentiment expressed.

This predict_sentiment function can be easily applied to all reviews using the apply method in the pandas library. This will apply the function to each review in the test set and return the results. We simply repeat this process for each prompt and then evaluate the performance of the model for each prompt using the accuracy_score, precision_score, recall_score, and f1_score functions from the sklearn.metrics library. This process can then be repeated for each model.

GPT Models

The script for GPT models has some similarities to the transformers models, but requires a different approach due to needing the openai package and API. We create a similar function craft_prompt that simply formats the review and prompt as needed. Unlike the previous models, we do not always require a masking token, as the openai package's Completion method will simply continue the sentence for the Instructional and Question prompts. For the Completion prompts, we use a token labeled [SENTIMENT] that indicates where it should complete. We again create a predict_sentiment function that calls the Completion method in the openai package using either the babbage or davinci models. It then returns the sentiment predicted by the model.

Similarly to before, we apply the predict_sentiment function to all reviews using the apply method in the pandas library, which applies the function to each review in the test set. We repeat this process for all of the prompts. We then evaluate the performance of the model for each prompt using the accuracy_score, precision_score, recall_score, and f1_score functions from the sklearn.metrics library. This process can then be repeated for each model.

Llama 2

Lastly, we use a similar approach to the BERT-based models for Llama 2. In this case, we are again able to run the language model locally using the transformers library. We create a craft_prompt function that takes a review and prompt ID and returns the prompt with the review integrated. We then create a predict_sentiment function that takes the review and prompt ID and calls craft_prompt to predict the sentiment. We then apply the predict_sentiment function to all reviews using the apply method in the pandas library, which applies the function to each review in

the test set. We repeat this process for all of the prompts. We then evaluate the performance of the model for each prompt using the accuracy_score, precision_score, recall_score, and f1_score functions from the sklearn.metrics library.

Results

This section describes the results of Experiment 1. This experiment provides a baseline for the experiment, as we can observe the performance of the models when using the prompts without any additional fine-tuning or training. This experiment will provide insight into RQ1 and RQ2 as it demonstrates the effectiveness of the proposed prompts for sentiment classification.

Table 5.1: Accuracy comparison for each model and prompt combination for Experiment 1.

	Prompts											
	Instructional			**Completion**						**Question**		
Model	**1**	**2**	**3**	**1**	**2**	**3**	**4**	**5**	**6**	**1**	**2**	**3**
bbu	44.00	45.54	40.96	76.82	66.00	39.34	59.30	43.82	50.26	42.78	46.68	43.90
blu	47.46	49.32	39.76	71.20	74.20	52.24	57.74	47.54	60.28	48.86	48.76	48.36
xrb	46.14	44.38	59.78	75.52	56.20	74.44	54.36	53.36	63.32	46.44	43.68	68.72
xrl	51.40	39.22	56.98	84.04	67.70	61.86	40.52	49.98	64.66	49.06	40.46	66.76
dbu	40.02	38.76	42.90	66.16	55.00	46.86	53.52	52.00	60.20	41.90	39.78	44.22
esg	38.74	70.00	67.56	63.16	59.50	49.56	47.18	52.70	66.58	52.08	69.08	72.46
ebg	42.26	49.02	74.36	76.98	38.40	58.12	52.90	59.86	50.82	36.12	47.86	76.32
elg	43.54	49.04	69.56	80.66	76.12	70.44	75.52	54.52	68.54	40.38	47.66	75.78
g35b	81.25	82.13	82.83	65.21	63.24	60.65	58.05	66.75	61.94	81.46	81.19	81.33
g35d	81.85	84.21	84.91	63.14	81.58	63.32	65.69	69.07	62.11	84.21	91.53	91.79
g4	85.11	99.56	95.12	95.35	99.42	99.53	99.31	97.50	93.02	76.74	97.67	97.67
llama	75.44	94.74	85.96	96.49	85.96	89.47	85.96	92.98	89.47	92.98	98.25	94.74

Table 5.1 depicts the results for each prompt with each model. These results can provide significant insights into the behaviour of these models without further fine-tuning.

One key observation is the significant impact of prompt engineering on the usage of the models. For example, the performance of Llama 2 appears to improve when it is given a Question Prompt, achieving its highest results. Meanwhile, the GPT-3.5 models struggle with Completion Prompts compared to either Instructional or Question Prompts. This illustrates the importance of prompt engineering, as it can significantly impact the performance of the models. In fact, a strong prompt on a weak model can outperform a weak prompt on a strong model.

The results also shed insights into the strengths of certain models. For example, BERT-based masked language models achieve better results on Completion Prompts than Instructional or Question. This is because the Completion Prompts are essentially masked language models, so they are well-suited to this task.

We can make various additional observations on the models based on these results:

- The GPT models significantly outperform most other models. We observe a minor improvement when we move from the babbage model of gpt-3.5 to the davinci model, then a significant improvement when using gpt-4. This illustrates the dominance of an LLM over previous models.
- Llama 2 is the only model competitive to the GPT models, achieving results slightly below gpt-4 but vastly exceeding the two gpt-3.5 models. This illustrates the strength of the Llama 2 model, achieving extraordinary results without any fine-tuning.
- For most BERT-based models, the use of a completion prompt is best. This is not unexpected, as BERT-based models are masked language models, meaning they learn to predict missing words, which is the exact behaviour needed for the completion prompts. They typically perform poorly on the instructional or question prompts.
- Performance of ELECTRA models vary wildly, but are typically comparable to the BERT-based models. No specific type of model seems to outperform the others, with all three types achieving broad results.

Overall, these results provide an interesting baseline against which we can compare the fine-tuned models of Experiment 2. This comparison is discussed further in this chapter, while the implications of these results on our proposed research questions is discussed in Chapter 5.

5.1.2 Experiment 2: Evaluation of Models with Prompt Learning Training

This experiment implements the concept of prompt learning for the sentiment classifier. Overall, the objective of this experiment is to perform further training on our selected PLMs using the prompts identified in the previous chapter, teaching the models to output the correct sentiment for each prompt. To do this, we will begin by setting up our dataset by combining the reviews with our prompts. We will then perform the further training. After this training process, we will repeat the tests performed in Experiment 1 and observe the impact of prompt learning on the models' performance.

Dataset Setup

Our dataset must be modified to include the prompts created in the previous chapter. This data will then be used for further training of the selected PLMs, teaching them how to output the correct sentiment when given a prompt. The dataset is modified by integrating the reviews into any desired prompts. As described in the previous chapter, we will only use a subset of 8 out of the 12 prompts for training, which will allow us to see if there's a positive impact from this prompt learning process on prompts that have not been seen before.

Table 5.2 depicts the process of integrating a review into a prompt. The original review is shown, then the prompt is shown with the review integrated into it. This process is repeated for each prompt, creating a new dataset with the augmented reviews. This review expresses a positive sentiment, so the desired output for each prompt is "positive". We will use this dataset to train the models to output the correct sentiment for each prompt.

We create a script to automatically generate this augmented dataset, which takes the original training set as input and outputs the augmented dataset with each review integrated into the eight prompts we will use. After using the script to create the augmented dataset, we will use it for further training of the models, as described in the next section.

Further Training the Models

This section describes the process of further training the models using the augmented dataset. The augmented dataset is used to train the models to output the correct sentiment for each prompt. There is a slightly different process used for GPT and the BERT or ELECTRA models. This section describes the process for each model.

BERT and ELECTRA Models

To fine-tune the BERT and ELECTRA models, we can follow a process proposed in HuggingFace for fine-tuning masked language models [1]. This process can be used for all of the BERT and ELECTRA-based models to create fine-tuned versions.

This process uses the DatasetDict from the HuggingFace library, which can be created from a CSV file. Using the augmented dataset contained in a CSV file, we simply call the .from_csv() function from the DatasetDict object to create a dataset

[1]The process for training BERT models, including code, is described here: `https://huggingface.co/learn/nlp-course/chapter7/3?fw=tf`

Table 5.2: Example of a review integrated with prompts used for the augmented set.

Prompt	**Output**
Original	This episode is certainly different than all the other Columbos though some of the details are still there the setup is completely different That makes this Columbo unique and interesting to watch even though at times you might wish for the old Columbo I liked it a lot but then I like almost any Columbo
IP1	Determine the sentiment of the following review: This episode is certainly different than all the other Columbos though some of the details are still there the setup is completely different That makes this Columbo unique and interesting to watch even though at times you might wish for the old Columbo I liked it a lot but then I like almost any Columbo. The sentiment is [SENTIMENT].
IP2	Indicate whether the following review is positive or negative: This episode is certainly different than all the other Columbos though some of the details are still there the setup is completely different That makes this Columbo unique and interesting to watch even though at times you might wish for the old Columbo I liked it a lot but then I like almost any Columbo. The sentiment is [SENTIMENT].
CP1	This episode is certainly different than all the other Columbos though some of the details are still there the setup is completely different That makes this Columbo unique and interesting to watch even though at times you might wish for the old Columbo I liked it a lot but then I like almost any Columbo. The review expresses a [SENTIMENT] opinion of the movie.
CP2	This episode is certainly different than all the other Columbos though some of the details are still there the setup is completely different That makes this Columbo unique and interesting to watch even though at times you might wish for the old Columbo I liked it a lot but then I like almost any Columbo. The user has a [SENTIMENT] opinion of the movie.
QP1	What is the sentiment of the following review? This episode is certainly different than all the other Columbos though some of the details are still there the setup is completely different That makes this Columbo unique and interesting to watch even though at times you might wish for the old Columbo I liked it a lot but then I like almost any Columbo. The sentiment is [SENTIMENT].
QP2	Is the following review positive or negative? This episode is certainly different than all the other Columbos though some of the details are still there the setup is completely different That makes this Columbo unique and interesting to watch even though at times you might wish for the old Columbo I liked it a lot but then I like almost any Columbo. The sentiment is [SENTIMENT].

object. For simplicity, we rename the columns to "text" and "label" for this process. We then simply replace the dataset loaded in the proposed code with the newly created dataset object. The rest of the code can be used as-is to fine-tune the model, allowing us to evaluate all the proposed models using the same process as Experiment 1.

GPT Models

The fine-tuning on GPT models can only been done using the OpenAI command line with their API, as it is a closed-source model. We begin by preparing the training data, which must be in a JSONL document where each line represents a prompt-completion pair. For example, Instructional Prompt 1 and 2 from Table 5.2 would be structured as follows:

```
{"prompt": "Determine the sentiment of the following review: This
episode is certainly different than all the other Columbos though
some of the details are still there the setup is completely different
That makes this Columbo unique and interesting to watch even though
at times you might wish for the old Columbo I liked it a lot but then
I like almost any Columbo. The sentiment is [SENTIMENT].",
"completion": "positive"}
{"prompt": "Indicate whether the following review is positive or
negative: This episode is certainly different than all the other
Columbos though some of the details are still there the setup is
completely different That makes this Columbo unique and interesting
to watch even though at times you might wish for the old Columbo I
liked it a lot but then I like almost any Columbo. The sentiment is
[SENTIMENT].", "completion": "positive"}
```

We create a script to automatically convert our CSV to this format, and then we upload the JSONL file to the OpenAI API. We then run either of the following commands to fine-tune the model, depending on whether we fine-tune babbage or davinci:

```
openai api fine_tunes.create -t "sentiment.jsonl" -m "babbage" -n
"imdb-sentiment-analysis-babbage"
openai api fine_tunes.create -t "sentiment.jsonl" -m "davinci" -n
"imdb-sentiment-analysis-davinci"
```

The fine-tuning process is then handled by the OpenAI API. Once it is complete, we can use the fine-tuned model using the same code as for Experiment 1, replacing the model name with the name of the fine-tuned model.

Llama 2

Similarly to BERT and ELECTRA, we can follow a process proposed in previous works to fine-tune Llama 2 using our dataset [2]. This process can be used for Llama 2 to create a fine-tuned version. Note that we use Google Colab for this training process, as it requires stronger resources than our local machine. We use a T4 GPU with increased RAM on the Colab environment. We follow the process described in the article using the augmented dataset, which allows us to evaluate Llama 2 using the same process as Experiment 1.

Results

This section describes the results of Experiment 2. After fine-tuning our models in the previous step, we can evaluate them on the test set exactly as we did in Experiment 1. We again discuss the results for each type of prompt, then conclude with a discussion on the global performance of the system. The complete results for all models can be found in Appendix A, while this section provides an overview of the results.

Table 5.3: Accuracy comparison for each model and prompt combination for Experiment 2.

	Prompts											
	Instructional			Completion						Question		
Model	1	2	3	1	2	3	4	5	6	1	2	3
bbu	49.65	50.16	44.32	79.96	70.08	46.83	66.79	45.21	53.01	44.12	48.15	44.16
blu	55.67	56.82	42.95	79.66	80.32	45.31	65.32	45.33	51.95	48.95	49.98	44.50
xrb	53.14	54.87	68.33	82.05	67.03	80.37	60.95	59.32	67.21	52.75	51.67	70.83
xrl	62.38	53.85	62.87	91.37	79.53	78.42	60.36	59.58	77.32	57.59	55.86	78.93
dbu	50.37	51.67	53.34	71.19	63.80	56.29	59.97	61.39	69.58	48.77	46.05	51.37
esg	46.32	80.71	69.37	69.18	65.47	55.31	53.98	55.61	68.09	58.68	76.55	74.90
ebg	47.32	55.05	75.11	80.32	49.32	63.96	60.01	59.97	51.05	48.12	59.12	76.03
elg	56.69	57.22	72.58	87.95	82.36	79.86	82.20	58.32	70.06	52.10	58.37	77.98
g35b	86.64	86.13	88.29	78.39	81.02	79.90	77.48	71.18	73.54	83.23	91.87	91.06
g35d	84.55	87.37	91.88	70.13	89.33	69.28	72.18	75.06	68.35	75.76	94.21	95.11
llama	83.44	96.25	90.12	98.51	89.99	93.40	90.11	95.61	93.38	94.76	98.53	96.05

Table 5.3 depicts the results for each prompt with each model. These results provide insights into the behaviour of these models after further training. These results can also be compared to Experiment 1 to view the impact of this process, as we do in the following subsection.

[2]The process for training Llama 2, including code, is described here: `https://towardsdatascience.com/fine-tune-your-own-llama-2-model-in-a-colab-notebook-df9823a04a32`

We again observe the significant impact of prompt engineering, even after further training. The performance of all models varies wildly based on the prompt used. In this example, the xlm-roberta-large model improves to a range between 53.85% and 91.37%, the former being a fairly poor result that only marginally improves on a 50% random-guessing baseline and the latter outperforming most gpt results. This further illustrates the importance of a well-crafted prompt, even after further training.

We make a few additional observations on these results:

- Llama 2 stands alone as the best fine-tuned model in this evalaution, as gpt-4 is unable to be fine-tuned. The accuracies obtained by Llama 2 have improved over the non-fine-tuned version and dominate this comparison. This demonstrates the strength of Llama 2 for this task, especially with further training.
- The GPT models continue to excel. Interestingly, the babbage model outperforms certain prompts for the davinci model, although the davinci model achieves higher maximum and average results. They still fail to outperform the gpt-4 model's results from Experiment 1, demonstrating its total dominance.
- BERT-based models also observe a small improvement over their results from Experiment 1. DistilBERT demonstrates the weakest results of this group, while the xlm-roberta-large model obtains strong results that sometimes outperform the GPT models. In fact, all BERT-based models excluding DistilBERT can outperform GPT with appropriate prompt engineering, and the best results of xlm-roberta-large even outperform Llama 2.
- ELECTRA models have some of the most interesting results, as they observe relatively large improvements over Experiment 1. In fact, electra-large-generator obtains fairly similar results to xlm-roberta-large, with many of its results outperforming the GPT models.

5.1.3 Comparison of Results Between Experiments

This section compares the results of Experiment 1 and Experiment 2 to determine if the results of Experiment 2 are better than Experiment 1. This comparison is important as it validates our hypothesis that prompt learning improves the performance of these models. This section will compare the results of each model between the two experiments, then conclude with a discussion on the results.

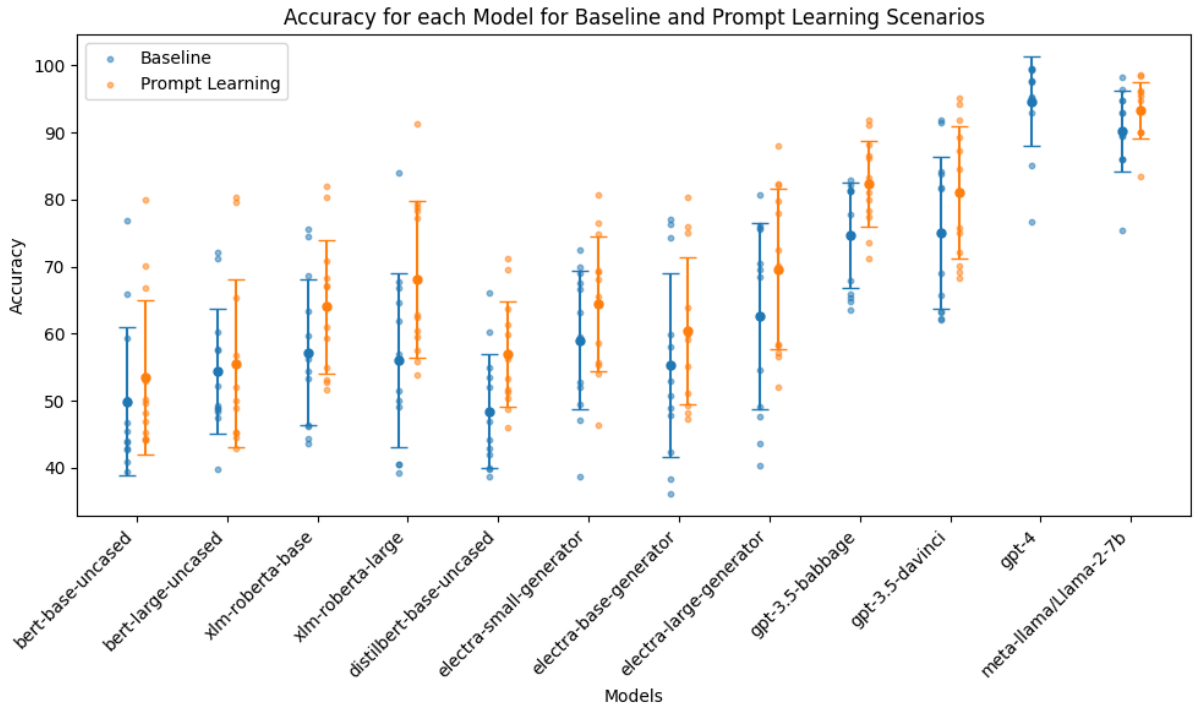

Figure 5.1: Comparison of results obtained in the baseline and fine-tuned experiments.

The model performance between the two experiments can be compared using Figure 5.1. Figure 5.1 depicts the raw data overlaid with the error bars for each model. We can observe a significant positive impact wherever prompt learning is applied with fine-tuning.

Using these figures, we observe a few interesting trends:

- Figure 5.1 demonstrates that the fine-tuning process has a positive impact on the performance of all models. All models observe an increase in average accuracy. This improvement ranges from small, such as a 1% improvement for bert-large-uncased, to large, such as a 12% improvement for xlm-roberta-large. This demonstrates that the fine-tuning process has a positive impact on the performance of these models.

- Figure 5.1 shows the usefulness of prompt engineering for the usage of these models. For each of the evalauted models, the error bars illustrate the massive

standard deviation in results on this task. This is due to the impact of prompt engineering, as some prompts are significantly more effective than others. This illustrates the importance of prompt engineering for the usage of these models.

- gpt-4 remains a dominant model, outperforming every other model. We are unable to perform any further training on this model due to limitations by OpenAI, but it is evident that fine-tuning is not necessary for it to achieve outstanding results. Even the fine-tuned Llama 2 model narrowly fails to achieve this level of performance.

- Llama 2 is the only model to achieve comparable results to gpt-4, falling slightly short. It sees minor improvements in both average and maximum accuracy when fine-tuned, but is unable to outperform gpt-4, illustrating the outstanding performance of these models.

Additionally, we can also make observations about the impact of fine-tuning on prompts that had not been used for training. Figure 5.2 depicts the prompts used for fine-tuning in blue and unseen prompts in orange for each model. We can make some interesting observations on this data:

- Globally, the benefit of fine-tuning typically applies to unseen prompts as well. This means that prompt learning can improve the results of prompts that had not been used for fine-tuning. This positive impact is seen on RoBERTa, DistilBERT, GPT-3.5, and Llama 2.

- For ELECTRA, this impact is much less noticeable. In fact, performance on the base generator even declines at some points. While seen prompts all greatly improve following prompt learning, the impact does not always affect unseen prompts.

- For BERT's large model, 3 of the 4 unseen prompts see a sizeable decrease in performance. Once again, this indicates that prompt learning does not benefit unseen prompts in this case.

Overall, this comparison demonstrates the usefulness of prompt learning for sentiment classification on all models for which this comparison can be performed. The outstanding results of gpt-4 without fine-tuning demonstrate the potential of LLMs for this task without any further training, but we are unable to determine the impact fine-tuning would have on its performance. For most models currently in use,

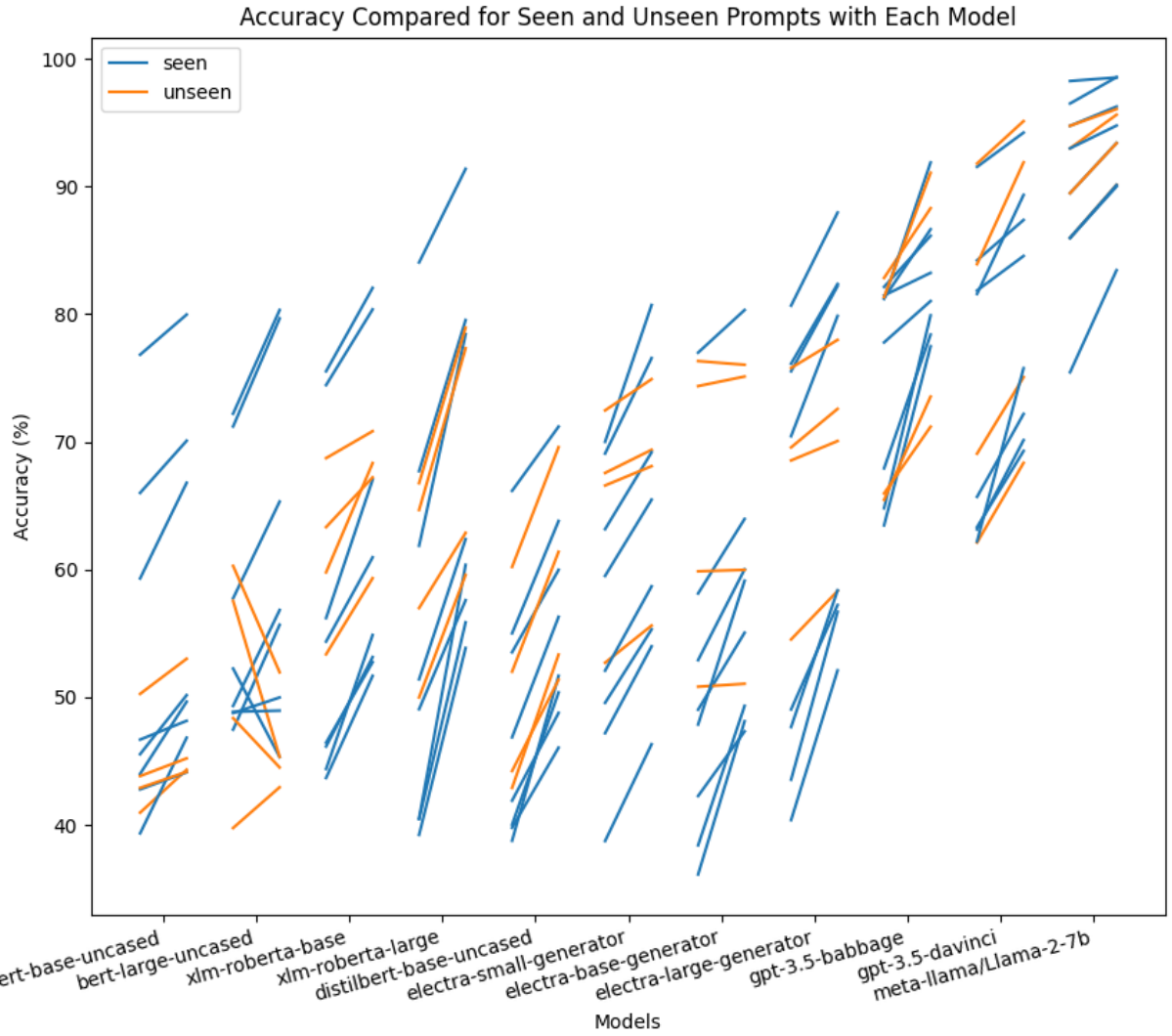

Figure 5.2: Impact of the prompt learning step on prompts used for training compared to previously unseen prompts.

fine-tuning has a positive impact on the performance of these models. This impact is even observed on prompts that had not been used for prompt learning in most cases, meaning that this fine-tuning benefits unseen prompts too. This illustrates the strength of this approach for this task.

5.1.4 Comparison to Existing Baselines

The best combinations of prompts and models from Experiment 2 are compared to some of the existing baselines and traditional approaches obtained from the PapersWithCode [3] reported results in Figure 5.3. This figure demonstrates the performance of these models compared to some of the existing baselines.

The comparison of these models demonstrates the strong performance of prompt learning-based models for this task, particularly when combined with effective prompt engineering.

We can make a few interesting observations based on this data:

- The Llama 2 fine-tuned model outperforms all other models, achieving a maximum accuracy of 98.53%. This is an extremely strong result, demonstrating the effectiveness of prompt learning for this task.
- The gpt-3.5-davinci fine-tuned model outperforms many reported results. Its results are within 1.1% of the best-reported model, showing the strong results of prompt learning and engineering for this task.
- The gpt-3.5-babbage fine-tuned model performs well, but is outperformed by most other models. This is likely due to the smaller size of the babbage model, which is unable to capture the same amount of information as the davinci model.
- The xlm-roberta-large model performs very well, narrowly outperformed by the gpt-3.5-babbage model. This is a strong result for an open-source model and illustrates the strong performance of BERT-based models for this task.

Overall, these four models are very effective at the IMDB sentiment analysis task, outperforming many of the existing baselines. This illustrates the effectiveness of prompt learning at achieving strong results for this task. We will analyze these four models further in the explainability section, attempting to produce explanations for their predictions.

[3] https://paperswithcode.com/sota/sentiment-analysis-on-imdb

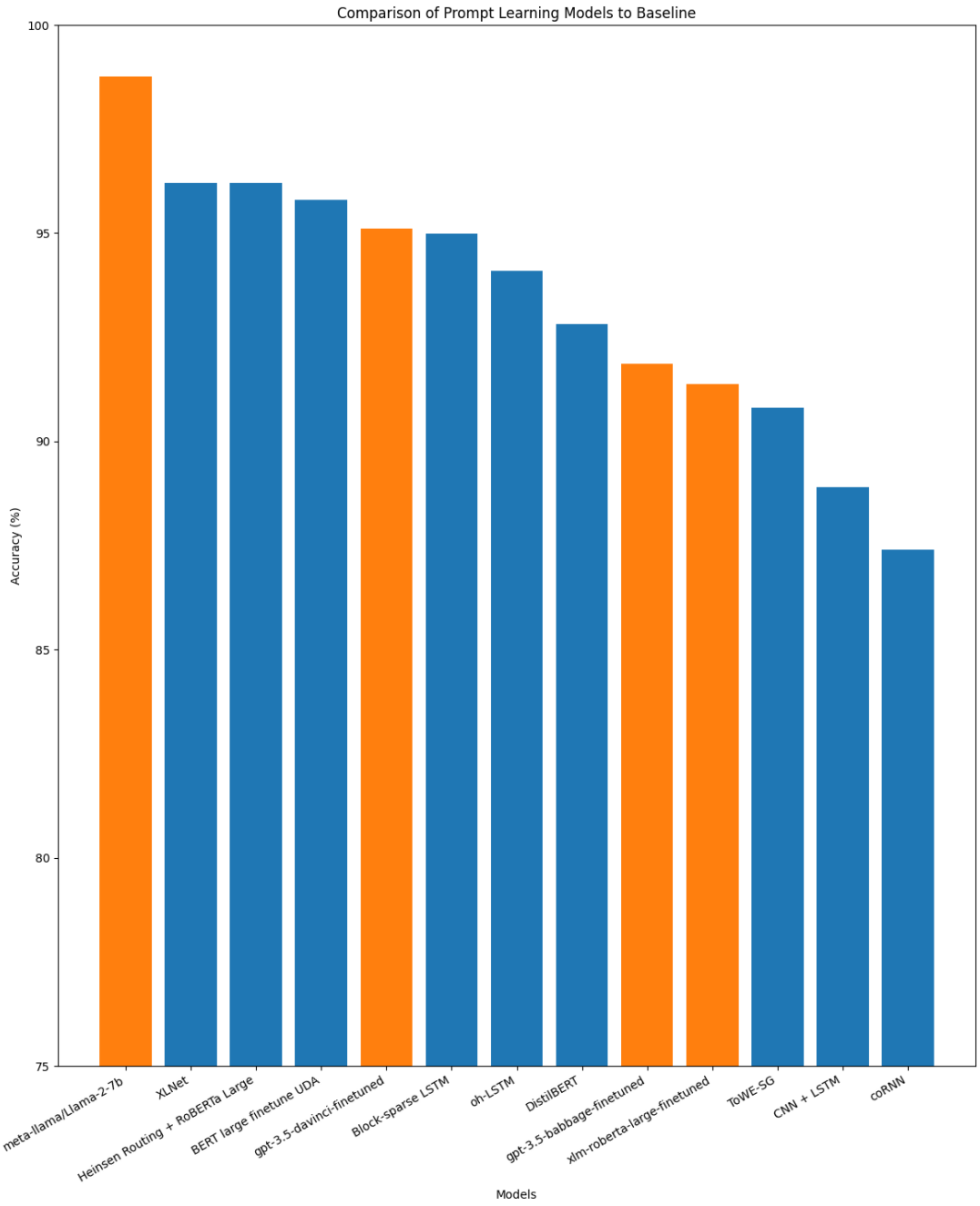

Figure 5.3: Maximum accuracy obtained for several prompt learning models compared to some existing baselines.

5.2 Explainability

As described in the methodology section, this chapter will describe the experiments conducted to create and evaluate various explainability methods for the sentiment classifier. As mentioned there, we will conduct the following experiments to evaluate the effectiveness of the proposed explainability methods:

- **Experiment 1: LIME and SHAP -** LIME and SHAP are two commonly used post-hoc explainability methods. LIME focuses on evaluating the impact of permutations on the result, while SHAP focuses on calculating Shapley Values to explain the decision. We will implement both of these methods and evaluate their effectiveness for explaining the predictions of the selected models for the sentiment classifier.
- **Experiment 2: Integrated Gradients with RoBERTa and Llama 2 -** Integrated Gradients is another post-hoc explainability method. Beginning with a baseline review (such as an empty review), we gradually progress towards the review to be explained, evaluating the impact of each token on the result. We will implement this method and evaluate its effectiveness for explaining the predictions of the selected models for the sentiment classifier.
- **Experiment 3: SELFEXPLAIN with RoBERTa -** SELFEXPLAIN is an approach discussed in the literature review for generating global and local explanations using two additional layers that augment existing neural classifiers. It has been implemented with RoBERTa previously, so we will implement it and evaluate its effectiveness for explaining the predictions of the selected models for the sentiment classifier.
- **Experiment 4: Generated Explanations -** This approach simply involves asking the model to explain its predictions alongside the prompts used to generate the prediction. We will implement this approach and evaluate its effectiveness for explaining the predictions of the selected models for the sentiment classifier, with a particular emphasis on determining whether the explanations actually reflect the decision-making process of the model.

These experiments are designed to address RQ4 and RQ5, which focus on how XAI techniques can be integrated into the classifier for human-understandable explanations and the quality of these explanations. Using the metrics described in the

methodology, we will evaluate the proposed techniques and select the best-performing technique for use in the proposed system.

For each of the experiments, we will provide an example of a produced explanation for the following negative sample review, selected from the IMDB dataset:

```
This is a tedious movie The real villains are the clunky adaptation
its embarrassingly easy to tell that the source material was a novel
and witless screenplayOn the credit side considering the budget was
tight due to wartime austerity the look of the film isnt at all bad
And the performances are by and large OK except for Phyllis Calvert
who is terrific  a miracle considering the potential for winsomeness
a pit into which she most definitely does not fall Ms Calvert with
a lot less to go on is as accomplished as Olivia de Havilland in
Gone With The WindThe one absolutely unbearable aspect of The Man
in Grey is the dreadfully conceived depiction of a black serving boy
No matter that hes meant to be a sympathetic character Played badly
by a white boy in blackface makeup it is impossible to bypass this
example of condescending racismGrim
```

5.2.1 Experiment 1: LIME and SHAP

We begin with LIME and SHAP as they are both frequently used XAI systems that can be easily integrated into our system using simple Python libraries. We will implement both of these methods and evaluate their effectiveness for explaining the predictions of the selected models for the sentiment classifier. They can both be implemented and evaluated using all the selected models.

LIME

We begin by integrating LIME into the programs developed in the previous chapter. We use the lime library to implement LIME, which provides a simple interface for integrating LIME into our system. We can use the existing functions for predicting the review's sentiment for the RoBERTa, GPT, and Llama 2 models with the LimeTextExplainer object's explain_instance function. This function takes the review and the function as input and returns the explanation.

An example explanation generated using the babbage version of GPT can be seen in Figure 5.4. This output shows the predicted sentiment for the review and the top

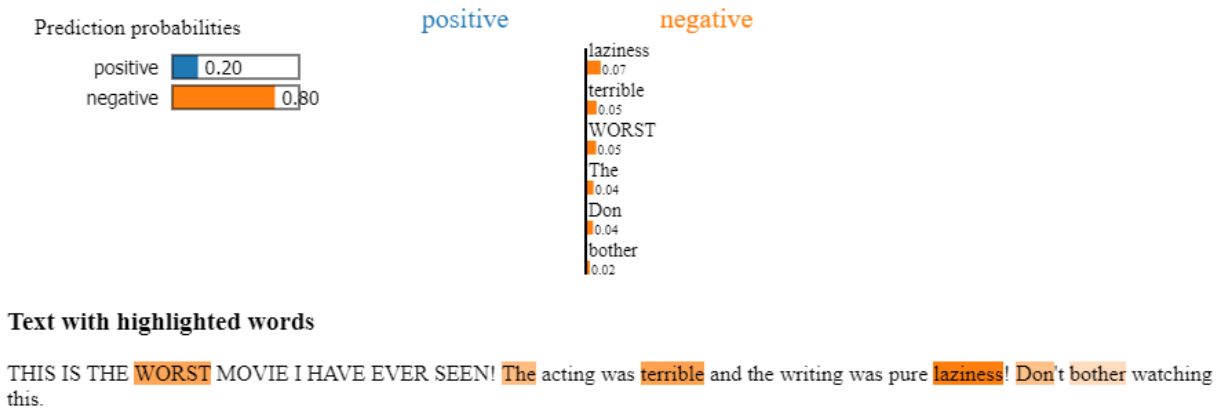

Figure 5.4: Example of a LIME explanation for a sample negative review.

tokens that impacted the prediction. Conveniently, it also shows the original review with these tokens specified. In this example, words like "laziness", "terrible", and "worst" had the most significant impact in the "negative" prediction. While this describes the model's prediction well, it is not very human-understandable for the average person. As such, we create an approach to convert this explanation to a more easily understandable format.

Using the explanation as an output, we can create a very simple human-understandable explanation by considering the scores assigned to the tokens. For example, with the review shown in Figure 5.4, we can create the explanation "Words like 'laziness, terrible, WORST, Don, bother, The' make the review seem negative to the model. Overall, the review is judged as negative."

This explanation is extremely simple but explains the output of LIME in a way that a human can easily understand. We can use this approach to create human-understandable explanations for any review using LIME. We apply this approach to the 100 reviews contained in the reduced explainability subset for each of the selected models. Examples of explanations generated using this approach can be found in Table 5.4, the full set will be evaluated in the results section.

We can observe some interesting issues with the example explanations provided in Table 5.4. In particular, they have the tenancy to identify useless or neutral terms as key input features. For example, identifying the word "the" or "This" as impacting the prediction. This trend occurs in various forms across all models evaluated and could lead to problematic results as they do not justify a prediction on their own.

Table 5.4: Examples of LIME explanations of a sample review using each selected model.

Model	Explanation
xlm-roberta-large	Words like 'the' make the review seem positive to the model. However, words like 'absolutely', 'impossible', 'tedious', 'isnt', 'depiction' make the review seem negative to the model. Overall, the review is judged as negative.
text-babbage-001	Words like 'racismGrim', 'example' make the review seem positive to the model. However, words like 'This', 'condescending', 'is', 'Played' make the review seem negative to the model. Overall, the review is judged as negative.
text-davinci-003	Words like 'OK', 'example' make the review seem positive to the model. However, words like 'embarrassingly', 'condescending', 'unbearable', 'impossible' make the review seem negative to the model. Overall, the review is judged as negative.
meta-llama/Llama-2-7b	Words like 'OK', 'terrific' make the review seem positive to the model. However, words like 'tedious', 'white', 'villains', 'bad' make the review seem negative to the model. Overall, the review is judged as negative.

SHAP

The second XAI technique we apply is SHAP. Similar to our implementation with LIME, we can simply use the shap library to implement this technique with our sentiment classifier system. Using the shap.KernelExplainer function, we can create an explainer object that can be used to explain the predictions of the models. From that object, we can use the shap_values function to retrieve the Shapley values for the review. These values can then be used to create an explanation for the review.

This approach produces the Shapley values for each token in the review, which can be used to create an explanation for the review. We can use a similar approach as with LIME to create a human-understandable explanation for the review. For example, with the review shown in Figure 5.4, we can create the explanation "The word 'laziness' has a significant negative impact on the model's prediction. The word 'terrible' has a moderate negative impact on the model's prediction. The word 'WORST' has a significant negative impact on the model's prediction. The word 'The' has a negligible negative impact on the model's prediction. The word 'Don'

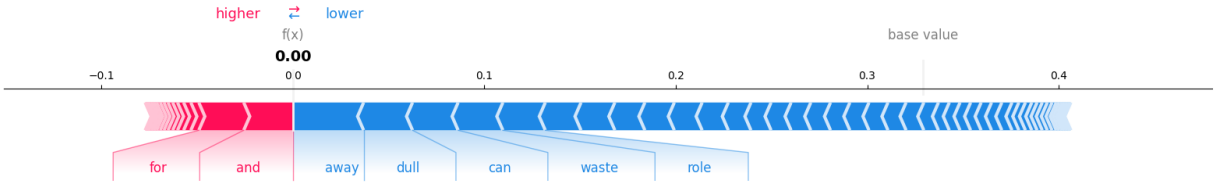

Figure 5.5: Example of a SHAP explanation for a sample review.

has a negligible negative impact on the model's prediction. Overall, the review is judged as negative.".

Examples of explanations for the sample review can be found in Table 5.5, depicting the output of the selected models using SHAP. We can observe some similarities between the outputs of LIME and SHAP, as they identify some of the same features as important towards a model's prediction. This is a positive indication of fidelity between these models and the reasoning of the sentiment classifier. We hope to continue observing this trend with the other explainability methods. The full set of explanations generated using SHAP will be evaluated in the results section later in this chapter.

Similar to LIME, though, we can observe that neutral words seem to be chosen as a feature. For example, the word "and" is selected as a negative feature, despite not providing a negative meaning.

5.2.2 Experiment 2: Integrated Gradients with RoBERTa and Llama 2

The second experiment we conduct is with Integrated Gradients. This approach is a post-hoc explainability method that gradually progresses from a baseline review to the review to be explained, evaluating the impact of each token on the model's prediction. Unlike the previously implemented SHAP or LIME methods, there is not a standard library that makes this approach easy to implement. As such, we must implement this approach ourselves. This requires access to the model's internals, which is regrettably not possible with the GPT models. As such, we can only implement this approach for the RoBERTa and Llama 2 models.

A simplified approach to implementing Integrated Gradients is the following:

Table 5.5: Examples of SHAP explanations of a sample review using each selected model.

Model	Explanation
xlm-roberta-large	The word 'depiction' has a significant negative impact on the model's prediction. The word 'in' has a significant negative impact on the model's prediction. The word 'itself' has a significant negative impact on the model's prediction. The word 'keep' has a negligible positive impact on the model's prediction. The word 'example' has a negligible positive impact on the model's prediction. Overall, the review is judged as negative.
text-babbage-001	The word 'OK' has a moderate positive impact on the model's prediction. The word 'dull' has a minor positive impact on the model's prediction. The word 'waste' has a moderate negative impact on the model's prediction. The word 'dumb' has a minor negative impact on the model's prediction. The word 'and' has a negligible negative impact on the model's prediction. Overall, the review is judged as negative.
text-davinci-003	The word 'some' has a significant positive impact on the model's prediction. The word 'clunky' has a significant negative impact on the model's prediction. The word 'OK' has a minor positive impact on the model's prediction. The word 'badly' has a minor negative impact on the model's prediction. The word 'comes' has a negligible positive impact on the model's prediction. Overall, the review is judged as negative.
meta-llama/Llama-2-7b	The word 'terrific' has a moderate positive impact on the model's prediction. The word 'tedious' has a significant negative impact on the model's prediction. The word 'villains' has a significant negative impact on the model's prediction. The word 'bad' has a moderate negative impact on the model's prediction. The word 'white' has a negligible positive impact on the model's prediction. Overall, the review is judged as negative.

1. **Baseline Input:** We select a baseline input to begin the process. This baseline input x' is typically one that would result in a neutral prediction, such as an

empty review.

- For the RoBERTa model, we use a string containing only the padding token as the baseline input. The string is of the same length as the review to be explained.
- For the Llama 2 model, we use an empty review as the baseline input.

2. **Path Integration:** We generate a base from the baseline input x' and the review to be explained x. This base is a linear interpolation between the two inputs using N discrete steps. This base is generated using the following formula:

$$x_b^i = x' + \frac{i}{N} \times (x - x')$$

where x_b^i is the base at step i and N is the number of steps. This formula is applied to each token in the review, generating a base for each token. We can then compute the gradients of the model's prediction with respect to the base using the following formula:

$$\frac{\partial F(x_b^i)}{\partial x_b^i}$$

where F is the model's prediction function. This formula is applied to each token in the review, generating a gradient for each token. Lastly, we integrate the gradients to obtain the integrated gradients using the following formula:

$$\int_{x'}^{x} \frac{\partial F(x_b^i)}{\partial x_b^i} dx_b^i$$

This formula is applied to each token in the review, generating an integrated gradient for each token.

3. **Attribution:** We then attribute the integrated gradients to each token in the review, where the attributions reflect the importance of each token towards the prediction.

4. **Conversion to Human-Readable Format:** Using the attribution scores from the previous step, we can convert the attributions to a human-readable format using a similar approach to the SHAP experiment, where the tokens with the most significant impact are highlighted in the explanation, along with how they impact the prediction.

This approach is straightforward to implement for both RoBERTa and Llama 2. We implement this approach for both models and evaluate the results in the following section. Examples of explanations for the sample review can be found in Table 5.6, depicting the output of the selected models using Integrated Gradients. We can again observe various similarities with previously evaluated methods. This is further evidence of fidelity between these models and the reasoning of the sentiment classifier. The full set of explanations generated using Integrated Gradients will be evaluated in the results section later in this chapter.

Table 5.6: Examples of Integrated Gradient explanations of a sample review using each selected model.

Model	**Explanation**
xlm-roberta-large	The word 'unbearable' has a significant negative impact on the model's prediction. The word 'badly' has a significant negative impact on the model's prediction. The word 'terrible' in training data has a minor negative impact on the model's prediction. The word 'go' in training data has a minor positive impact on the model's prediction.
meta-llama/Llama-2-7b	The word 'tedious' has a moderate negative impact on the model's prediction. The word 'dreadfully' has a significant negative impact on the model's prediction. The word 'tight' in training data has a significant negative impact on the model's prediction. The word 'out' in training data has a minor negative impact on the model's prediction

5.2.3 Experiment 3: SELFEXPLAIN with RoBERTa

The third experiment we conduct is with the SELFEXPLAIN (Rajagopal et al. 2021) approach using RoBERTa. Unlike the previously implemented methods, this approach is not a post-hoc explainability method. Instead, it is an approach that augments the model with two additional layers that are trained to generate global and local explanations for the model's predictions. The code for the original SELFEXPLAIN work is available on GitHub [4] and the paper provides a detailed description of the approach. We will implement this approach and evaluate its effectiveness for explaining the predictions of the selected models for the sentiment classifier.

[4] `https://github.com/dheerajrajagopal/SelfExplain/tree/master`

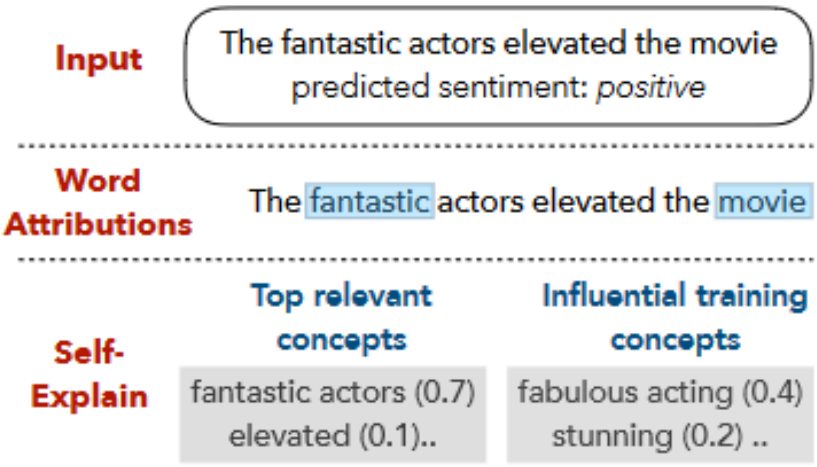

Figure 5.6: Example of a SELFEXPLAIN explanation for a sample review.

One strength of the SELFEXPLAIN approach is that this method provides insights into both the tokens that influenced its decision (for local explanations) and into the training that justifies the prediction (for global explanations). These insights cannot be obtained from post-hoc explanation approaches and are illustrated in Figure 5.6. In the output, we can view the relevant concepts from the input to the left. To the right, we view influential training concepts that were a factor in the model's decision. This provides a more complete explanation of the model's decision than the post-hoc approaches and give insight into the justification for the model's decision.

By default, this approach provides three terms from the input and three terms from the training data that influences its decision. We combine these reviews into a simple human-readable explanation like for the previous experiments and produce a simple explanation, as depicted in Table 5.7.

Table 5.7: Example of a SELFEXPLAIN explanation of a sample review.

Model	**Explanation**
xlm-roberta-large	The review is judged as negative. The words "unbearable", "clunky adaptation", "condescending" from the review influence its prediction. The terms "infuriating", "spiteful idiots", and "lazy" from the training data influence its prediction.

The explanation shown in Table 5.7 provides insight into the behaviour of the approach. It selects the three influential terms from the input and the three influential terms from the training data. This provides a more complete explanation of the model's decision than the post-hoc approaches and give insight into the justification for the model's decision. We anticipate that this will lead to more trustworthy explanations, as the model is able to justify its decision using the training data. We will evaluate the full set of generated explanations in the results section later in this chapter.

5.2.4 Experiment 4: Generated Explanations

The final experiment is to use the selected models themselves to generate explanations for their predictions. This approach is very simple, as it simply involves asking the model to explain its predictions alongside the prompts used to generate the prediction. For each selected model, we will simply append "Explain your reasoning" to the end of the prompt and evaluate the results. This approach is very simple to implement and can be applied to all selected models.

Table 5.8: Examples of generated explanations of a sample review using each selected model.

Model	Explanation
xlm-roberta-large	Negative.
text-babbage-001	The sentiment of the review is negative.
text-davinci-003	Negative. The review mentions that the movie is tedious, the adaptation is clunky, the screenplay is witless, and the depiction of a black serving boy is dreadfully conceived and played badly by a white boy in blackface makeup.
meta-llama/Llama-2-7b	The review is negative. The reviewer mentions the movie is tedious, has a witless screenplay, and they criticize the portrayal of a black serving boy. They do, however, give credit to the performances and the look of the film, but overall, the review is more negative than positive.

The generated explanations depicted in Table 5.8 provide interesting insights into the behaviour of these models. The RoBERTa model fails to provide any explanation, identifying only that the review is negative. Interestingly, the babbage variant of GPT-3.5 also fails to provide reasoning beyond its answer that the sentiment is

negative. In contrast, the davinci model provides a brief explanation using elements of the movie review to justify its decision. Llama 2 also uses elements of the movie review to justify its decision, but is also able to identify some positive elements presented in the review. This demonstrates the potential of these models to provide explanations for their predictions, but also the limitations of these models. We will evaluate the full set of generated explanations in the results section later in this chapter.

5.2.5 Evaluation

This section will discuss the evaluation and results of the explainability methods. We will begin with a discussion of the results according to each of the selected metrics. We will then discuss the results as a whole and compare the results achieved by each method and model. We will conclude with a discussion of the results of the experiments. The complete results of the evaluation can be found in Table 5.9 and are depicted in Figure 5.7 for visualization.

Due to constraints on the amount of time available to conduct the evaluation, we are unable to conduct a full human evaluation of all explainability methods. As such, we are focusing on a single model for the LIME and Integrated Gradients experiments. We select Llama 2 as it is the best-performing model for the sentiment classifier. We also evaluate the generated explanations of GPT-3.5 davinci and Llama 2. This ensures we can evaluate a diverse set of explainability methods with our best-performing model.

Human evaluation is being conducted by two human evaluators, consisting of one PhD student and a MSc student. The evaluators are given a questionnaire for each of the reviews in the reduced explainability subset, as described in the methodology. Additional information about the evaluators, including their agreement for each metric, can be found in Appendix B.

Sufficiency (Automatic Evaluation)

As described in the methodology, sufficiency is a metric that evaluates how well the explanation indicates the label predicted by the model. We use the FRESH pipeline (Jain et al. 2020), which uses a BERT classifier trained to perform the task using only the explanations without the rest of the input. If the explanation is sufficient, the classifier should be able to achieve a high accuracy. If the explanation is not sufficient, the classifier will achieve a poor accuracy.

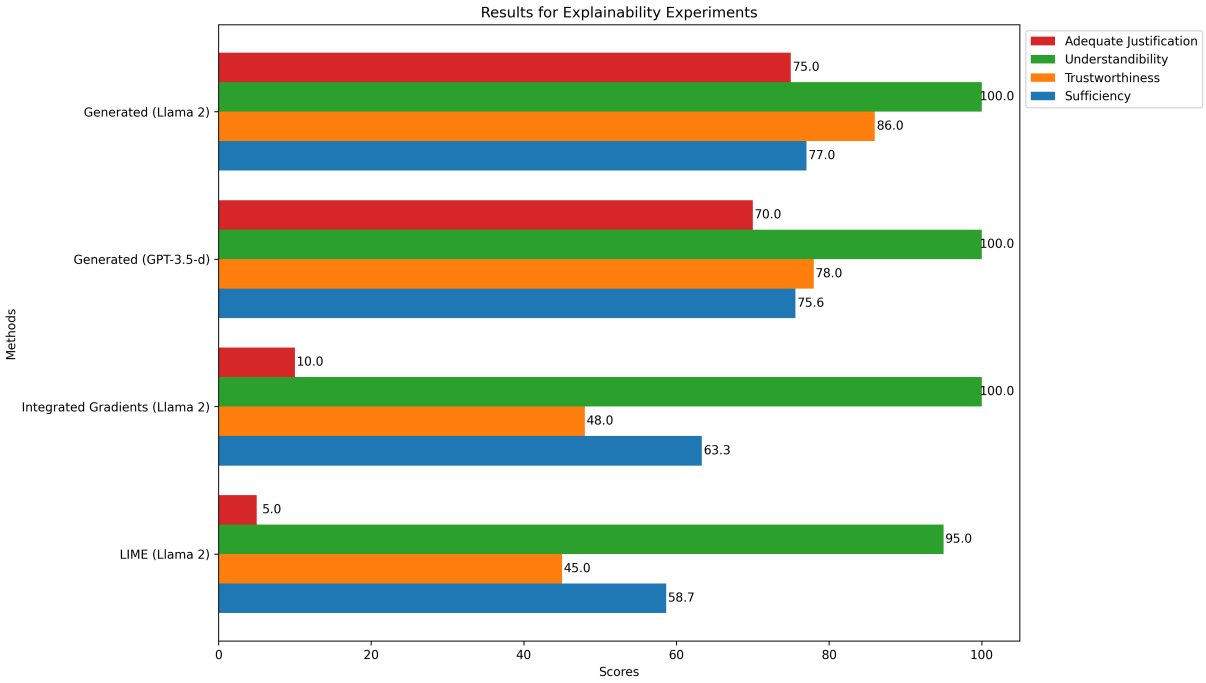

Figure 5.7: Results for the Explainability Methods

In Table 5.9 and Figure 5.7, we provide the accuracy of this trained classifier for the sufficiency score. We can make some key observations from this data:

- **LIME -** Sufficiency scores for LIME are relatively weak, all around 58%. This indicates that the explanations provided by LIME are not sufficient to explain the model's predictions. This is likely due to the fact that LIME is a post-hoc explainability method that struggles to explain the decision making of the model itself. This is evidence that post-hoc explainability methods are not sufficient for explaining the predictions of these models.

- **SHAP -** As observed for LIME, the sufficiency scores for SHAP are also weak, indicating that explanations produced with SHAP are not sufficient to explain the model's predictions. This is further evidence that post-hoc explainability methods are not sufficient for explaining the predictions of these models.

- **Integrated Gradients -** The sufficiency scores for the two Integrated Gradients approaches are marginal improvements over LIME and SHAP. The Llama

Table 5.9: Summary of Results for Explainability Methods

Method	**Adeq. Just.**	**Und.**	**Trust**	**Suff.**
LIME (RoBERTa)	x	x	x	57.34
LIME (GPT-3.5-b)	x	x	x	57.96
LIME (GPT-3.5-d)	x	x	x	59.09
LIME (Llama 2)	5.00	95.00	45.00	58.67
SHAP (RoBERTa)	x	x	x	54.98
SHAP (GPT-3.5-b)	x	x	x	55.60
SHAP (GPT-3.5-d)	x	x	x	56.42
SHAP (Llama 2)	x	x	x	56.38
Integrated Gradients (RoBERTa)	x	x	x	59.77
Integrated Gradients (Llama 2)	10.00	100.00	48.00	63.33
SELFEXPLAIN (RoBERTa)	x	x	x	70.81
Generated Explanations (RoBERTa)	x	x	x	58.07
Generated Explanations (GPT-3.5-b)	x	x	x	72.12
Generated Explanations (GPT-3.5-d)	70.00	100.00	78.00	75.60
Generated Explanations (Llama 2)	75.00	100.00	86.00	77.05

2 model outperforms RoBERTa with this approach. Still, these scores are not sufficient to explain the model's predictions. This is further evidence that post-hoc explainability methods are not sufficient for explaining the predictions of these models.

- **SELFEXPLAIN -** SELFEXPLAIN achieves reasonably strong sufficiency scores, indicating that the explanations are typically better at explaining the model's decision making. This is likely due to the fact that this approach is not a post-hoc explainability method, but instead augments the model with additional layers that are trained to generate explanations, which provides greater information about the model's decision-making process.

- **Generated Explanations -** The generated explanations outperform all other

methods for sufficiency. The GPT and Llama models outperform RoBERTa with this approach, as expected since they are large language models. This provides significant information used to justify the model's decision making process, which is likely why it outperforms the other methods for sufficiency.

Overall, the best-performing system for sufficiency is the generated explanations approach with the Llama 2 model. This approach is able to achieve a sufficiency score of 77.05%, which is a significant improvement over the other approaches. This demonstrates the effectiveness of generated explanations for explaining the predictions of these models.

Adequate Justification (Human Evaluation)

As described earlier, adequate justification is a metric to evaluate how well the explanation justifies the model's decision. We use human evaluation to evaluate this metric, as described in the methodology. We provide the results of this evaluation in Table 5.9 and Figure 5.7. We can make some key observations from this data:

- **LIME -** The LIME explanations are not typically able to justify a model's decision effectively. We can observe that the decision was only adequately justified 5% of the time with LIME. While LIME would often identify some elements that were useful for justifying the decision, it is prone to identifying irrelevant details as well. This is evidence that post-hoc explainability methods are not sufficient for explaining the predictions of these models.
- **Integrated Gradients -** The Integrated Gradients explanations are similarly unable to justify a model's decision adequately, achieving a result of 10%. Like LIME, Integrated Gradients would often identify some elements that were useful for justifying the decision, but it is also prone to identifying irrelevant details. This is further evidence that post-hoc explainability methods are not sufficient for explaining the predictions of these models.
- **Generated Explanations with GPT-3.5 davinci -** In contrast to the earlier models, GPT-3.5 typically justifies the decisions very well, achieving a score of 70%. It is capable of identifying key elements that justify the decision. However, it occasionally ignores the request to explain the decision and simply provides the decision itself, which is why it does not achieve a higher score. This demonstrates the potential of generated explanations for explaining the predictions of these models.

- **Generated Explanations with Llama 2 -** The Llama 2 model is able to justify its decisions very well, achieving a score of 75%. It is typically capable of identifying key elements that justify the decision. Similar to GPT-3.5, it sometimes states the decision without providing an explanation, weakening its performance on this metric. This demonstrates the potential of generated explanations for explaining the predictions of these models.

Overall, the best-performing system for adequately justifying a prediction is using a generated explanation from the Llama 2 metric. This approach achieves a score of 75%, and it is the best-performing system for this metric. This demonstrates the effectiveness of generated explanations for explaining the predictions of these models.

Understandability (Human Evaluation)

Understandability is simply an evaluation of how well the explanation can be understood by a human. We use human evaluation to evaluate this metric, as described in the methodology. We provide the results of this evaluation in Table 5.9 and Figure 5.7.

All methods achieve fairly strong results for understandability, with all methods achieving a score of 100% except for Llama 2 with LIME. This indicates that all methods are able to produce explanations that are understandable by humans. This is a positive indication of the effectiveness of these methods for explaining the predictions of these models. Our method for interpreting the results of LIME and Integrated Gradients into human-readable explanations is likely a factor in this success. The strong results for generated explanations also indicate that these explanations are easily understandable by humans, which is a positive indication of the effectiveness of these methods for explaining the predictions of these models.

Trustworthiness (Human Evaluation)

Trustworthiness is an evaluation of whether a human finds the explanations provided by the method to be trustworthy. As described in the methodology, we are using human evaluation to measure this metric. We provide the results of this evaluation in Table 5.9 and Figure 5.7. We can make some key observations from this data:

- **Generated Explanations -** The generated explanations achieve the strongest results for trustworthiness, with all models achieving a score of 100%. This in-

dicates that the generated explanations are trustworthy to humans. These explanations often contain strong justification for their decisions, which is likely why they are perceived as trustworthy. This demonstrates the potential of generated explanations for explaining the predictions of these models.

- **Integrated Gradients and LIME -** The explanations created through Integrated Gradients and LIME have relatively low trustworthiness, at under 50%. In contrast to the generated explanations, these justifications are often weaker and therefore convey less trustworthiness to humans. This is evidence that post-hoc explainability methods are not sufficient for explaining the predictions of these models.

The best-performing method for trustworthiness is using generated explanations from Llama 2. These explanations achieve an outstanding trustworthiness rating of over 80% from the human evaluation. This illustrates the effectiveness of generated explanations for explaining the predictions of these models.

5.3 Discussion

This section will discuss the results of the experiments conducted in this chapter. We will begin with a discussion of the results of our experiments and the proposed system as a whole. We will then discuss the research questions presented in the introduction, revisiting each question and discussing the results in the context of each question. We will conclude with a discussion of the limitations of this work.

5.3.1 Proposed System

Based on the results of Experiment 2 for the sentiment classifier, we found that Llama 2 achieves the best results with an accuracy of 98.53%. This is a significant improvement over the previous state-of-the-art XLNet system (Z. Yang et al. 2020). These results make it the obvious choice for inclusion in our proposed system.

We then explored a variety of approaches for explainability, including with the other models evaluated. Once again, we found that the Llama 2 model achieved the best results for explainability, even with weaker XAI approaches. We find that Llama 2's generated explanations are the best at adequately justifying their decisions and are the most trustworthy, according to human evaluators. We also found

that the generated explanations reflect the actual decision making of the model, as they are able to achieve strong sufficiency scores. This makes Llama 2's generated explanations ideal for inclusion in our proposed system.

This proposed system therefore achieves state-of-the-art accuracy for the sentiment classifier and provides strong generated explanations for the predictions of the sentiment classifier. This achieves the goal of this work, as we have created a system that can achieve state-of-the-art results for the sentiment classifier and provide strong explanations for the predictions of the sentiment classifier.

5.3.2 Are the Models Trained on the IMDB Dataset?

One concern is the possibility that the IMDB dataset used in this study has already been included in the training data for models evaluated, particularly the best performing Llama 2 model. This possibility raises the following concerns:

- **Overfitting:** LLMs may have already learned patterns and information from the data used. This could lead to the models performing very well during the evaluation but poorly on new, unseen data.
- **Lack of Generalization:** If the evaluation data is part of the model's training data, the model may not generalize well to new or unseen data. This could limit its utility in real-world applications where encountering novel data is common.
- **Difficulty in Assessing Progress:** If the evaluation data is part of the training data, it becomes challenging to accurately measure improvements in model performance. This is because improvements in performance on the evaluation data may simply reflect the model's ability to memorize rather than its ability to generalize and learn.

For many of the models evaluated, such as BERT, RoBERTa, or DistilBERT, the training data has been clearly established and provided by the model developers. This is important to ensure reproducibility by other researchers, transparency into potential biases or concerns, and to understand limitations of the system. However, recent models like GPT-3.5 or Llama 2 do not provide this information, creating confusion about how the model was trained, as shown with the possibility of the IMDB dataset being used.

To explore whether data was used in these closed-source models, we have a few possible options. Grynbaum and Mac 2024 describes an approach used by The

New York Times in justifying their allegations of copyright infringement by the developers of GPT. In that case, they prompted the model and alleged that the "chatbot provided users with near-verbatim excerpts from Times articles that would otherwise require a paid subscription to view". We can leverage a similar approach for exploring the use of IMDB data in the language models by attempting to solicit information about the reviews.

We examine a few different approaches to prompt this information from untrained models:

- We remove segments of the review and inquire about the missing information. If the model successfully discusses the missing information, it is an indicator that the system was trained on the data beforehand.
- We ask the model to provide samples of movie reviews for a movie and then match the output against test data. If there is a match, it is another indicator that the system was trained on the data beforehand.
- We provide a review and ask for new reviews to be generated based on it. The IMDB dataset includes multiple samples for the same movies, so we can attempt to match generated reviews.

We conducted these experiments with both GPT-3.5 and Llama 2. In both cases, we found no evidence that the IMDB dataset was used in training their models. The models were sometimes able to successfully identify the movie being reviewed or that it seemed to be an online review, given the presence of HTML tags. However, they provided no evidence that they knew the content of the reviews otherwise.

If the IMDB dataset has been included in the training set for any of these models, it should not include the IMDB test set. Our reported results are on the IMDB test set, like other researchers, so this should not impact the conclusions reached in this work.

Overall, it is disheartening to observe the recent trend to omit discussion of training data from these models. This information is crucial to helping researchers reproduce their work and analyze the results of those models.

5.3.3 Research Questions Revisited

In this section, we will revisit the research questions presented in the introduction and discuss the results in the context of each question.

RQ1 - How can we apply prompting techniques to movie review sentiment classification?

In this work, we have demonstrated two distinct approaches to movie review sentiment classification using prompting techniques:

- **Prompt Completion:** We evaluate standard language models by providing them with a prompt and asking them to complete it with the sentiment of the review. This approach is applied and evaluated in Experiment 1 of the sentiment classifier. We find that this approach varies wildly based on the prompt being used, as shown in Table 5.1. This illustrates the importance of prompt engineering for this approach, as models can achieve very strong or very poor results based on the selected prompt. It demonstrates a very simple approach to applying prompting techniques to movie review sentiment classification that can be easily applied to a wide variety of models and tasks.

- **Prompt Learning:** We conduct fine-tuning on the language models using an augmented dataset that demonstrates how to answer the prompts to the language models. With this additional training, we could then repeat the previous experiment to observe the impact of this fine-tuning process. This approach is applied and evaluated in Experiment 2 of the sentiment classifier. We find that this approach achieves moderate improvements over the previous approach, as shown in Figure 5.1. This demonstrates a second approach to applying prompting techniques to movie review sentiment classification that requires additional effort but can achieve stronger results.

We demonstrate these two approaches that use prompting techniques in this work and achieve strong results using both. These two approaches vary in complexity, with the first approach being very simple to implement and the second approach requiring additional effort. Both approaches can be applied to a wide variety of models and tasks, demonstrating the flexibility of prompting techniques for movie review sentiment classification. This supports the current trend in NLP towards prompting techniques, as they are simple to implement and can achieve strong results.

RQ2 - What are the most suitable language models and architectures to perform prompt-based movie sentiment classification?

This work has evaluated a total of eleven language models for the prompt completion approach and ten language models for the prompt learning approach, achieving results that varied significantly between models.

The best model is Llama 2, which achieves state-of-the-art results on this task, as we observe in Figure 5.1. It has outperformed the reported results of every other model for the prompt learning approach, achieving a 98.53% accuracy. Since this model is available for fine-tuning, it is well suited for use on many prompt-based tasks, particularly those that require fine-tuning.

Strong results are also attained using the gpt-4 model by OpenAI, which achieves outstanding results without any fine-tuning. This model was only evaluated for the prompt completion approach, as fine-tuning is not available. Despite this, it outperforms many models for both experiments, as we can observe in Figure 5.1. In fact, like Llama 2, it outperforms any model on the PapersWithCode leaderboard for this task, demonstrating its outstanding performance.

Unsurprisingly, the two OpenAI models from the gpt-3.5 family (babbage and davinci) also attained excellent results. These models were evaluated for both the prompt completion and prompt learning approaches, achieving promising results for the completion approach but stronger results with prompt learning. The davinci model outperforms the babbage model for both approaches, demonstrating the impact of model size on performance. Based on the excellent performance of gpt-3.5 and gpt-4, it is evident that the GPT family is the most suitable language model for this task.

In any case, we observe that any model can achieve promising results when we apply appropriate prompt engineering. With certain prompts, RoBERTa achieves results that are comparable to the high standard set by the GPT models, while even smaller models like DistilBERT and ELECTRA are able to outperform GPT when GPT is paired with a poor prompt. This demonstrates the importance of prompt engineering for this task, as it can have a significant impact on the performance of these models.

RQ3 - How does the performance of a prompt-based movie sentiment classification system compare to traditional Machine Learning (ML) and non-prompt-based language models?

This work has compared the best-performing models from the sentiment classification system to a selection of models from the PapersWithCode leaderboard for this task. The best-performing models on this dataset ranged from 87.40% to 96.21% accuracy, while our results obtained were 91.37% for xlm-roberta-large, 91.87% for gpt-3.5-babbage, 95.11% for gpt-3.5-davinci, and 98.53% for Llama 2. These results are depicted in Figure 5.3, comparing our obtained results to the best results from the leaderboard.

These results demonstrate that the best-performing models from our prompt-based sentiment classification system outperform many of the existing baselines for this task. This supports the current trend in NLP towards prompting techniques, as they are simple to implement and can achieve strong results. Prompt-based systems also possess some of the following advantages that are observed in this work:

- **Training Data:** Prompt-based systems require significantly less training data than traditional ML systems to achieve competitive results. In this work, our prompt-based systems achieve results that outperform models with far more training data. This is a significant advantage, as it is often difficult to obtain large amounts of training data for a specific task.
- **Flexibility:** Prompt-based systems are very flexible, as a prompt can simply be modified to improve the results. For traditional ML models, developers would typically need to re-train their system. In this work, we demonstrate the performance of our systems using a set of 12 varying prompts that achieve significantly different results. This demonstrates the flexibility of prompt-based systems, as they can be easily modified to achieve different results.

It is also worth noting that weaknesses possessed by prompt-based systems are typically shared with traditional ML systems. For example, these systems are not inherently interpretable and require ongoing research to better understand their inner workings. This is a weakness shared with many traditional ML systems, as they are also not inherently interpretable. This is an area of ongoing research, as we have demonstrated in this work.

Additionally, in Experiment 1, we evaluated the performance of gpt-4 on the same set of prompts with this dataset. This model achieved a maximum accuracy of 99.56%, which vastly outperforms any existing baseline for this dataset, as well as the best results from our prompt-based sentiment classification system. This demonstrates the outstanding performance of gpt-4 for this task, but also the importance of prompt engineering. With the right prompt, these models can achieve outstanding results for this task.

RQ4 - How can XAI techniques be integrated into a prompt-based movie sentiment classification system to provide human-understandable explanations for model predictions?

In this work, we have explored several approaches to integrating XAI techniques into a prompt-based movie sentiment classification system. These approaches include both self-explaining and post-hoc explanation approaches, including the following:

- **LIME:** LIME is a post-hoc explainability method that focuses on evaluating the impact of permutations on the result.
- **SHAP:** SHAP is a post-hoc explainability method that focuses on calculating Shapley Values to explain the decision.
- **Integrated Gradients:** Integrated Gradients is a post-hoc explainability method that focuses on evaluating the impact of each token on the result using the model gradients.
- **SELFEXPLAIN:** SELFEXPLAIN is a self-explaining approach that augments the model with two additional layers that are trained to generate global and local explanations for the model's predictions.
- **Generated Explanations:** This approach involves simply appending a request for an explanation to the prompt and evaluating the output of the model.

These approaches illustrate that a wide variety of XAI techniques can be integrated into a prompt-based movie sentiment classification system, including many of the most common approaches. This supports the current trend in NLP towards prompting techniques, as we can continue to explore XAI techniques with these systems.

RQ5 - What is the quality of explanations generated by the XAI component, and how can it be assessed?

To assess the quality of explanations generated by our XAI methods, we used the metrics of Sufficiency, Understandability, Trustworthiness, and Adequate Justification that were proposed in Rajagopal et al. 2021:

These metrics provide a global view of the quality of explanations generated by the XAI methods. Based on the evaluation of these metrics, we make the following observations:

- Generated explanations provide the strongest overall results, with the Llama 2 model performing slightly better than GPT-3.5 davinci. They are excellent at providing a strong justification for their prediction, as indicated by the strong score for Adequate Justification. They also convey an elevated level of trustworthiness, as measured by human evaluation. Lastly, they achieve strong sufficiency results, showing that the explanations are genuinely reflective of the model's decision making process.

- Integrated Gradients, SHAP, and LIME each achieve weaker results for their justification and trustworthiness. They are prone to using irrelevant tokens or attributes that do not adequately explain their prediction or convey a sense of trustworthiness. Their sufficiency scores are also quite low, with Integrated Gradients slightly outperforming SHAP and LIME.

Overall, the selected approach of using generated explanations from Llama 2 provides the highest level of trustworthiness and understandability. Its predictions are adequately justified by the provided explanations and these explanations strongly reflective of the model's decision making process. Based on this evaluation, we can say that the quality of explanations generated by the XAI component are of high quality.

5.3.4 Limitations

As with any work, there are some limitations to this work that should be considered. Some limitations of this work include:

- **Single Dataset:** This work only evaluates the proposed system on a single dataset, the IMDB dataset. While this dataset is commonly used for this task, there are a wide variety of sentiment analysis datasets available for use. In particular, this dataset uses binary labels (positive/negative) for reviews, while others include a neutral label or even numerical scores for sentiment. We cannot make conclusions about the performance of the proposed system on other datasets based on the results of this work.

- **Human Engineered Prompts:** This work only uses a set of 12 human-engineered prompts. These prompts achieve strong results on this task, but there are likely other prompts that could achieve even stronger results. Our results indicate that the proposed system performs well with unseen prompts, but we cannot make conclusions about the performance of the proposed system on other prompts based on the results of this work.

- **GPT-3.5 and GPT-4 are Closed Source:** GPT-3.5 and GPT-4 are two state-of-the-art LLMs. Their evaluation on this task is a useful contribution of this work. Unfortunately, both models are closed source at the time of writing and are only available for use through the OpenAI API. This means we are unable to fully evaluate them for our explainability task, as we cannot access

their internals for the Integrated Gradients approach, for example. For GPT-4, it also means we are unable to perform fine-tuning to evaluate it for prompt learning. This is a limitation of this work, as we cannot fully evaluate these models for this task.

- **Computationally Expensive Approaches:** Some of the approaches used in this work are computationally expensive. In particular, LIME and SHAP require a large number of model evaluations to generate explanations. This is a limitation of this work that limits the practical use of these approaches, particularly for real-time usage.

- **Reduced Sample Size for Explanations:** Due to time constraints for the evaluation of explanations, we reduced our original sample size of 100 reviews down to 50 reviews and only performed human evaluations for the Llama 2 implementations of LIME, SHAP, and Integrated Gradients. While this sample size is the same as the original SELFEXPLAIN work (Rajagopal et al. 2021), it would be beneficial to evaluate the other models and approaches using a larger sample size. This is a limitation of this work that could be addressed in future work.

Some of these limitations can be addressed in future work. We propose possible avenues for further work in the following chapter.

5.4 Summary

In this chapter, we discussed the experiments performed to create and evaluate the proposed sentiment classifier system on the IMDB dataset and for the explainability component of this work.

For the sentiment classifier, we conducted two experiments using two distinct prompt-based approaches on a variety of language models. In the first experiment, we analyzed the performance of these language models with prompt engineering alone, with no fine-tuning on the data. We used a set of 12 prompts proposed in the previous chapter on the IMDB dataset. The results of this experiment vary significantly by language model, but demonstrated that these models are strong at this task when paired with the right prompts. This experiment served as a benchmark to compare to the second experiment, where we fine-tuned the language models on a dataset consisting of the completed prompts. This process is referred to as prompt

learning, as the language models learn how to complete the prompts properly. We used the same set of prompts as the previous experiment and re-ran the previous experiment with the fine-tuned models. We found that the fine-tuning process has a positive impact on the performance of all models, demonstrating the usefulness of prompt learning for this task. We compared the results of our prompt learning systems to some existing baselines, where we found that the best results of our second experiment outperform many previous works on this task. These experiments have resulted in a set of prompt learning-based models that perform well on the IMDB sentiment analysis task and can be used for the proposed system.

For the explainability component, we explored a set of 5 different approaches to provide human-understandable explanations of the prompt learning-based models. In the first experiment, we used two standard post-hoc explainability methods, LIME and SHAP, to generate explanations for the predictions of the models and then convert the models to a human-readable format. LIME uses small permutations to the input to determine key tokens in a model's decision, while SHAP calculates Shapley values to indicate how important various tokens are in the decision. In the second experiment, we applied an additional post-hoc explainability method called Integrated Gradients to the models. With Integrated Gradients, we began with an empty string and observed how progressing towards the final review influences the decision. From this, we can determine which tokens were valuable in the model's decision. Next, we conducted an experiment with SELFEXPLAIN, which consists of two additional explainability layers that augment the existing neural classifier. We used the RoBERTa model for this experiment, as it has been implemented previously, and we cannot accomplish it with GPT since they are closed-source models. Lastly, we studied the ability of Llama 2 and GPT-3.5 models to generate explanations for their own predictions. We then evaluated the effectiveness of these approaches using the metrics described in the methodology section, then selected the best-performing approach for use in the proposed system.

Lastly, we provided a discussion of the work performed in these experiments. We began by reviewing the research questions proposed in the introduction to this thesis and providing the findings with respect to each question. We then discussed the limitations of this work that can be addressed in future work. Overall, our findings are that Llama 2 achieves state-of-the-art results on the IMDB dataset and we can apply a wide variety of XAI to Llama 2 to achieve explainability of its predictions.

Chapter 6

Conclusion and Future Work

This chapter reflects on the key contributions and findings of this work towards the related fields. It then provides a comprehensive summary of the work. Lastly, it proposes some avenues for future work that address some of the limitations discussed in the previous chapter.

6.1 Summary of Contributions

Through this research, we have made several contributions towards the fields of Prompt Learning, Sentiment Analysis, and Explainable AI. The key contributions of this work are as follows:

- **Novel Explainable Prompt Learning System:** We developed an innovative prompt learning system that combines the benefits of prompt learning with the benefits of explainable AI. To our knowledge, this is the first system to incorporate XAI methods to a prompt learning-based system and the first system to apply prompt learning to movie review sentiment analysis.
- **Comprehensive Model Evaluation:** We have conducted a comprehensive evaluation of 12 language models, including the latest large language models like GPT-4 and Llama 2. We have evaluated these models using two approaches, prompt completion and prompt learning, and compared the results to existing baselines for this task. We achieved state-of-the-art results for this task using our system. These results have illustrated the strength of the latest

models and the importance of effective prompt engineering to obtain strong results.

- **Use and Evaluation of Explainability Approaches:** In addition to our generative explanation method, we have applied and evaluated four other XAI approaches with this system, including LIME, SHAP, Integrated Gradients, and SELFEXPLAIN. We have evaluated them comprehensively using four standard metrics and compared them to find the strongest method for this task.
- **Foundational Work for Future Research:** As a relatively novel field, prompt learning is still in its infancy. This work provides a strong foundation for future research in this field, as it demonstrates the effectiveness of prompt learning for movie review sentiment analysis and provides a set of models that can be used for future research.

The findings from this thesis help provide a strong foundation for further work in this field. We hope that this work will inspire future research in this field and help to advance the fields of Prompt Learning, Sentiment Analysis, and Explainable AI.

6.2 Conclusion

As the development of complicated large language models continues rapidly, the field of NLP has been focused on prompting these models to achieve outstanding results in a wide variety of tasks. In particular, these models are being used by the general public as a resource for a wide variety of tasks, including general question and answering, supporting their work, or facilitating decision making. While these are exciting advancements for the field, there has been slow progress on truly understanding the decision making of these models. This is a significant issue, as it is difficult to trust the decisions of these models without understanding their inner workings.

In this work, we have conducted two experiments that explored different prompting approaches for sentiment analysis on movie reviews from the IMDB dataset. Our first experiment uses human-generated prompts to solicit predictions from a set of 12 language models. Our second experiment uses the same set of prompts to fine-tune these models, a process called prompt learning. This second experiment achieves state-of-the-art results for this task, achieving a 98.53% accuracy using the fine-tuned Llama 2 model. This has further reinforced the power of prompting these

models, as we have achieved state-of-the-art results with a relatively small amount of training data.

Following the experiments with these prompting-based approaches, we explore the field of Explainable Artificial Intelligence (XAI), providing the first exploration of XAI techniques with prompt learning-based systems. We explore five distinct XAI approaches, including LIME, SHAP, Integrated Gradients, SELFEXPLAIN, and the explanations generated by the model. We evaluate these approaches using four standard metrics, including adequate justification, understandability, trustworthiness, and sufficiency. Automated evaluation of the sufficiency attains 77.05% for Llama 2 generated explanations. Human evaluation gives Llama 2 generated explanations a 75% adequate justification rate, a 100% understandability score, and an 86% trustworthiness score. These are extremely strong results for a novel system like this.

This work has provided numerous key contributions to the fields of NLP, sentiment analysis, and XAI. It has produced a new state-of-the-art system for sentiment analysis on the IMDB dataset, alongside the comprehensive evaluation performed on a wide variety of modern language models. It has also provided the first exploration of XAI techniques with prompt learning-based systems, providing a strong foundation for future research in this field. This research has demonstrated the power of prompt learning-based systems and the importance of XAI techniques for these systems.

6.3 Future Work

At the end of the discussion in Chapter 5, we mentioned some of the key limitations of this work. These limitations provide a set of avenues for future work that can be explored to improve upon this work. Some of these avenues for future work include:

- **Further Datasets:** This work exclusively evaluated the proposed system on the IMDB dataset. There are a wide variety of sentiment analysis datasets available for use, including datasets with neutral labels and numerical scores for sentiment. Future work could create and evaluate systems on these datasets to determine the performance of the proposed system on these datasets. This would provide a more comprehensive evaluation of the proposed system and provide a more complete understanding of the performance of the proposed system.

- **Automated Prompt Engineering:** This work used a set of 12 human-engineered prompts for the sentiment classifier. These prompts obtained strong results, but there are likely other prompts that could achieve even stronger results. Future work could explore automated approaches to prompt engineering, where a system could automatically generate a set of prompts to evaluate. This would provide an even stronger understanding of the impact of prompt engineering on the performance of the proposed system.
- **Advanced Explanation Techniques:** This work explored a limited set of four XAI approaches for the selected models. Future work could explore additional XAI approaches for these models. One possibility is a SELFEXPLAIN-like approach for more complicated models that can draw on the training data to support justification for its decisions. This would provide a more complete understanding of the inner workings of these models and provide a more complete set of explanations for the proposed system.

These research avenues provide a set of opportunities for future work that can be explored to improve upon this work. We hope that this work will inspire future research in this field and help to advance the fields of Prompt Learning, Sentiment Analysis, and Explainable AI.

In particular, we hope that this work encourages further work in the explainability of these large language models to properly understand how they make predictions. The field has progressed towards a more closed source field, with many LLM developers failing to provide the training data, for example. This limits the reproducibility and transparency of their models. This unfortunate limitation could be mitigated with further work on explainability with these systems.

6.4 Ethical Statement

This research followed all the ethical guidelines and principles of the University of Ottawa and the Ottawa-Carleton Institute of Computer Science.

During this work, we leveraged AI tools such as ChatGPT and GitHub Copilot to assist with the development of the proposed system. These tools were exclusively used to assist with the development of the proposed system and were not used to generate any of the content of this thesis. All AI-generated code was validated and modified by the author to ensure it was suitable for use in this research. The content of this thesis was generated exclusively by the author of this thesis.

Ethics is a key concern for this research, like with all ongoing AI development. One key concern is that training data may introduce issues regarding privacy or biases that get reflected by the models trained. To mitigate this concern, we are using the IMDB dataset from Maas et al. 2011, which has been a standard for sentiment analysis for over a decade. This data is well anonymized. It also mitigates bias concerns by limiting the number of reviews for a single film and by ensuring balance of the dataset. These mitigation strategies are not perfect, but they help to address some of the ethical concerns, limitations, and potential biases of using AI models.

These concerns are important to consider for AI systems and have become a major priority for researchers in the field. We have attempted to address these concerns in this work to the best of our ability. In addition to the measures taken above, we have voluntarily completed the Association for Computational Linguistics (ACL) Responsible NLP Checklist to address issues of research ethics, societal impact, and reproducibility. This completed checklist is provided in Appendix C.

Bibliography

Attanasio, Giuseppe et al. (May 2023). "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*. Dubrovnik, Croatia: Association for Computational Linguistics, pp. 256–266. URL: https://aclanthology.org/2023.eacl-demo.29.

Beltagy, Iz, Matthew E. Peters, and Arman Cohan (2020). *Longformer: The Long-Document Transformer*. arXiv: 2004.05150 [cs.CL].

Bingyu, Zhang and Nikolay Arefyev (May 2022). "The Document Vectors Using Cosine Similarity Revisited". In: *Proceedings of the Third Workshop on Insights from Negative Results in NLP*. Dublin, Ireland: Association for Computational Linguistics, pp. 129–133. DOI: 10.18653/v1/2022.insights-1.17. URL: https://aclanthology.org/2022.insights-1.17.

Boyd-Graber, Jordan et al. (July 2022). "Human-Centered Evaluation of Explanations". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorial Abstracts*. Seattle, United States: Association for Computational Linguistics, pp. 26–32. DOI: 10.18653/v1/2022.naacl-tutorials.4. URL: https://aclanthology.org/2022.naacl-tutorials.4.

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

Clark, Kevin et al. (2020). *ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators*. arXiv: 2003.10555 [cs.CL].

Clinciu, Miruna-Adriana and Helen Hastie (2019). "A Survey of Explainable AI Terminology". In: *Proceedings of the 1st Workshop on Interactive Natural Language*

Technology for Explainable Artificial Intelligence (NL4XAI 2019). Association for Computational Linguistics, pp. 8–13. DOI: 10.18653/v1/W19-8403. URL: https://aclanthology.org/W19-8403.

Danilevsky, Marina et al. (Dec. 2020). "A Survey of the State of Explainable AI for Natural Language Processing". In: *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. Suzhou, China: Association for Computational Linguistics, pp. 447–459. URL: https://aclanthology.org/2020.aacl-main.46.

Deng, Yong et al. (July 2022). "BEIKE NLP at SemEval-2022 Task 4: Prompt-Based Paragraph Classification for Patronizing and Condescending Language Detection". In: *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Seattle, United States: Association for Computational Linguistics, pp. 319–323. DOI: 10.18653/v1/2022.semeval-1.41. URL: https://aclanthology.org/2022.semeval-1.41.

Devlin, Jacob et al. (June 2019). "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)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

Ding, Ning et al. (May 2022). "OpenPrompt: An Open-source Framework for Prompt-learning". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Dublin, Ireland: Association for Computational Linguistics, pp. 105–113. DOI: 10.18653/v1/2022.acl-demo.10. URL: https://aclanthology.org/2022.acl-demo.10.

Gao, Tianyu, Adam Fisch, and Danqi Chen (Aug. 2021). "Making Pre-trained Language Models Better Few-shot Learners". 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, pp. 3816–3830. DOI: 10.18653/v1/2021.acl-long.295. URL: https://aclanthology.org/2021.acl-long.295.

Goyal, Saurabh et al. (July 2020). "PoWER-BERT: Accelerating BERT Inference via Progressive Word-vector Elimination". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 3690–3699. URL: https://proceedings.mlr.press/v119/goyal20a.html.

Grynbaum, Michael and Ryan Mac (Feb. 2024). *The Times Sues OpenAI and Microsoft Over A.I. Use of Copyrighted Work*. URL: https://www.nytimes.com/2023/12/27/business/media/new-york-times-open-ai-microsoft-lawsuit.html.

Guan, Ziyu et al. (2016). "Weakly-Supervised Deep Learning for Customer Review Sentiment Classification." In: *IJCAI*, pp. 3719–3725.

Gupta, Himanshu et al. (May 2023). ""John is 50 years old, can his son be 65?" Evaluating NLP Models' Understanding of Feasibility". In: *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. Dubrovnik, Croatia: Association for Computational Linguistics, pp. 407–417. URL: https://aclanthology.org/2023.eacl-main.30.

Han, Xu et al. (2018). "FewRel: A Large-Scale Supervised Few-Shot Relation Classification Dataset with State-of-the-Art Evaluation". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 4803–4809. DOI: 10.18653/v1/D18-1514. URL: https://aclanthology.org/D18-1514.

Harris, Charles R. et al. (Sept. 2020). "Array programming with NumPy". In: *Nature* 585.7825, pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

Hase, Peter and Mohit Bansal (July 2020). "Evaluating Explainable AI: Which Algorithmic Explanations Help Users Predict Model Behavior?" In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 5540–5552. DOI: 10.18653/v1/2020.acl-main.491. URL: https://aclanthology.org/2020.acl-main.491.

He, Ruining and Julian McAuley (Apr. 2016). "Ups and Downs". In: *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee. DOI: 10.1145/2872427.2883037. URL: https://doi.org/10.1145%2F2872427.2883037.

Heinsen, Franz A. (2022). *An Algorithm for Routing Vectors in Sequences*. arXiv: 2211.11754 [cs.LG].

Jain, Sarthak et al. (July 2020). "Learning to Faithfully Rationalize by Construction". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 4459–4473. DOI: 10.18653/v1/2020.acl-main.409. URL: https://aclanthology.org/2020.acl-main.409.

Jimenez Gutierrez, Bernal et al. (Dec. 2022). "Thinking about GPT-3 In-Context Learning for Biomedical IE? Think Again". In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Kokalj, Enja et al. (Apr. 2021). "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, pp. 16–21. URL: https://aclanthology.org/2021.hackashop-1.3.

Lester, Brian, Rami Al-Rfou, and Noah Constant (Nov. 2021). "The Power of Scale for Parameter-Efficient Prompt Tuning". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 3045–3059. DOI: 10.18653/v1/2021.emnlp-main.243. URL: https://aclanthology.org/2021.emnlp-main.243.

Lewis, D. (1997). "Reuters-21578 text categorization test collection". In: *Distribution 1.0, ATT Labs-Research*. URL: https://cir.nii.ac.jp/crid/1570291224116302976.

Li, Xiang Lisa and Percy Liang (Aug. 2021). "Prefix-Tuning: Optimizing Continuous Prompts for Generation". 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, pp. 4582–4597. DOI: 10.18653/v1/2021.acl-long.353. URL: https://aclanthology.org/2021.acl-long.353.

Liu, Pengfei et al. (2021). *Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing*. arXiv: 2107.13586 [cs.CL].

Liu, Yiheng et al. (2023). *Summary of ChatGPT/GPT-4 Research and Perspective Towards the Future of Large Language Models*. arXiv: 2304.01852 [cs.CL].

Liu, Yinhan et al. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. arXiv: 1907.11692 [cs.CL].

Lu, Yang (2019). "Artificial intelligence: a survey on evolution, models, applications and future trends". In: *Journal of Management Analytics* 6.1, pp. 1–29. DOI: 10.1080/23270012.2019.1570365. eprint: https://doi.org/10.1080/23270012.2019.1570365. URL: https://doi.org/10.1080/23270012.2019.1570365.

Lucaci, Diana and Diana Inkpen (2021). "Towards Unifying the Explainability Evaluation Methods for NLP". In: *Natural Language Processing and Chinese Computing - 10th CCF International Conference, NLPCC 2021, Qingdao, China, October 13-17, 2021, Proceedings, Part II*. Ed. by Lu Wang et al. Vol. 13029. Lecture Notes in Computer Science. Springer, pp. 303–314. DOI: 10.1007/978-3-030-88483-3_23. URL: https://doi.org/10.1007/978-3-030-88483-3%5C_23.

Lundberg, Scott and Su-In Lee (2017). *A Unified Approach to Interpreting Model Predictions*. arXiv: 1705.07874 [cs.AI].

Maas, Andrew L. et al. (June 2011). "Learning Word Vectors for Sentiment Analysis". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, pp. 142–150. URL: http://www.aclweb.org/anthology/P11-1015.

McKinney, Wes (2010). "Data Structures for Statistical Computing in Python". In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

Mehra, Sidharth and Mohammed Hasanuzzaman (May 2020). "Detection of Offensive Language in Social Media Posts." PhD thesis. DOI: 10.13140/RG.2.2.23097.80485.

Migliaccio, Alessandro (2023). *Systems engineering neural networks*. eng. Hoboken, NJ, USA: Wiley. ISBN: 1119902010.

Misra, Rishabh (June 2018). *News Category Dataset*. DOI: 10.13140/RG.2.2.20331.18729.

Moreno, Antonio (2020). *Sentiment Analysis for Social Media*. eng. MDPI - Multidisciplinary Digital Publishing Institute. ISBN: 3-03928-573-4.

Mosca, Edoardo et al. (Oct. 2022). "SHAP-Based Explanation Methods: A Review for NLP Interpretability". In: *Proceedings of the 29th International Conference on Computational Linguistics*. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, pp. 4593–4603. URL: https://aclanthology.org/2022.coling-1.406.

OpenAI (2023). *GPT-4 Technical Report*. arXiv: 2303.08774 [cs.CL].

Pang, Bo and Lillian Lee (June 2005). "Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales". In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 115–124. DOI: 10.3115/1219840.1219855. URL: https://aclanthology.org/P05-1015.

Papineni, Kishore et al. (2002). "BLEU: A Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. ACL '02. Philadelphia, Pennsylvania: Association for Computational Linguistics, pp. 311–318. DOI: 10.3115/1073083.1073135. URL: https://doi.org/10.3115/1073083.1073135.

Pluciński, Kamil and Hanna Klimczak (Aug. 2021). "GHOST at SemEval-2021 Task 5: Is explanation all you need?" In: *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*. Online: Association for Computational Linguistics, pp. 852–859. DOI: 10.18653/v1/2021.semeval-1.114. URL: https://aclanthology.org/2021.semeval-1.114.

Radford, Alec, Karthik Narasimhan, et al. (2018). "Improving language understanding by generative pre-training". In.

Radford, Alec, Jeffrey Wu, et al. (2019). "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8, p. 9.

Rajagopal, Dheeraj et al. (Nov. 2021). "SELFEXPLAIN: A Self-Explaining Architecture for Neural Text Classifiers". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 836–850. DOI:

10.18653/v1/2021.emnlp-main.64. URL: https://aclanthology.org/2021.emnlp-main.64.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2016). *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*. arXiv: 1602.04938 [cs.LG].

Richardson, Leonard (2007). "Beautiful soup documentation". In: *April*.

Saini, Lovedeep (2023). "Sentiment Analysis with GPT-3 and GPT-3.5". In: URL: https://medium.com/@lvdeep9/sentiment-analysis-on-imdb-dataset-with-gpt-3-and-gpt-3-5-b9ae8c5bc910.

Sanh, Victor et al. (2019). "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". In: *arXiv preprint arXiv:1910.01108*.

Sanyal, Soumya and Xiang Ren (Nov. 2021). "Discretized Integrated Gradients for Explaining Language Models". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 10285–10299. DOI: 10.18653/v1/2021.emnlp-main.805. URL: https://aclanthology.org/2021.emnlp-main.805.

Shapley, Lloyd S et al. (1953). "A value for n-person games". In.

Su, Yusheng et al. (July 2022). "On Transferability of Prompt Tuning for Natural Language Processing". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States: Association for Computational Linguistics, pp. 3949–3969. DOI: 10.18653/v1/2022.naacl-main.290. URL: https://aclanthology.org/2022.naacl-main.290.

Sun, Chi et al. (2019). "How to Fine-Tune BERT for Text Classification?" eng. In: *Chinese Computational Linguistics*. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 194–206. ISBN: 3030323803.

— (2020). *How to Fine-Tune BERT for Text Classification?* arXiv: 1905.05583 [cs.CL].

Sundararajan, Mukund, Ankur Taly, and Qiqi Yan (2017). *Axiomatic Attribution for Deep Networks*. arXiv: 1703.01365 [cs.LG].

Tan, Kian Long et al. (2022). "RoBERTa-LSTM: A Hybrid Model for Sentiment Analysis With Transformer and Recurrent Neural Network". In: *IEEE Access* 10, pp. 21517–21525. DOI: 10.1109/ACCESS.2022.3152828.

Touvron, Hugo et al. (2023). *Llama 2: Open Foundation and Fine-Tuned Chat Models*. arXiv: 2307.09288 [cs.CL].

Turek, Matt (n.d.). *Explainable Artificial Intelligence (XAI) (Archived)*. Website. Accessed: May 11 2023. URL: https://www.darpa.mil/about-us/about-darpa.

Vale, Daniel, Ali El-Sharif, and Muhammed Ali (2022). "Explainable artificial intelligence (XAI) post-hoc explainability methods: risks and limitations in non-discrimination law". eng. In: *Ai and ethics (Online)* 2.4, pp. 815–826. ISSN: 2730-5953.

Van Rossum, Guido and Fred L. Drake (2009). *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace. ISBN: 1441412697.

Vaswani, Ashish et al. (2017). *Attention Is All You Need*. arXiv: 1706.03762 [cs.CL].

Wang, Song et al. (2022). "Trustworthy assertion classification through prompting". In: *Journal of biomedical informatics* 132, p. 104139.

Wang, Ye et al. (July 2022). "PINGAN Omini-Sinitic at SemEval-2022 Task 4: Multi-prompt Training for Patronizing and Condescending Language Detection". In: *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Seattle, United States: Association for Computational Linguistics, pp. 313–318. DOI: 10.18653/v1/2022.semeval-1.40. URL: https://aclanthology.org/2022.semeval-1.40.

Wang, Zhiguo et al. (Aug. 2019). "Multi-Granular Text Encoding for Self-Explaining Categorization". In: *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Ed. by Tal Linzen et al. Florence, Italy: Association for Computational Linguistics, pp. 41–45. DOI: 10.18653/v1/W19-4805. URL: https://aclanthology.org/W19-4805.

Webson, Albert and Ellie Pavlick (July 2022). "Do Prompt-Based Models Really Understand the Meaning of Their Prompts?" In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States: Association for Compu-

tational Linguistics, pp. 2300–2344. DOI: 10.18653/v1/2022.naacl-main.167. URL: https://aclanthology.org/2022.naacl-main.167.

Wilson, Aidan (2020). "A Brief Introduction to Unsupervised Learning". In: *Towards Data Science* 2020. URL: https://towardsdatascience.com/a-brief-introduction-to-unsupervised-learning-20db46445283.

Wolf, Thomas et al. (2020). *HuggingFace's Transformers: State-of-the-art Natural Language Processing*. arXiv: 1910.03771 [cs.CL].

Wu, Hui and Xiaodong Shi (May 2022). "Adversarial Soft Prompt Tuning for Cross-Domain Sentiment Analysis". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, pp. 2438–2447. DOI: 10.18653/v1/2022.acl-long.174. URL: https://aclanthology.org/2022.acl-long.174.

Xia, Mengzhou et al. (2022). "Prompting ELECTRA: Few-Shot Learning with Discriminative Pre-Trained Models". In.

Xie, Qizhe et al. (2020). *Unsupervised Data Augmentation for Consistency Training*. arXiv: 1904.12848 [cs.LG].

Xu, Kelvin et al. (2016). *Show, Attend and Tell: Neural Image Caption Generation with Visual Attention*. arXiv: 1502.03044 [cs.LG].

Yang, Chenghao et al. (July 2023). "Efficient Shapley Values Estimation by Amortization for Text Classification". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Toronto, Canada: Association for Computational Linguistics, pp. 8666–8680. URL: https://aclanthology.org/2023.acl-long.483.

Yang, Sen et al. (Dec. 2020). "Making the Best Use of Review Summary for Sentiment Analysis". In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, pp. 173–184. DOI: 10.18653/v1/2020.coling-main.15. URL: https://aclanthology.org/2020.coling-main.15.

Yang, Zhilin et al. (2020). *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. arXiv: 1906.08237 [cs.CL].

Yasen, Mais and Sara Tedmori (2019). "Movies reviews sentiment analysis and classification". In: *2019 IEEE jordan international joint conference on electrical engineering and information technology (JEEIT)*. IEEE, pp. 860–865.

Zhang, Haoxing et al. (Dec. 2022). "Prompt-Based Meta-Learning For Few-shot Text Classification". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 1342–1357. URL: https://aclanthology.org/2022.emnlp-main.87.

Zhang, Xiang, Junbo Zhao, and Yann LeCun (2016). *Character-level Convolutional Networks for Text Classification*. arXiv: 1509.01626 [cs.LG].

Zhou, Zhi-Hua (2021). *Machine learning*. eng. Singapore: Springer. ISBN: 9789811519673.

Appendix A: Results for Sentiment Classifier Experiments

This appendix provides the detailed results for the sentiment classifier experiments. The results are given for accuracy, precision, recall, and F1-score. The results are provided for each model and prompt combination for both experiments. The results for the first experiment are provided in Table A.2, Table A.3, Table A.4, and Table A.5. The results for the second experiment are provided in Table A.6, Table A.7, Table A.8, and Table A.9.

To reduce the size of the tables, we use the acronyms described in Table A.1 to refer to the models:

Table A.1: Acronyms used to refer to the models in the results tables.

Acronym	**Model**
bbu	bert-base-uncased
blu	bert-large-uncased
xrb	xlm-roberta-base
xrl	xlm-roberta-large
dbu	distilbert-base-uncased
esg	electra-small-generator
ebg	electra-base-generator
elg	electra-large-generator
g35b	gpt-3.5-babbage
g35d	gpt-3.5-davinci
g4	gpt-4
llama	meta-llama/Llama-2-7b

A.1 Experiment 1

Table A.2: Accuracy comparison for each model and prompt combination for Experiment 1.

	Prompts											
	Instructional			**Completion**						**Question**		
Model	**1**	**2**	**3**	**1**	**2**	**3**	**4**	**5**	**6**	**1**	**2**	**3**
bbu	44.00	45.54	40.96	76.82	66.00	39.34	59.30	43.82	50.26	42.78	46.68	43.90
blu	47.46	49.32	39.76	71.20	74.20	52.24	57.74	47.54	60.28	48.86	48.76	48.36
xrb	46.14	44.38	59.78	75.52	56.20	74.44	54.36	53.36	63.32	46.44	43.68	68.72
xrl	51.40	39.22	56.98	84.04	67.70	61.86	40.52	49.98	64.66	49.06	40.46	66.76
dbu	40.02	38.76	42.90	66.16	55.00	46.86	53.52	52.00	60.20	41.90	39.78	44.22
esg	38.74	70.00	67.56	63.16	59.50	49.56	47.18	52.70	66.58	52.08	69.08	72.46
ebg	42.26	49.02	74.36	76.98	38.40	58.12	52.90	59.86	50.82	36.12	47.86	76.32
elg	43.54	49.04	69.56	80.66	76.12	70.44	75.52	54.52	68.54	40.38	47.66	75.78
g35b	81.25	82.13	82.83	65.21	63.24	60.65	58.05	66.75	61.94	81.46	81.19	81.33
g35d	81.85	84.21	84.91	63.14	81.58	63.32	65.69	69.07	62.11	84.21	91.53	91.79
g4	85.11	99.56	95.12	95.35	99.42	99.53	99.31	97.50	93.02	76.74	97.67	97.67
llama	75.44	94.74	85.96	96.49	85.96	89.47	85.96	92.98	89.47	92.98	98.25	94.74

Table A.3: Precision comparison for each model and prompt combination for Experiment 1.

	Prompts											
	Instructional			**Completion**						**Question**		
Model	**1**	**2**	**3**	**1**	**2**	**3**	**4**	**5**	**6**	**1**	**2**	**3**
bbu	21.74	23.62	37.86	82.10	62.73	41.76	55.60	45.76	49.88	19.37	24.40	43.55
blu	23.43	29.37	39.57	81.12	70.26	51.06	54.11	47.63	55.76	18.94	35.55	48.56
xrb	25.45	40.49	55.30	78.26	89.69	70.32	52.09	54.01	59.42	17.78	30.15	62.03
xrl	53.81	23.30	81.03	79.33	60.97	62.77	3.88	30.30	63.26	33.49	25.00	72.25
dbu	37.66	34.58	44.50	60.13	52.53	47.57	51.62	50.93	55.94	36.16	33.02	44.99
esg	38.42	67.22	61.97	57.60	55.24	48.23	22.52	52.57	67.24	51.82	66.55	67.86
ebg	22.61	48.20	69.90	87.27	34.51	64.46	53.91	62.29	50.56	20.22	46.48	82.29
elg	33.80	48.76	62.44	75.98	74.38	70.87	73.63	52.88	67.32	32.06	47.47	68.88
g35b	81.58	83.94	97.00	65.00	60.68	58.57	58.68	66.75	61.35	76.31	78.33	85.24
g35d	71.88	90.91	85.03	57.14	81.58	60.34	59.68	70.10	63.18	78.05	85.71	92.00
g4	79.41	99.73	95.45	89.47	99.37	99.44	99.27	100.00	95.24	61.54	96.77	95.24
llama	70.21	91.67	80.49	94.29	80.49	84.62	80.49	89.19	86.49	89.19	97.06	91.67

Table A.4: Recall comparison for each model and prompt combination for Experiment 1.

	Prompts											
	Instructional			**Completion**						**Question**		
Model	**1**	**2**	**3**	**1**	**2**	**3**	**4**	**5**	**6**	**1**	**2**	**3**
bbu	5.05	4.48	30.06	68.00	77.17	57.09	88.28	72.77	97.25	4.93	3.68	45.05
blu	2.71	1.70	41.17	54.51	83.03	84.89	96.20	60.04	95.64	1.01	7.71	73.13
xrb	4.57	26.30	97.74	69.98	13.01	83.68	97.25	38.91	81.62	2.26	10.46	94.91
xrl	12.85	9.94	17.09	91.64	96.53	56.40	0.85	0.81	68.24	2.95	10.14	53.33
dbu	32.32	26.59	62.14	93.86	94.51	72.08	96.93	83.03	92.24	36.16	33.02	44.99
esg	39.39	76.89	89.21	96.93	95.80	25.82	2.75	45.41	63.35	45.54	75.47	84.28
ebg	6.87	40.00	84.65	62.63	27.23	34.30	33.41	47.92	29.29	9.86	35.19	66.46
elg	14.67	57.86	96.61	89.09	78.95	68.40	78.75	74.59	70.83	18.26	53.86	93.17
g35b	91.18	81.18	70.53	79.41	100.00	94.44	97.37	100.00	97.44	93.59	87.12	78.24
g35d	100.00	74.07	83.65	82.35	81.58	94.59	100.00	100.00	100.00	100.00	100.00	88.46
g4	100.00	100.00	96.31	100.00	100.00	100.00	100.00	93.33	90.91	100.00	100.00	100.00
llama	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00

Table A.5: F1-score comparison for each model and prompt combination for Experiment 1.

	Prompts											
	Instructional			**Completion**						**Question**		
Model	**1**	**2**	**3**	**1**	**2**	**3**	**4**	**5**	**6**	**1**	**2**	**3**
bbu	8.20	7.54	33.51	74.39	69.20	48.24	68.23	56.18	65.94	7.86	6.39	44.29
blu	4.85	3.21	40.36	65.20	76.11	63.76	69.27	53.12	70.45	1.92	7.71	58.37
xrb	7.74	31.89	70.64	73.89	22.72	76.42	68.84	45.23	68.78	4.01	15.54	75.02
xrl	20.74	13.93	28.23	85.04	74.74	59.42	1.39	1.57	65.66	5.42	14.43	61.37
dbu	34.79	30.06	51.86	73.30	67.52	57.32	67.37	63.13	69.65	27.90	25.71	50.26
esg	38.90	71.73	73.14	72.26	70.08	33.62	4.90	48.73	65.24	48.47	70.73	75.18
ebg	10.54	43.72	76.57	72.92	30.44	44.78	41.26	54.17	37.09	13.25	40.06	73.54
elg	20.46	52.92	75.86	82.02	76.60	69.61	76.10	61.88	69.03	23.27	50.46	79.20
g35b	86.11	82.54	81.75	71.49	75.53	72.30	73.23	80.06	75.29	84.09	82.51	81.60
g35d	83.64	81.63	84.33	67.47	81.58	73.68	74.74	82.40	77.44	87.67	92.31	90.20
g4	88.52	99.86	95.88	94.44	99.68	99.72	99.63	96.55	93.02	76.19	98.36	97.56
llama	82.50	95.65	89.19	97.06	89.19	91.67	89.19	94.29	91.43	94.29	98.51	95.65

A.2 Experiment 2

Table A.6: Accuracy comparison for each model and prompt combination for Experiment 2.

	Prompts											
	Instructional			**Completion**						**Question**		
Model	**1**	**2**	**3**	**1**	**2**	**3**	**4**	**5**	**6**	**1**	**2**	**3**
bbu	49.65	50.16	44.32	79.96	70.08	46.83	66.79	45.21	53.01	44.12	48.15	44.16
blu	55.67	56.82	42.95	79.66	80.32	45.31	65.32	45.33	51.95	48.95	49.98	44.50
xrb	53.14	54.87	68.33	82.05	67.03	80.37	60.95	59.32	67.21	52.75	51.67	70.83
xrl	62.38	53.85	62.87	91.37	79.53	78.42	60.36	59.58	77.32	57.59	55.86	78.93
dbu	50.37	51.67	53.34	71.19	63.80	56.29	59.97	61.39	69.58	48.77	46.05	51.37
esg	46.32	80.71	69.37	69.18	65.47	55.31	53.98	55.61	68.09	58.68	76.55	74.90
ebg	47.32	55.05	75.11	80.32	49.32	63.96	60.01	59.97	51.05	48.12	59.12	76.03
elg	56.69	57.22	72.58	87.95	82.36	79.86	82.20	58.32	70.06	52.10	58.37	77.98
g35b	86.64	86.13	88.29	78.39	81.02	79.90	77.48	71.18	73.54	83.23	91.87	91.06
g35d	84.55	87.37	91.88	70.13	89.33	69.28	72.18	75.06	68.35	75.76	94.21	95.11
llama	83.44	96.25	90.12	98.51	89.99	93.40	90.11	95.61	93.38	94.76	98.53	96.05

Table A.7: Precision comparison for each model and prompt combination for Experiment 2.

	Prompts											
	Instructional			**Completion**						**Question**		
Model	**1**	**2**	**3**	**1**	**2**	**3**	**4**	**5**	**6**	**1**	**2**	**3**
bbu	32.18	32.83	40.06	84.73	65.18	46.27	61.22	48.41	55.32	30.68	36.19	42.90
blu	32.33	34.95	48.67	88.42	80.10	65.36	70.88	56.68	73.05	35.89	56.38	64.49
xrb	39.19	62.35	66.91	94.69	92.38	83.68	63.03	63.19	67.92	53.61	36.48	73.63
xrl	74.85	58.34	91.03	88.65	71.32	70.83	48.50	51.85	75.86	49.68	40.23	82.81
dbu	44.70	41.84	50.64	73.24	62.46	57.99	63.65	57.78	67.74	46.94	43.89	54.82
esg	42.07	72.87	69.47	69.70	63.69	57.06	32.22	57.58	65.90	54.62	81.12	71.15
ebg	48.75	53.74	72.92	87.65	63.60	72.78	64.75	68.83	52.88	67.65	66.18	86.76
elg	63.21	67.29	76.05	83.26	80.75	75.00	60.59	53.83	68.87	50.75	57.82	69.56
g35b	80.67	88.06	100.00	71.73	85.25	68.58	76.45	71.18	71.28	72.50	90.16	94.44
g35d	75.00	92.59	92.60	63.13	88.84	66.68	67.45	67.45	72.05	69.41	90.82	95.58
llama	80.74	94.42	86.12	97.12	87.28	91.27	90.54	92.35	91.19	93.52	98.31	94.84

Table A.8: Recall comparison for each model and prompt combination for Experiment 2.

	Prompts											
	Instructional			**Completion**						**Question**		
Model	**1**	**2**	**3**	**1**	**2**	**3**	**4**	**5**	**6**	**1**	**2**	**3**
bbu	28.38	23.89	33.01	74.66	80.72	61.83	88.99	74.08	95.60	15.42	11.40	45.46
blu	13.69	18.68	48.58	65.14	81.37	85.23	92.06	71.75	89.71	10.40	27.60	77.63
xrb	32.23	35.51	97.97	70.68	34.48	89.54	98.81	72.14	87.99	22.10	40.38	98.70
xrl	43.86	33.85	43.84	96.34	98.56	68.93	24.06	27.83	80.29	23.58	38.56	70.95
dbu	39.11	34.83	68.23	94.39	94.86	85.78	98.73	82.00	90.27	31.43	31.18	73.12
esg	43.49	84.19	88.05	92.47	92.16	31.37	17.08	55.85	76.76	54.15	70.04	77.43
ebg	36.41	45.74	92.85	74.23	33.17	41.91	38.42	55.59	32.03	32.34	65.99	62.30
elg	36.23	82.16	91.63	87.13	81.48	74.19	82.55	77.49	70.05	58.14	65.12	88.23
g35b	98.47	84.29	75.73	84.52	75.36	98.39	100.00	100.00	96.77	92.59	93.22	86.44
g35d	100.00	81.80	92.35	87.16	90.06	96.19	100.00	100.00	100.00	98.50	100.00	100.00
llama	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00

Table A.9: F1-score comparison for each model and prompt combination for Experiment 2.

	Prompts											
	Instructional			**Completion**						**Question**		
Model	**1**	**2**	**3**	**1**	**2**	**3**	**4**	**5**	**6**	**1**	**2**	**3**
bbu	30.16	29.26	36.19	79.38	72.12	52.93	72.53	58.56	70.08	20.53	17.34	44.14
blu	19.23	24.35	48.63	75.02	80.73	73.98	80.10	63.33	80.52	16.13	37.06	70.46
xrb	35.37	45.25	79.52	80.94	50.21	86.51	76.96	67.37	76.66	31.30	38.33	84.34
xrl	55.31	42.28	59.18	92.34	82.76	69.87	32.16	36.22	78.01	31.98	40.23	76.42
dbu	41.72	38.02	58.13	82.48	75.32	69.20	77.40	67.79	77.40	37.65	36.46	65.66
esg	42.77	78.12	77.66	79.48	75.33	40.48	22.32	56.70	70.87	54.38	75.17	74.15
ebg	41.69	49.84	81.69	80.38	43.60	53.19	48.23	61.50	39.89	43.76	66.08	72.53
elg	46.06	73.98	83.12	85.15	81.12	74.60	69.88	63.53	69.45	54.19	61.25	77.79
g35b	88.69	86.13	86.19	77.60	80.00	80.82	86.65	83.16	82.09	81.32	91.67	90.27
g35d	85.72	86.86	92.47	73.22	89.45	78.76	80.56	80.56	83.75	81.43	95.19	97.74
llama	89.34	97.13	92.54	98.54	93.21	95.43	95.03	96.02	95.39	96.65	99.15	97.35

Appendix B: Explainability Human Evaluation Process

This appendix outlines the process used to conduct the human evaluation of the explainability step.

B.1. Evaluator Information

One additional human evaluator was recruited for this evaluation and was recruited from the NLP group at the University of Ottawa. The researcher is a PhD student studying cognitive science at Carleton University. They are an international student residing in Canada that speaks English at a native-level fluency.

No compensation was provided for participation in this evaluation. Both evaluators were informed about how their work would be used in this thesis and consented to this usage.

B.2. Questionnaire

The questionnaire used to evaluate the explanation methods follows this format:

REVIEW: I hate reading reviews that say something like, 'Don't waste your time, this film stinks on ice.' It does to that reviewer yet for me, it may have some sort of naive charm. If you like the other 'Whistler' series films, this one will be watchable. If you like 40s noirish films, this one will be watchable.¡br /¿¡br /¿This film is not as good, in my opinion, as any of the earlier series entries which starred Richard Dix as the protagonist. It's much slower, and the plot is trite. You've seen this same narrative device used in many other films, and usually better.¡br /¿¡br

/¿But the acting is good, and so is the lighting, and the dialog. It's just lacking in energy and you'll likely figure out exactly what's going on and how it's all going to come out in the end not more than a quarter of the way through.¡br /¿¡br /¿The 'Whistler' series is semi-noir, and there character, mood, lighting, camera movement and angles are more important than the story itself. But this film is not noir. It's too light weight and Hollywood innocent for that. Neither Richard Dix's character nor those of any of his ladies in the previous films had to come to a good end. You just never knew until the end.¡br /¿¡br /¿But still, I'll recommend this one for at least a single viewing. I've watched it at least twice myself, and got a reasonable amount of enjoyment out of it both times.

TRUE LABEL: Positive

PREDICTIONS AND EXPLANATIONS

1. The sentiment of the review is negative.
2. The review is mixed but slightly leaning more towards positive. The reviewer mentions that the film has some naive charm, and if you like other 'Whistler' series films, it might be watchable. Although they consider it not as good as the earlier series entries, they still appreciate the good acting, lighting, and dialog. They also mention they still recommend it for at least a single viewing.
3. Words like 'But'; 'naive'; 'not'; 'hate' make the review seem positive to the model. However, words like 'It'; 'is' make the review seem negative to the model. Overall, the review is judged as positive.
4. The sentiment of this review is positive. The reviewer acknowledges that the film is not as good as the earlier entries in the series, but still recommends it for at least one viewing. The reviewer also notes that the acting, lighting, and dialog are good, and that they got a reasonable amount of enjoyment out of it both times.

EXPLANATION EVALUATION: Answer each of the following for each of the given explanations.

1. Does the explanation adequately justify the model prediction? (YES or NO)
2. Is the explanation understandable? (YES or NO)
3. How much do you trust the prediction of the model, considering its explanation? (Rank 1 to 5, 1 = low trust, 5 = high trust)

B.3. Agreement Between Evaluators

We conduct some analysis on the agreement between the two evaluators to gain additional insight into their evaluation.

Table B.10: Human evaluator agreement for explainability evaluation.

	Adeq. Just.	**Und.**	**Trust**
Percent Agreement	92%	48%	85%
Cohen's Kappa	84%	x	70%
Krippendorff's Alpha	x	87%	x

We observe that the agreement between evaluators is strong. For the binary metrics of Adequate Justification and Trustworthiness, we also provide the Cohen's Kappa that confirms the strong agreement. For Understandability, we provide the Krippendorff's Alpha, which indicates a strong correlation between the numerical scores assigned by the evaluators.

Appendix C: Responsible NLP Checklist

We use the Responsible NLP Checklist [1] from the ACL 2023 conference to evaluate the ethical implications of our work. The checklist is displayed below with the answers to each question provided in italics.

Section A - For every submission

1. Did you describe the limitations of your work? *Yes, Section 5.3.2 of the thesis discusses limitations of this work.*
2. Did you discuss any potential risks of your work? *Not applicable.*
3. Do the abstract and introduction summarize the paper's main claims? *Yes, the abstract and introduction both summarize the contributions of this work.*

Section B - Did you use or create scientific artifacts?

Yes, we used and created various scientific artifacts. We created the source code used to train and evaluate the models and the explainability step. We also used existing artifacts, such as Python libraries, pre-trained models, and the IMDB dataset. See Chapter 4 (Methodology) and Chapter 5 (Experiments).

1. Did you cite the creators of artifacts you used? *Yes, all artifacts are cited where they are mentioned in the text.*
2. Did you discuss the license or terms for use and / or distribution of any artifacts? *Yes, licenses and conditions are discussed where the artifacts are mentioned in the text.*

[1] `https://aclrollingreview.org/responsibleNLPresearch/`

3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)? *Yes, we describe the intended use of all existing artifacts and our created artifacts where used in Chapters 5 and 6.*

4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it? *Yes, we use a standard dataset that takes steps to anonymize the data. We discuss the IMDB dataset used in Section 4.3.*

5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.? *Yes, we provide this discussion in Section 4.3.*

6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? *Yes, we provide this information in Section 4.3.*

Section C - Did you run computational experiments?

Yes, we ran computational experiments to train and evaluate the models and to conduct the explainability step. Chapter 5 provides a detailed description of the experiments and the results.

1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used? *Yes. Computing infrastructure is described in Section 4.2. and parameters are discussed in Section 4.5.1.*

2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values? *Yes, Section 4.5.1. and Chapter 5.*

3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run? *Yes, included in Chapter 5.*

4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)? *Yes, in Section 4.2.*

Section D - Did you use human annotators (e.g., crowdworkers) or research with human participants?

Yes, human annotators were used during the explainability step.

1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.? *Yes, the instructions provided are in Appendix 2.*

2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)? *Yes, this is discussed in Appendix 2.*

3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used? *Yes, this is discussed in Appendix 2.*

4. Was the data collection protocol approved (or determined exempt) by an ethics review board? *Not applicable*

5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data? *Yes, this is discussed in Appendix 2.*